AWS Simple Storage Service (S3)

Prepared By –Kumuda Chandra Behera

AWS Simple Storage Service (S3): S3 is the object storage service provided by AWS. It is probably the most commonly used, go-to storage service for AWS users given the features like extremely high availability, security, and simple connection to other AWS Services

AWS S3 Terminology:

Bucket: Data, in S3, is stored in containers called *buckets*.

- Each bucket will have its own set of policies and configuration. This enables users to have more control over their data.
- Bucket Names must be unique.
- Can be thought of as a parent folder of data.
- There is a limit of 100 buckets per AWS accounts. But it can be increased if requested from AWS support.

Bucket Owner: The person or organization that owns a particular bucket is its *bucket owner*.

Import/Export Station: A machine that uploads or downloads data to/from S3.

Key: Key, in S3, is a unique identifier for an object in a bucket. For example, in a bucket 'ABC' your *GFG.java* file is stored at *javaPrograms/GFG.java* then *'javaPrograms/GFG.java'* is your object key for *GFG.java*.

- It is important to note that 'bucketName+key' is unique for all objects.
- This also means that there can be only one object for a key in a bucket. If you upload 2 files with the same key. The file uploaded latest will overwrite the previously contained file.

Versioning: Versioning means to always keep a record of previously uploaded files in S3. Points to note:

- Versioning is not enabled by default. Once enabled, it is enabled for all objects in a bucket.
- Versioning keeps all the copies of your file, so, it adds cost for storing multiple copies of your data. For example, 10 copies of a file of size 1GB will have you charged for using 10GBs for S3 space.
- Versioning is helpful to prevent unintended overwrites and deletions.
- Note that objects with the same key can be stored in a bucket if versioning is enabled (since they have a unique version ID).

Object: Fundamental entity type stored in AWS S3.

Access Control Lists (ACL): A document for verifying the access to S3 buckets from outside your AWS account. Each bucket has its own ACL.

Bucket Policies: A document for verifying the access to S3 buckets from within your AWS account, this controls which services and users have what kind of access to your S3 bucket. Each bucket has its own Bucket Policies.

Lifecycle Rules: This is a cost-saving practice that can move your files to AWS Glacier (The AWS Data Archive Service) or to some other S3 storage class for cheaper storage of old data or completely delete the data after the specified time.

Static Website Hosting: We can host static website using S3

Features of AWS S3:

- **Durability:** AWS claims Amazon S3 to have a 99.999999999% of durability (11 9's). This means the possibility of losing your data stored on S3 is one in a billion.
- **Availability:** AWS ensures that the up-time of AWS S3 is 99.99% for standard access.
    - Note that availability is related to being able to access data and durability is related to losing data altogether.
- **Server-Side-Encryption (SSE):** AWS S3 supports three types of SSE models:
    - **SSE-S3:** AWS S3 manages encryption keys.
    - **SSE-C:** The customer manages encryption keys.
    - **SSE-KMS:** The AWS Key Management Service (KMS) manages the encryption keys.
- **File Size support:** AWS S3 can hold files of size ranging from 0 bytes to 5 terabytes. A 5TB limit on file size should not be a blocker for most of the applications in the world.
- **Infinite storage space:** Theoretically AWS S3 is supposed to have infinite storage space. This makes S3 infinitely scalable for all kinds of use cases.
- **Pay as you use:** The users are charged according to the S3 storage they hold.
- **AWS-S3** is region-specific.

S3 storage classes:

AWS S3 provides multiple storage types that offer different performance and features and different cost structure.

- **Standard:** Suitable for frequently accessed data, that needs to be highly available and durable.

