**AMAZON S3 SECURITY**

It is mandatory to secure our s3 bucket.so we use the below ways to secure our s3 bucket

1.**user based:**

This is the user based security in which we specify which user is allowed to do action on an s3 bucket.

There are IAM policies assigned which specify which api calls should be allowed for a specific user from iam.



The above is an iam policy which tells that the principle which is the root user is allowed to access our resource mybucket and the api calls allowed by him are getobject from s3 and putobject into s3.

2.**Resource based**

These are the resource based policies.

1.**BUCKET POLICIES:**bucket wide rules from the s3 console which allows cross account access to our bucket

**2.OBJECT ACCESS CONTROL LIST:**finer grain(can be disabled)

**3.BUCKET ACCESS CONTROL LIST:**less common and can be disabled.

Note:an iam principal can access an s3 object if:

1the iam permissions allow it or the resource policy allows it.

2.and there is no explicit deny

3.**ENCRYPTION:**we can secure our bucket by encrypting 

the objects using the encryption keys.

**S3 BUCKET POLICY**

We can see an s3 bucket policy a sthe one shown below:

Diagram

Description automatically generated with medium confidence

It says to allow all the users/principle to allow to perform action of getobject(make api call to retrieve data from bucket) from the resource..and everything inside the exaplmebucket.

**VARIOUS USE CASES OF A BUCKET POLICY**

**Graphical user interface, application

Description automatically generated**

Here a user from outside world want to access the content of our s3 bucket so we will set a bucket policy to allow public access on it.

Graphical user interface, application

Description automatically generated

Second we have an account user or an iam user who wants toacces our s3 bucket so we will attach an iam policy to it saying to allow access to an s3 bucket.

Graphical user interface, application

Description automatically generated

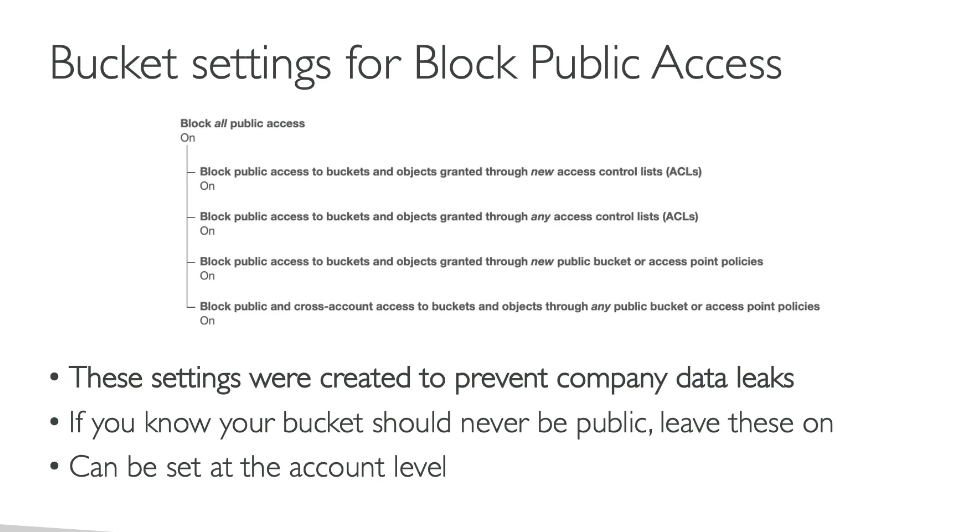
When there is an ec2 instance that needs to connect to our s3 bucket so in that case we will assign an iam role to the ec2 instance with the correct iam permissions to access the s3 bucket.

Graphical user interface, application

Description automatically generated

Suppose there is an iam user from some other aws account and he wants to get access to our s3 bucket so in that case we will assign an s3 bucket policy that will allow cross account access so that the cross account user can make api calls to our s3 bucket.

**BUCKET SETTINGS FORBLOCK PUBLIC ACCESS**

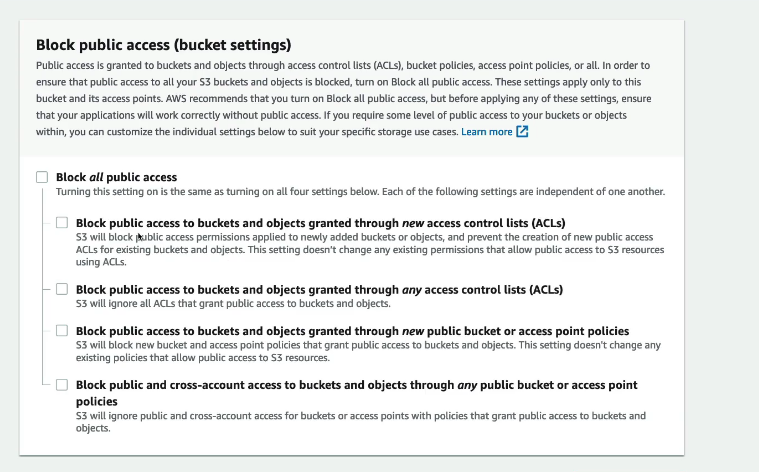
****

**There** is a setting to block the public access on our s3 bucket in order to prevent company data leaks.So if we want that our bucket should dnever be public then leave these settings on.