- **Standard Infrequent Access (Standard IA):** This is a cheaper data-storage class and as the name suggests, this class is best suited for storing infrequently accessed data like log files or data archives. Note that there may be a per GB data retrieval fee associated with Standard IA class.
- **Intelligent Tiering:** This service class classifies your files automatically into frequently accessed and infrequently accessed and stores the infrequently accessed data in infrequent access storage to save costs. This is useful for unpredictable data access to an S3 bucket.
- **One Zone Infrequent Access (One Zone IA):** All the files on your S3 have their copies stored in a minimum of 3 Availability Zones. One Zone IA stores this data in a single availability zone. It is only recommended to use this storage class for infrequently accessed, non-essential data. There may be a per GB cost for data retrieval.
- **Amazon S3 Glacier Instant Retrieval:** Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds. With S3 Glacier Instant Retrieval, you can save up to 68% on storage costs compared to using the S3 Standard-Infrequent Access (S3 Standard-IA) storage class,
- **Amazon S3 Glacier Deep Archive:** S3 Glacier Deep Archive is Amazon S3's lowest-cost storage class and supports long-term retention and digital preservation for data that may be accessed once or twice in a year. It is designed for customers—particularly those in highly-regulated industries, such as financial services, healthcare, and public sectors—that retain data sets for 7—10 years or longer to meet regulatory compliance requirements. S3 Glacier Deep Archive can also be used for backup and disaster recovery use cases, and is a cost-effective and easy-to-manage alternative to magnetic tape systems, whether they are on-premises libraries or off-premises services. S3 Glacier Deep Archive complements Amazon S3 Glacier, which is ideal for archives where data is regularly retrieved and some of the data may be needed in minutes. All objects stored in S3 Glacier Deep Archive are replicated and stored across at least three geographically-dispersed Availability Zones, protected by 99.999999999% of durability, and can be restored within 12 hours.

Uploading and copying objects using multipart upload

Multipart upload allows you to upload a single object as a set of parts. Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order. If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. In general, when your object size reaches 100 MB, you should consider using multipart uploads instead of uploading the object in a single operation.

Using multipart upload provides the following advantages:

- **Improved throughput** – You can upload parts in parallel to improve throughput.
- **Quick recovery from any network issues** – Smaller part size minimizes the impact of restarting a failed upload due to a network error.
- **Pause and resume object uploads** – You can upload object parts over time. After you initiate a multipart upload, there is no expiry; you must explicitly complete or stop the multipart upload.

- **Begin an upload before you know the final object size** – You can upload an object as you are creating it.

Copying objects

The copy operation creates a copy of an object that is already stored in Amazon S3.

You can create a copy of your object up to 5 GB in a single atomic operation. However, to copy an object that is greater than 5 GB, you must use the multipart upload API.

Using the copy operation, you can:

- Create additional copies of objects
- Rename objects by copying them and deleting the original ones
- Move objects across Amazon S3 locations (for example, us-west-1 and Europe)
- Change object metadata

- Each Amazon S3 object has metadata. It is a set of name-value pairs. You can set object metadata at the time you upload it. After you upload the object, you cannot modify object metadata. The only way to modify object metadata is to make a copy of the object and set the metadata. In the copy operation, set the same object as the source and target.

Amazon S3 – Cross Region Replication

The AWS S3 – Cross-region replication (CRR) allows you to replicate or copy your data in two different regions. But why do you need to set up CRR? There are many possible scenarios where setting up cross-region replication will prove helpful. Some of them are enlisted below:

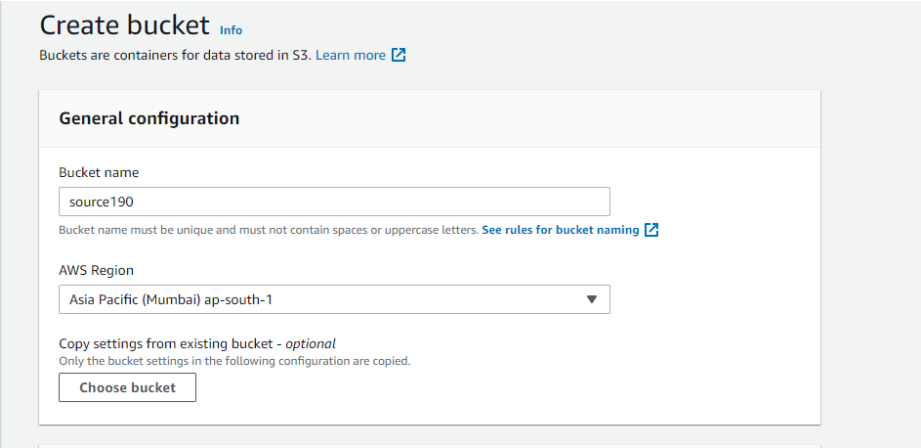
1. Improving latency and enhancing availability: If you are running a big organization with customers all around the world then making objects available to them with low latency is of great importance. By setting up cross-region replication you can enable your customers to get objects from S3 buckets which are nearest to their geographic location.
2. Disaster recovery: Having your data in more than one region will help you prepare and handle data loss due to some unprecedented circumstances.

3. **To meet compliance requirements:** Sometimes just to meet compliance requirements you will need to have a copy of your data in more than one region and cross-region replication can help you achieve that.
4. **Owner override:** With AWS S3 object replication in place you can maintain the same copy of data under different ownership. You can change the ownership to the owner of the AWS destination bucket even if the source bucket is owned by someone else.

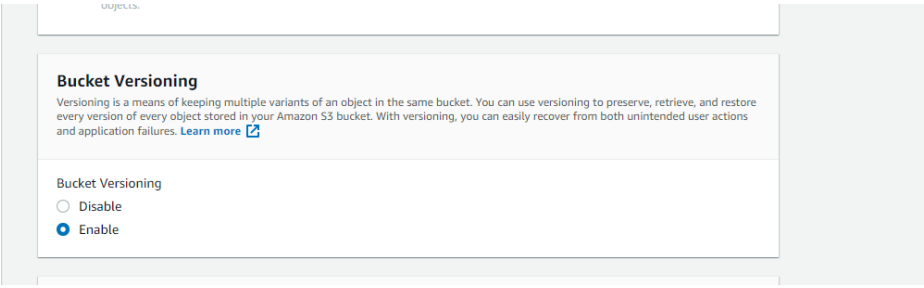
Setting up CRR:

Follow the below steps to set up the CRR:

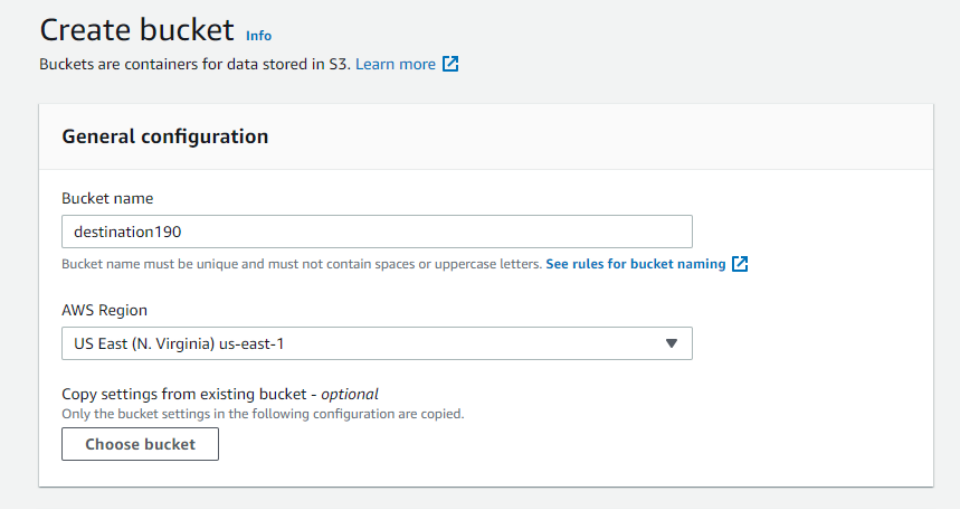
- Go to the AWS s3 console and create two buckets.
- Let's name our source bucket as source190 and keep it in the Asia Pacific (Mumbai) ap-south 1 region. Do not forget to **enable versioning**. Also, note that the S3 bucket name needs to be globally unique and hence try adding random numbers after bucket name.