Secondly if you never want the content of your any s3 bucket to be accessed by any account so you can do settings at the account level for such.

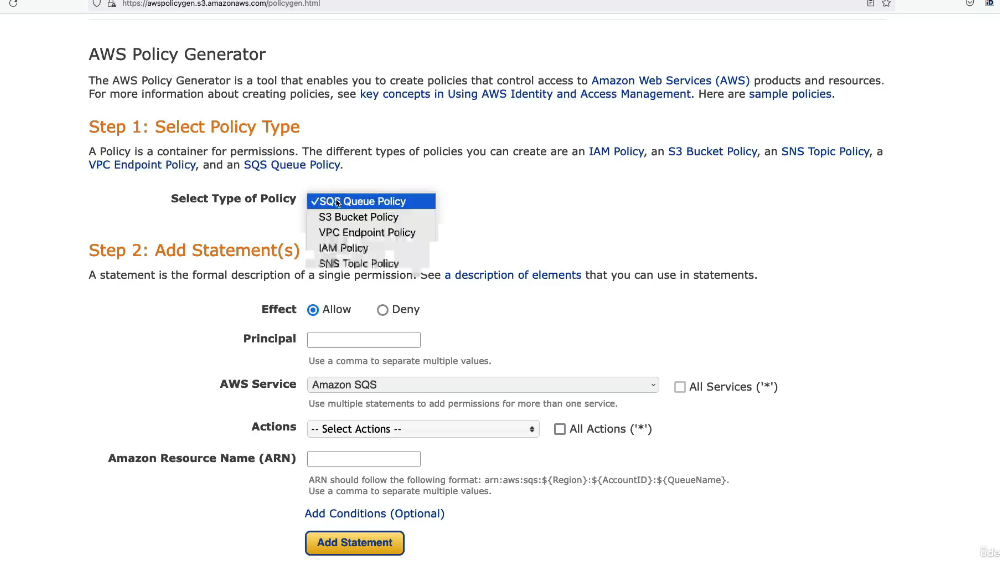
**HANDS ON SECURITY OF S3**

If you are not able to access the content of s3 bucket from outside then in that case you need to untick all the block statements.

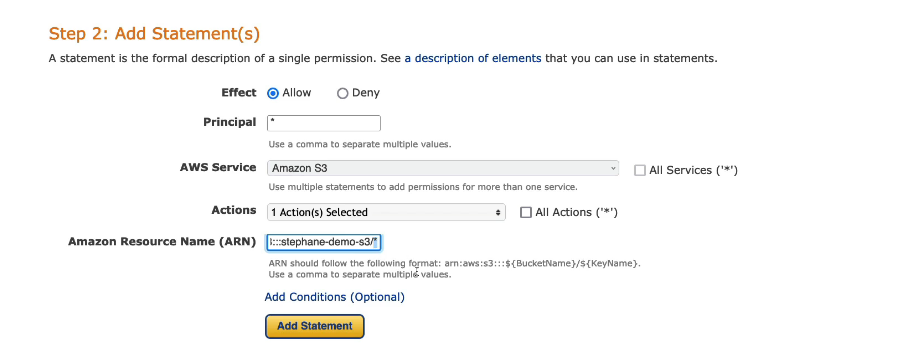


Now w e can create a bucket policy

We will use policy generator

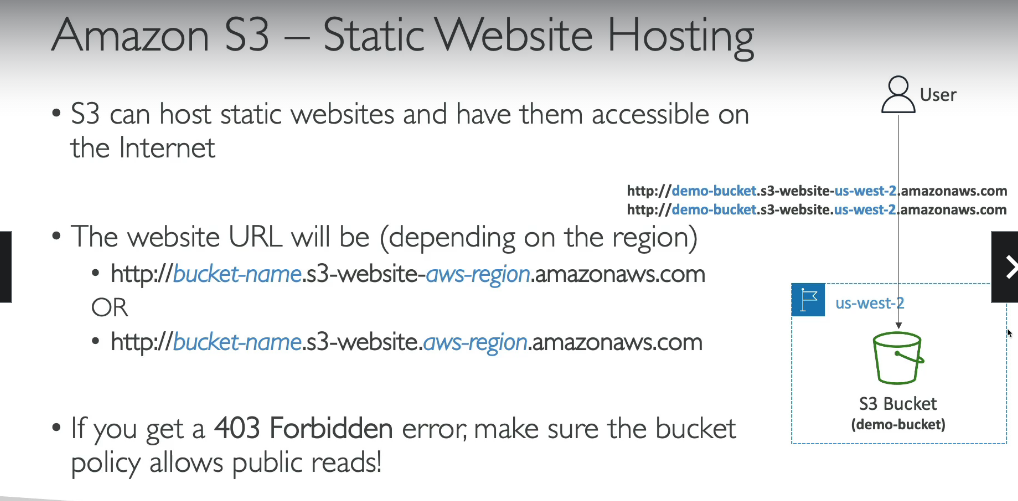


We add the below statement to it.we want to allow all the principals to access amazon s3 service to do the action of read object from the arn(resourcename of a bucket)



Now our bucket will be accessible from the public .