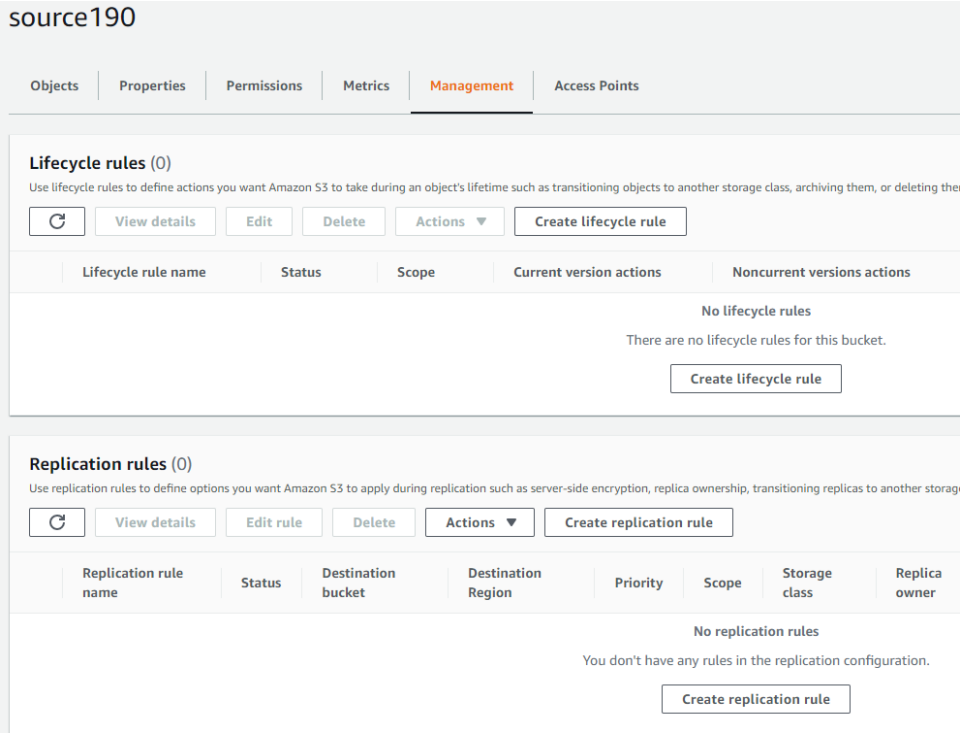
Source bucket: source190



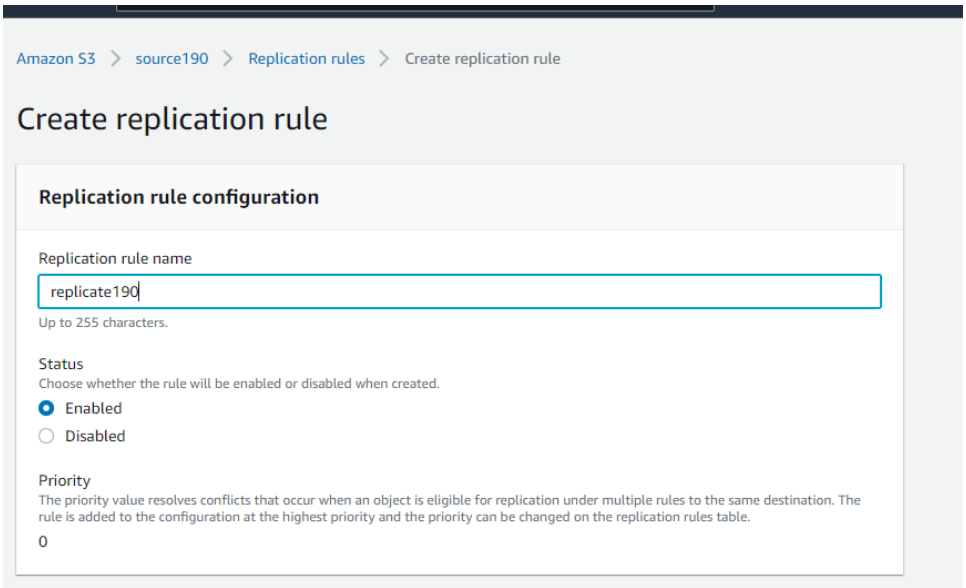
- Now following the same steps create a destination bucket: destination190 with versioning enabled but choose a different region this time.



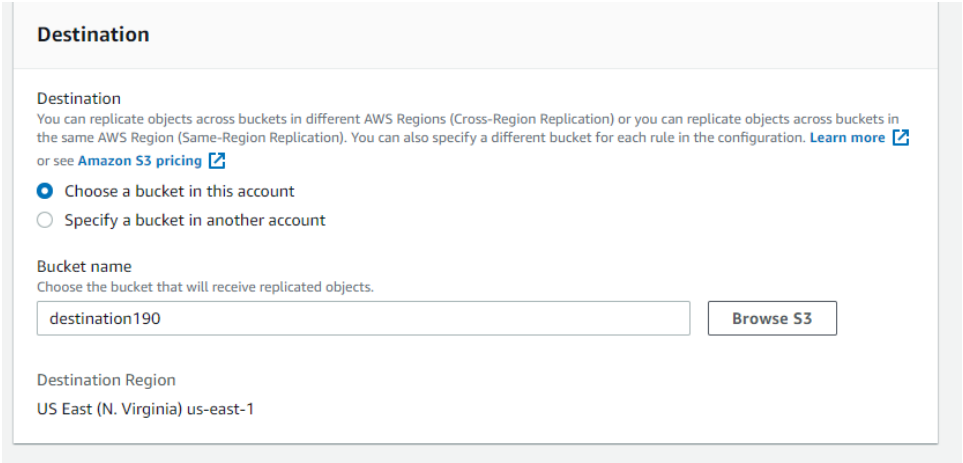
- Now click on your source bucket and head over to the management tab:



- Now, click on “Create a replication rule” and give your replication rule a name as “ replicate190”



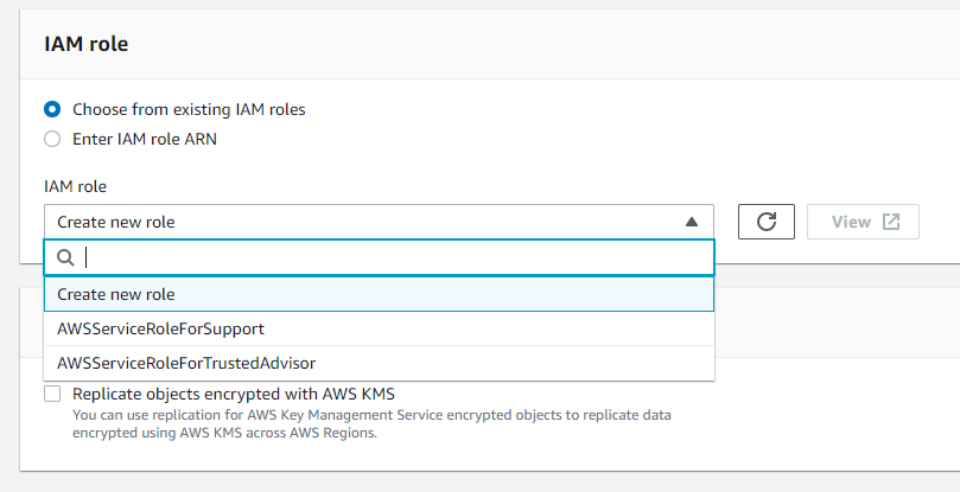
- Choose the destination bucket as “destination190”.



Set destination bucket

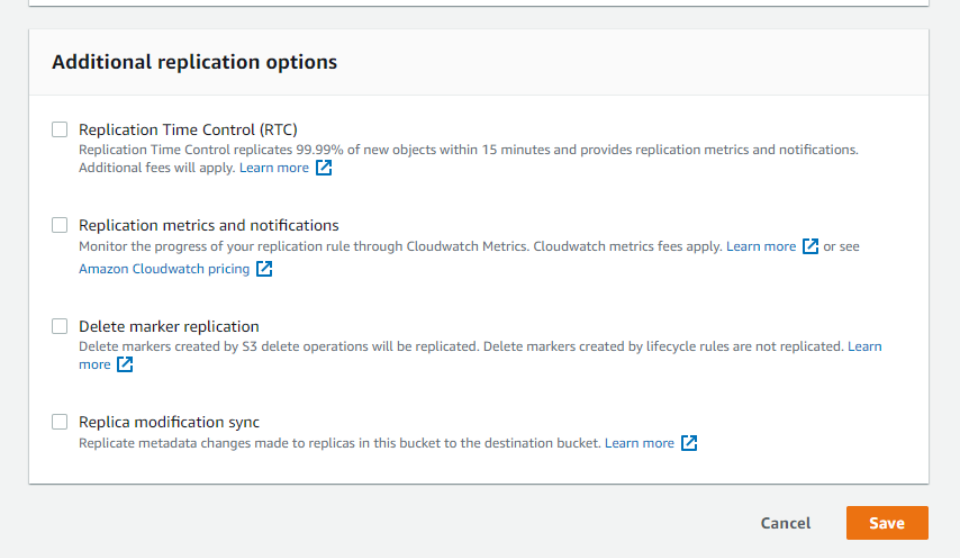
Notice that you have an option to choose a destination bucket in another account.

- In order to replicate objects from the source bucket to the destination bucket, you need to create an IAM role. So just create one by clicking on “create a new role”.