**AMAZON S3 STATIC WEBSITE HOSTING**



While accessing the website if you get a 403 forbidden then please make sure that the public read is not blocked

**HANDS ON**

1.go to your bucket



2.go to properties and scroll down to static website hosting

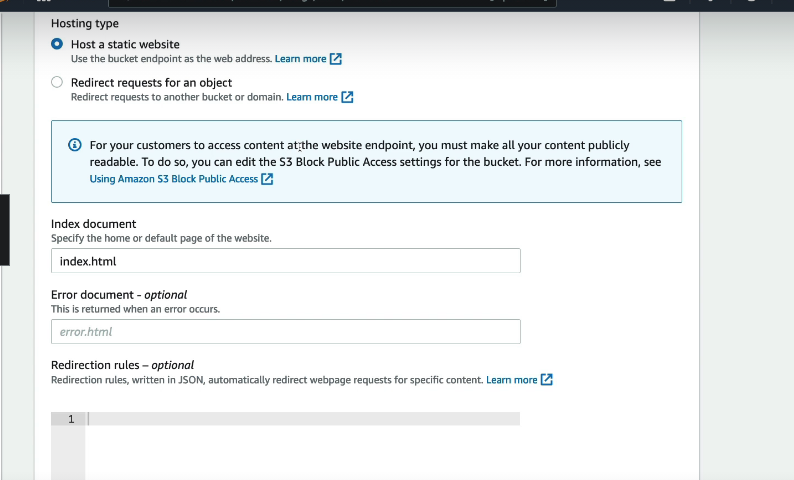
Graphical user interface, text, application

Description automatically generated

3.click on edit and enble static website hosting.

Now we want to access our website with an index page .

So we will go back and give our index.html file



Graphical user interface, text, application, email

Description automatically generated

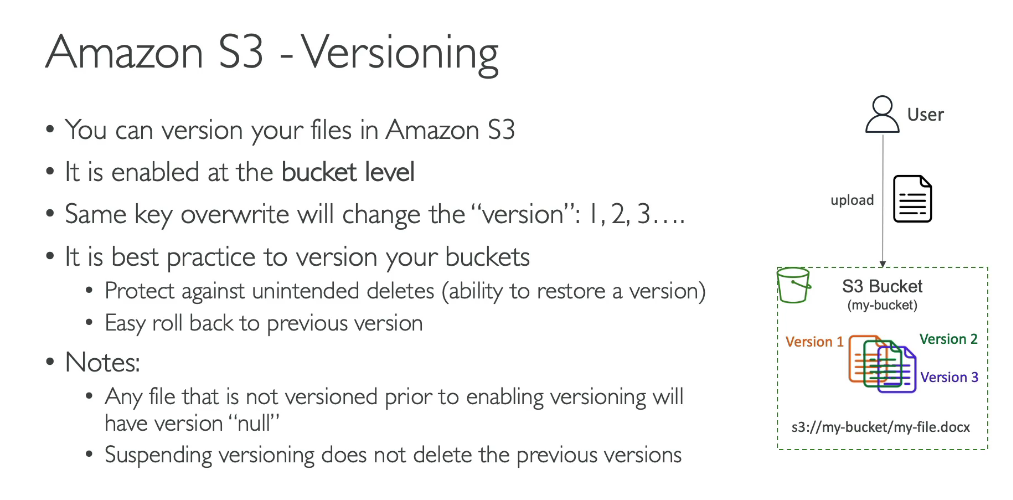
4.after all this we go to properties and we have a bucket website endpoint

Graphical user interface, text, application, email

Description automatically generated

So when we paste this in our browser we will be able to access our static website.

**S3 VERSIONING**

We need versoning in case we need 

**HANDS ON**

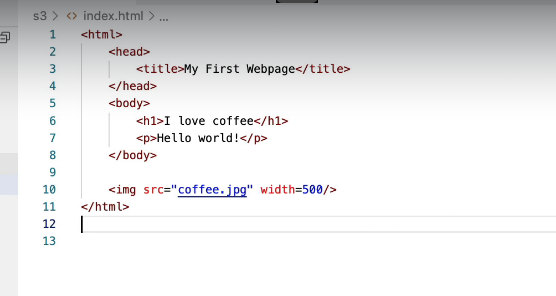
**1.**go to properties and bucket versioning setting and enable bucket versioning.

So any files we add will be versioned.

Graphical user interface, application, Teams

Description automatically generated

2.Now to demonstrate versioning we need to edit something into our html file as shown below:



We have chaged content of index.html and then save it

Graphical user interface, text, application, email

Description automatically generated

Now upload this file again.