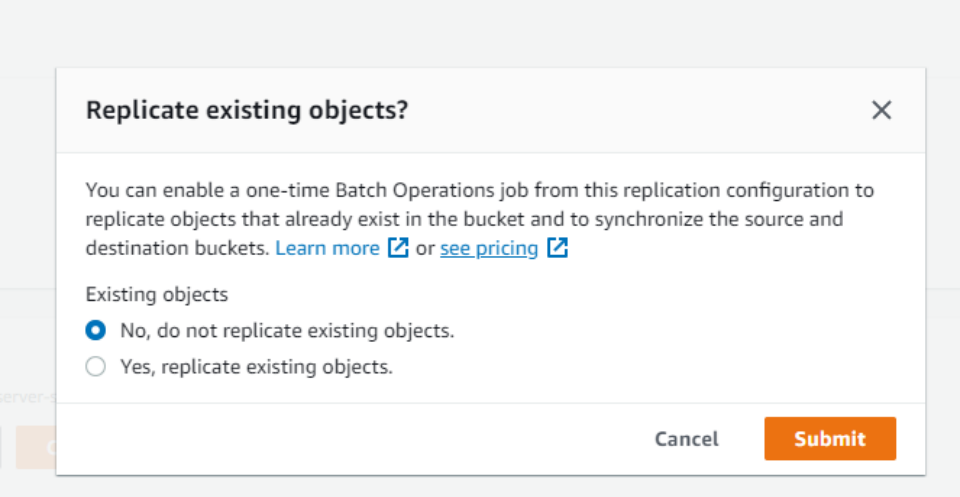


Create IAM role

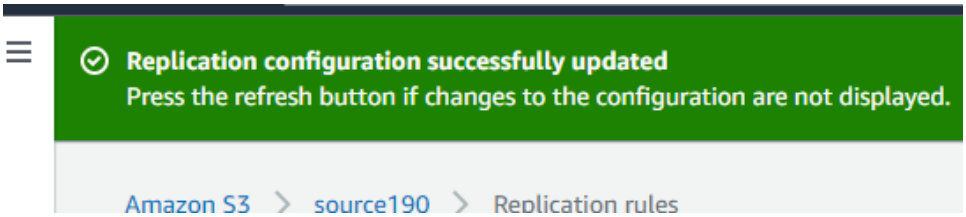
- If you want your S3 objects to be replicated within 15 minutes you need to check the "Replication Time Control (RTC) box. But you will be charged for this. So we will move forward without enabling that for now and click on save.



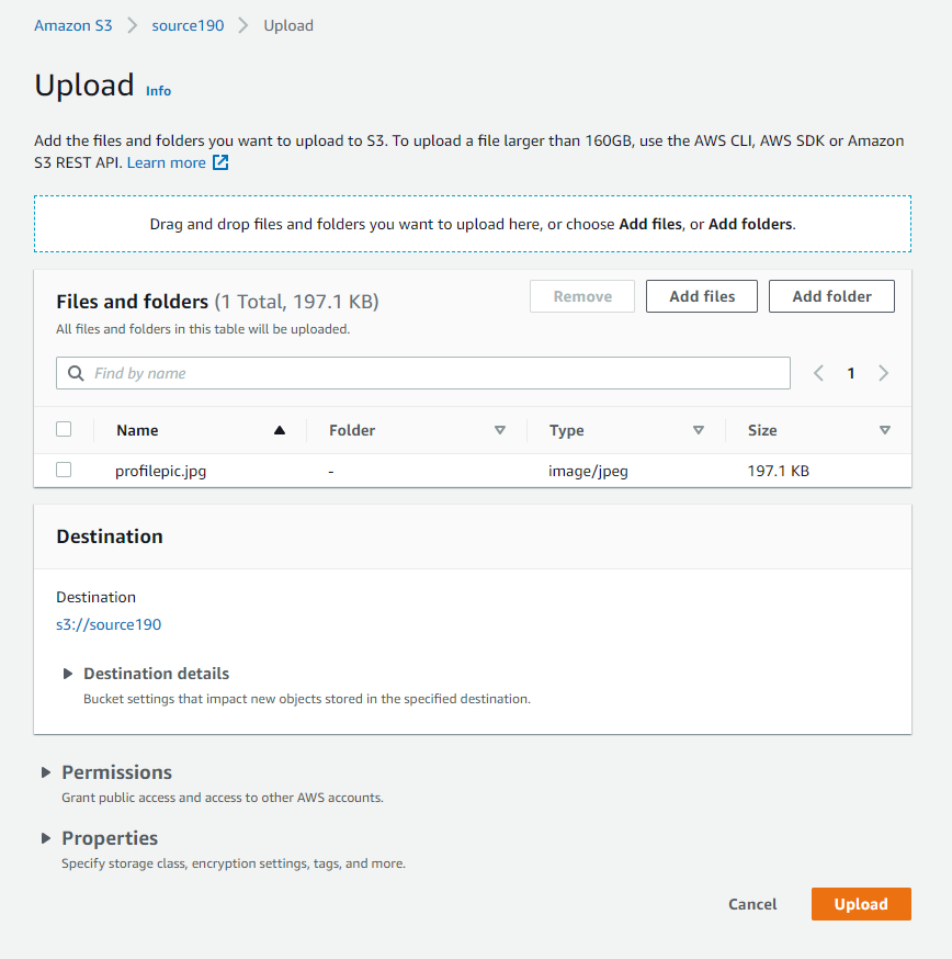
As soon as you click on save, a screen will pop up asking if you want to replicate existing objects in the S3 bucket. But that will incur charges so we will proceed without replicating existing objects and click on submit.