After uploading

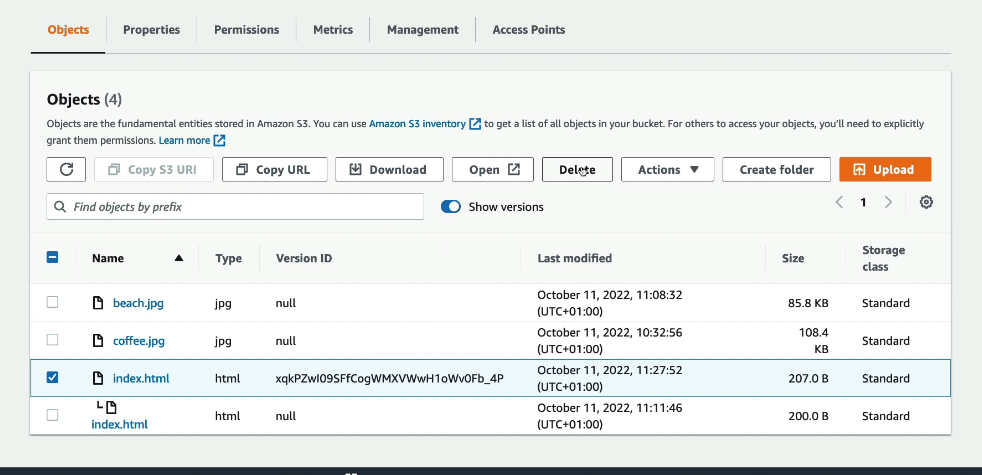
Graphical user interface, text, application, email

Description automatically generated

Here null version id denotes while we had no versions and the other is the one after we had versioning enabled.

3.Now if we want to roll back to our previous ste then we can delete the latest version

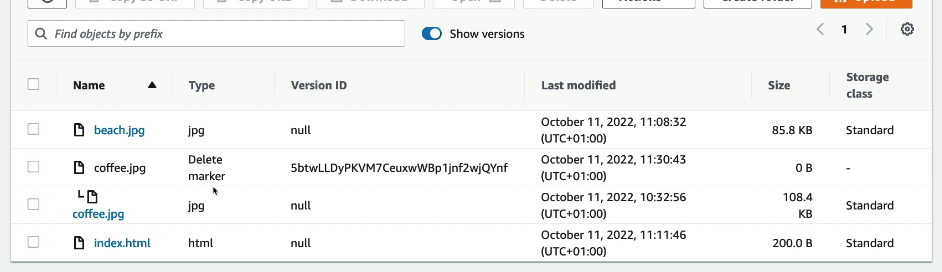
D



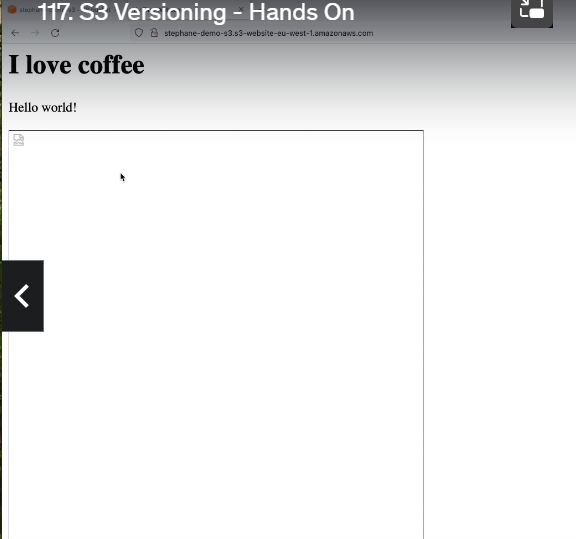
Deletion is a destructive option and cannot be undone.

**DELETING SOMETHING IN AN S3 BUCKET**

When we delete something from our bucket it does not gets deleted rather a delete marker is created for it and if we click on show versions we can see that with our eyes that there is a version maintained there.



But now if we go to our website we will see that there is no image



So if we want to get image back we need to dlete the delete marker

**S3 REPLICATION**

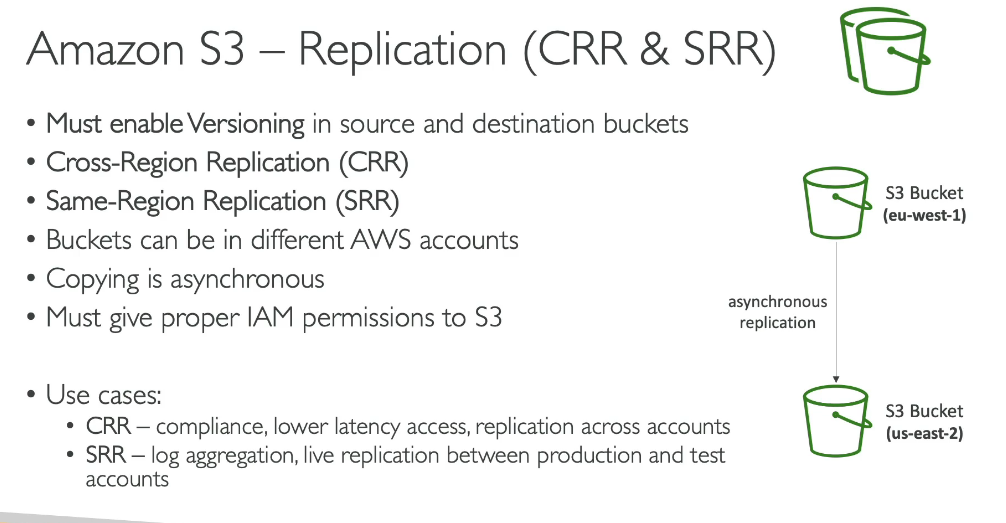
There are two kinds of amazon replication.:

1.CRR: it is for crosss region replication

2.SRR:It us for same region replication

So the idea is that we have an s3 bucket in one region an danother s3 bucket in another region and we want to setup an aasync replication in the two buckets

So we need to enable versioning in the se buckets.



**SOME IMPORTANT NOTES**

After we enable replication only new objects are replicated.Optionally we can replicate existing objects using s3 batch processing.This replicates existing objects and objects that failed replication.

**For delete operations**

We can replicate delete markers from source to target

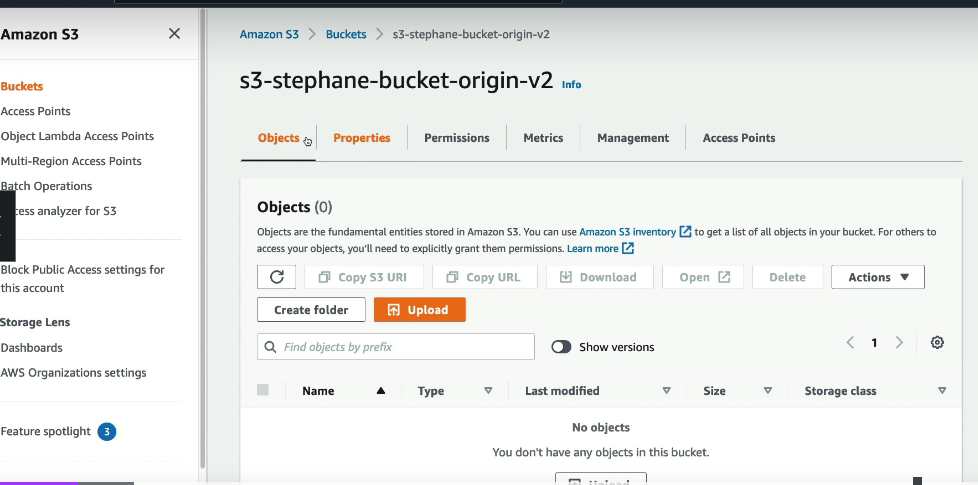
Deletion with version id are not replicated to avoid malicious deletes.

There is no chaining of replication

If bucket 1 has replication into bucket2 which has replication into bucket 3 then objects created in bucket 1 are not replicated to bucket3.

**REPLICATION HANDS ON**

First we need to create a bucket and enable versioning on it.

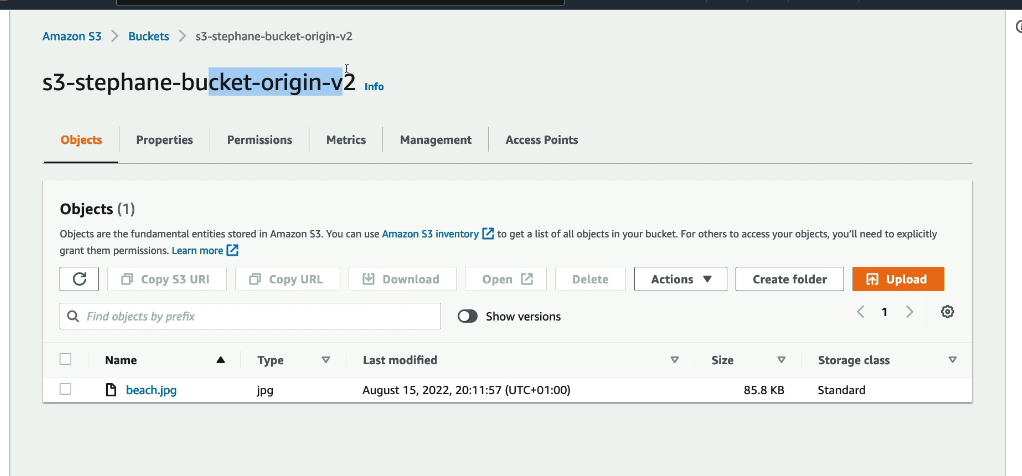


2.Now create anew bucket either in same region or in different region.

Graphical user interface, application

Description automatically generated

3.now we upload some files into our origin bucket.