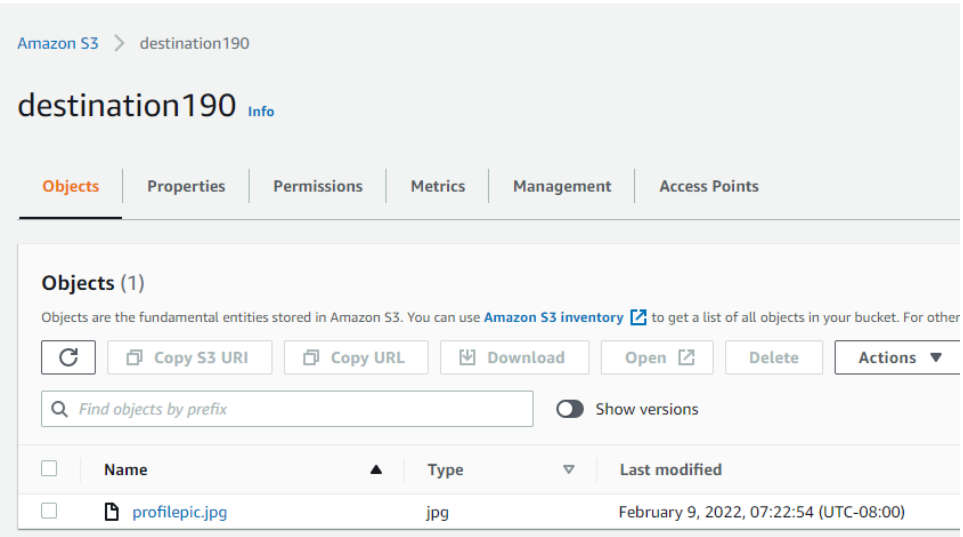
- After completing this setup you can see a screen saying “Replication configuration successfully updated”.



It’s time to test! Now go to the source bucket: source190 and upload a file.



Now head over to our destination bucket: destination190 to check if the uploaded file is replicated to our destination bucket. You can see that our uploaded file is successfully copied to the destination bucket:



Some important points about CRR:

For cross-region replication you must have:

- Source bucket and destination bucket in different regions (for the same region you can use the same region replication or SRR).
- Versioning is enabled in both the source as well as destination bucket.

When objects are replicated to a different region then:

- Object metadata, Access control list (ACL), and object tags are also replicated.
- The objects which were already present in the source bucket before setting up replication will not be replicated or copied to the destination bucket by default but you can perform a one-time batch operations job but that will incur additional charges.
- If your source bucket is acting as a destination bucket for another bucket or there are objects replicated in the source bucket from another bucket, then those objects will not be replicated to the destination bucket.

You can also enable bi-directional CRR by making the source bucket also the destination bucket for the destination bucket and vice versa.

Lastly, it is not necessary to have a destination bucket in the same account. AWS Cross-Region Replication can also be implemented in cross accounts (given that the owner of the source bucket have the permission to copy data in the destination bucket)