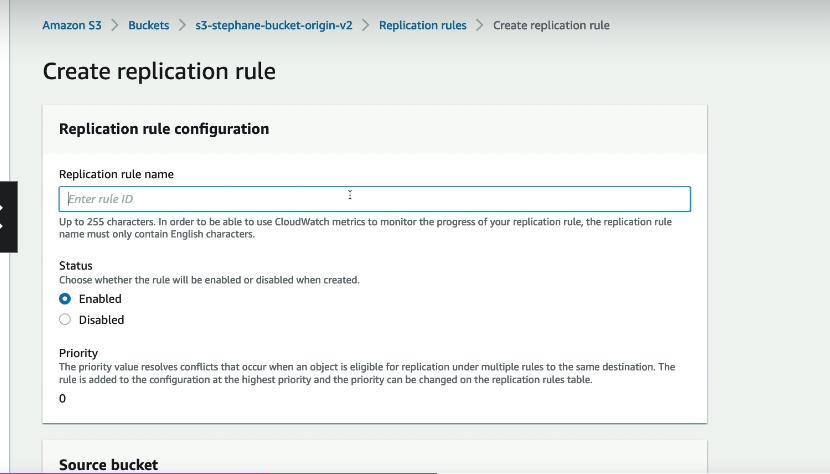


4.Now in our origin bucket we go under the management and set replication rules there.

Graphical user interface, text, application, email

Description automatically generated

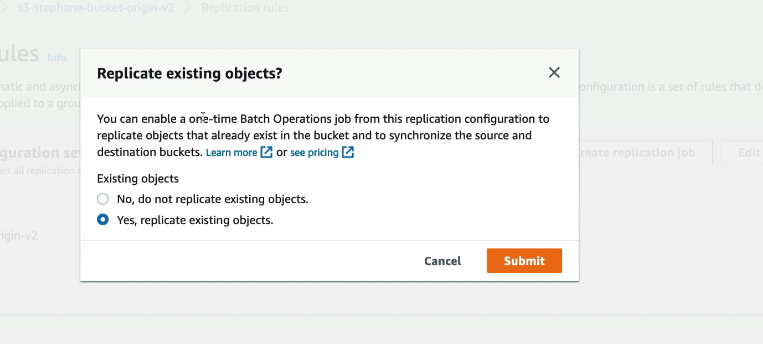
We create a replication rule.



We specify our other bucket

5.create new iam role.

6.now it will prompt if we need to replicate existing content or not or if we want to replicate from the moment we are setting this replication rule.

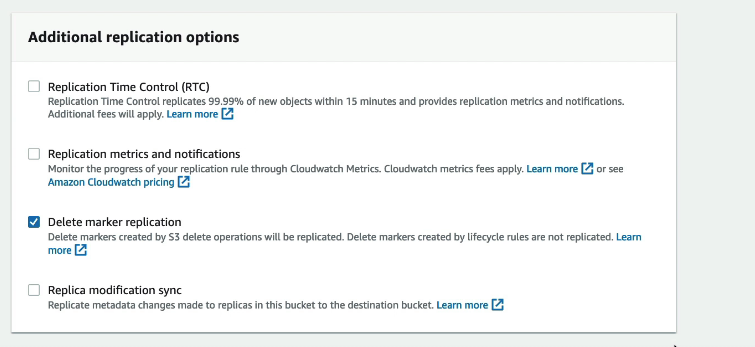


Thus if we say a yes to replicate existing objects then it will be meant to be a batch processing.

Now we go to our bucket and check that there is no replicated data and when we upload new data into our origin bucket then it will be replicated in the replicated bucket.

Version ids are also replicated.

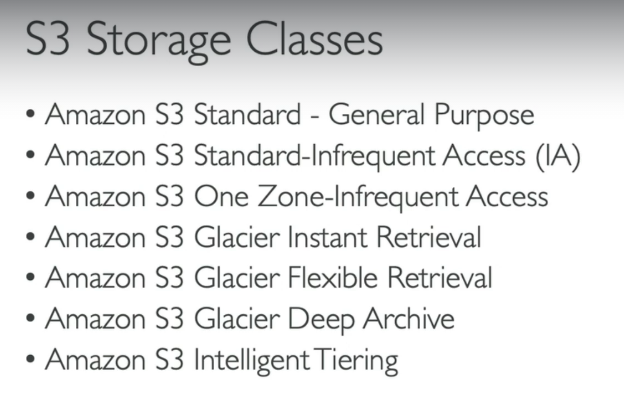
New versions are replicated.

**DELETE MARKER REP LICATION **

So now when we go in our origin bucket and delete some file then a delete marker will be created in origin.Markers can also be replicated.

Did not understand s3 delete replication.

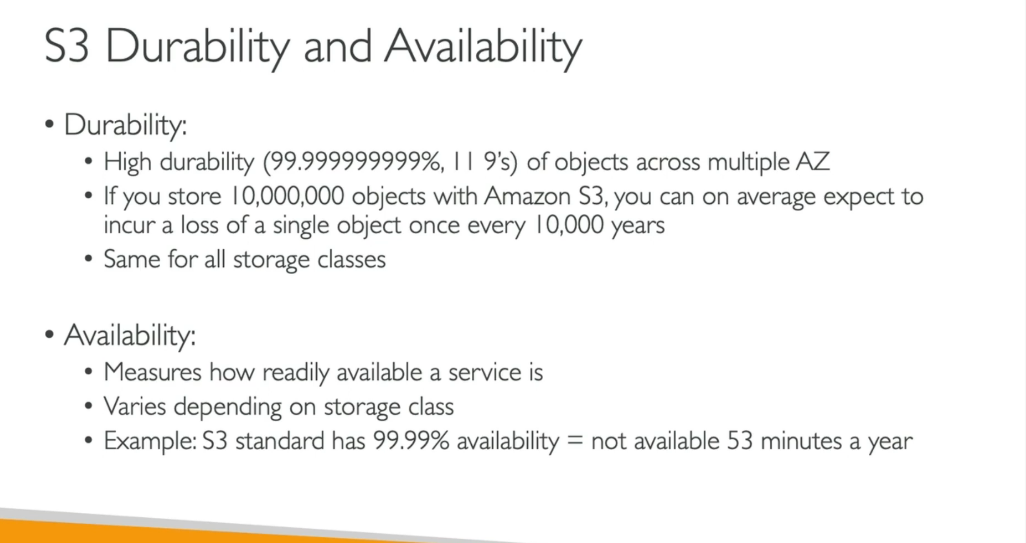
**S3 STORGE CLASSES**

****

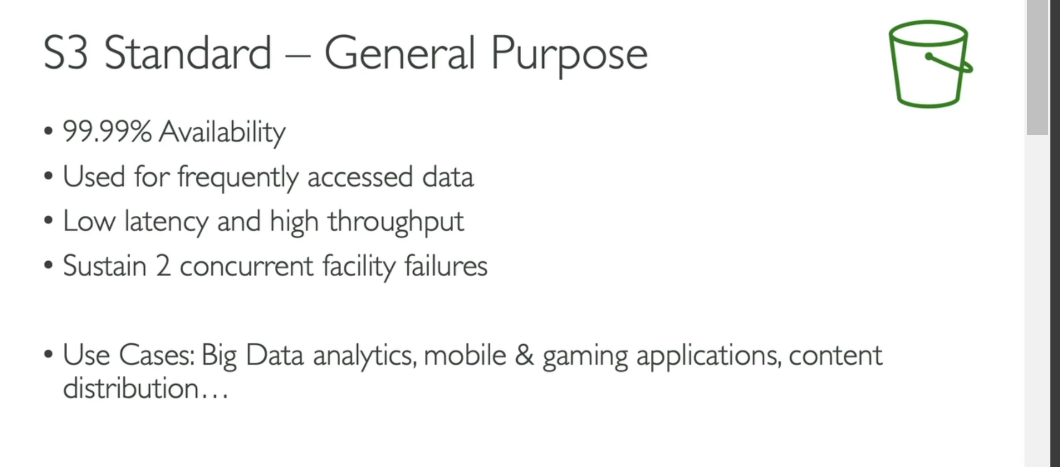
**You can move between classes manually or using s3 lifecycle configurations.**

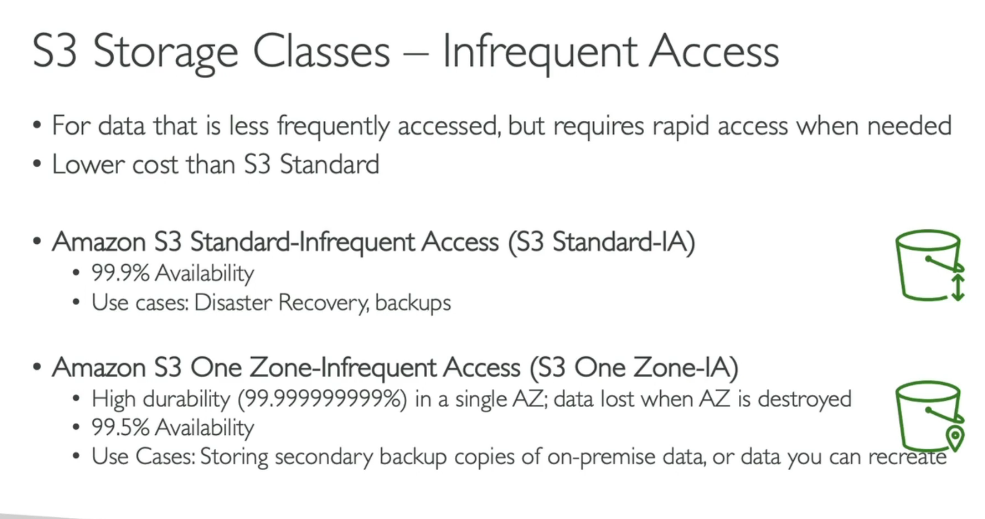
**DURABILITY AND AVAILABILITY**

**Durability** shows how many tikmes an object is going to be lost in amazon s3.

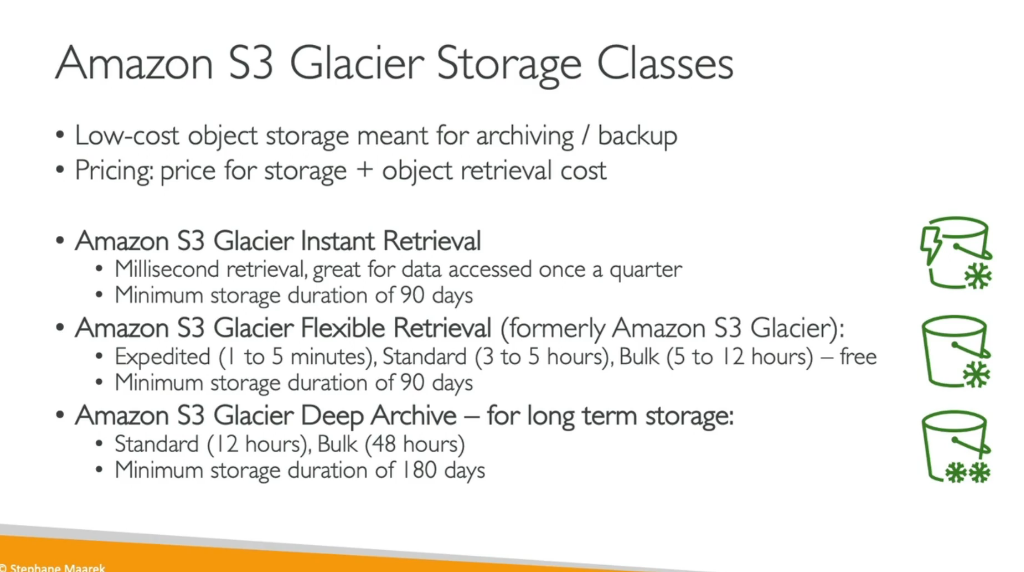


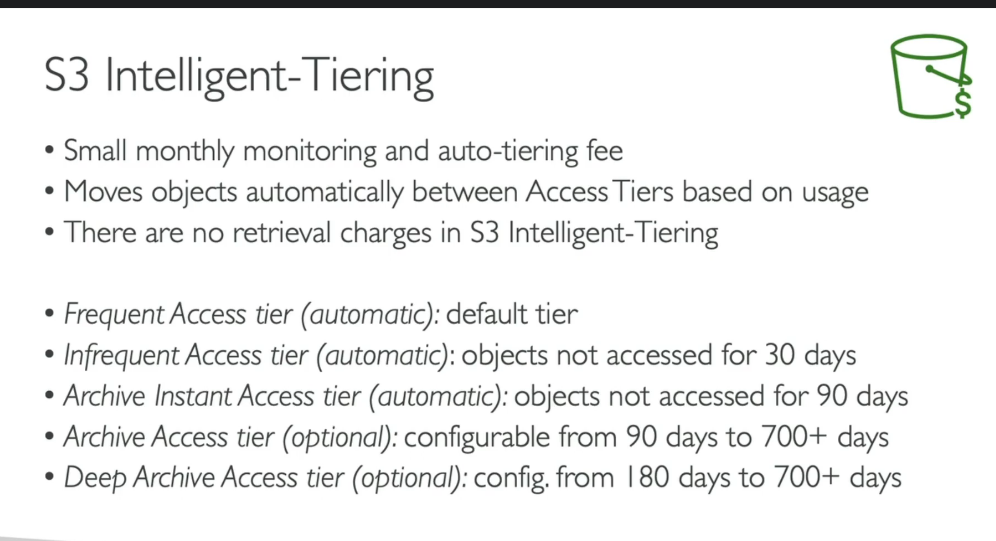
**GENERAL PURPOSE**

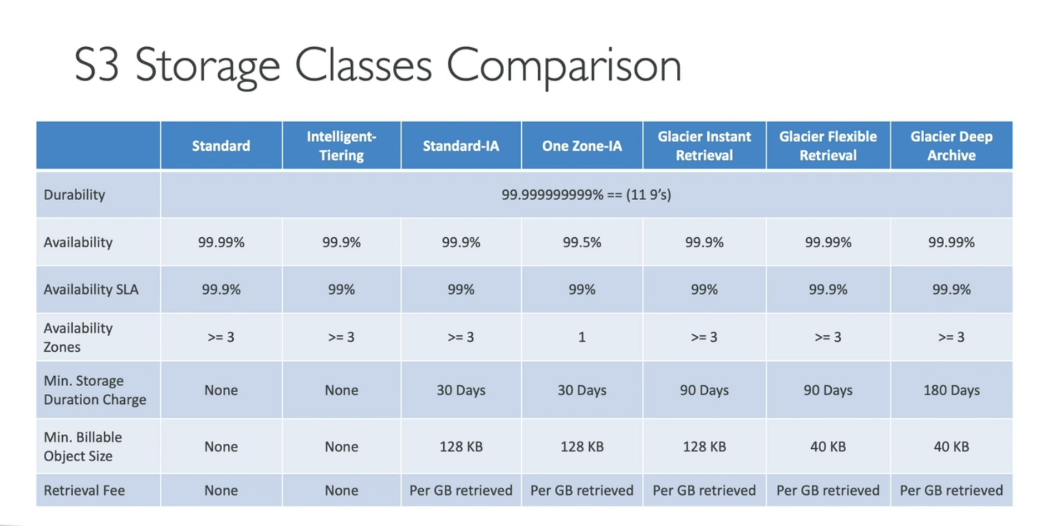
****

****

**GLACIER**

****

****

****

**Table

Description automatically generated**