Assignment-11-May-2022

1. What are the difference AWS storage services available in AWS?

: There are three types of cloud data storage:

object storage (S3)

file storage (EFS)

block storage (EBS)

Instance storage

2. Explain the difference between S3, EFS, EBS and Instance store?

:

* S3 is a storage facility accessible any where
* EBS is a device you can mount onto EC2
* EFS is a file system you can mount onto EC2

There are a lot more differences between these services:

**EFS is:**

* Generally Available (out of preview), but may not yet be available in your region
* Network filesystem (that means it may have bigger latency but it can be shared across several instances; even between regions)
* It is expensive compared to EBS (~10x more) but it gives extra features.
* It's a highly available service.
* It's a managed service
* You can attach the EFS storage to an EC2 Instance
* Can be accessed by multiple EC2 instances simultaneously
* Since 2016.dec.20 it's possible to attach your EFS storage directly to on-premise servers via Direct Connect. ()

**EBS is:**

* A block storage (so you need to format it). This means you are able to choose which type of file system you want.
* As it's a block storage, you can use Raid 1 (or 0 or 10) with multiple block storages
* It is really fast
* It is relatively cheap
* With the new announcements from Amazon, you can store up to 16TB data per storage on SSD-s.
* You can snapshot an EBS (while it's still running) for backup reasons
* But it only exists in a particular region. Although you can migrate it to another region, you cannot just access it across regions (only if you share it via the EC2; but that means you have a file server)
* You need an EC2 instance to attach it to
* New feature (2017.Feb.15): You can now increase volume size, adjust performance, or change the volume type while the volume is in use. You can continue to use your application while the change takes effect.

**S3 is:**

* An object store (not a file system).
* You can store files and "folders" but can't have locks, permissions etc like you would with a traditional file system
* This means, by default you can't just mount S3 and use it as your webserver
* But it's perfect for storing your images and videos for your website
* Great for short term archiving (e.g. a few weeks). It's good for long term archiving too, but Glacier is more cost efficient.
* Great for storing logs
* You can access the data from every region (extra costs may apply)
* Highly Available, Redundant. Basically data loss is not possible (99.999999999% durability, 99.9 uptime SLA)
* Much cheaper than EBS.
* You can serve the content directly to the internet, you can even have a full (static) website working direct from S3, without an EC2 instance

3. What are the different classes of S3 or category of s3?

: Types of S3 Storage Classes -

S3 Standard.

S3 Standard-IA.

S3 Intelligent-Tiering.

S3 One Zone-IA.

S3 Glacier.

S3 Glacier Deep Archive.

S3 Outposts.

**S3 Standard**

S3 Standard is the default storage class if none of the storage class is specified during upload.

It is ideal for frequently accessed data because it provides low latency and high availability.

It has a wide range of use cases from cloud applications and web services, websites hosting, big data analytics, mobile gaming, and content distribution. It is the most expensive storage class among all others.

**Key Points**:

* High Availability and low latency
* Data is stored in multiple locations. So it is resilient against events that affect an entire Availability Zone
* The durability of 99.999999999% and availability of 99.99% availability over a given year
* Most expensive storage class among all others.

**S3 Standard-IA**

S3 Standard-Infrequent Access is optimized for long-lived and less frequently accessed data but requires rapid access whenever required.

Similar to S3 Standard, it also offers high durability, low latency, and high throughput but has a low per GB storage price and per GB retrieval fee.

The S3 Standard-IA is ideal for backups, long-term storage, and as a data store for disaster recovery

**Key Points**:

* High Availability and Low Latency (Same as S3 Standard)
* Offers greater availability and resiliency than the OneZone-IA storage.
* The durability of 99.999999999% and availability of 99.99% availability over a given year
* Less expensive than S3 Standard storage but you will be charged a retrieval fee hence suitable for infrequently accessed data.

**S3 Intelligent-Tiering**

S3 Intelligent-Tiering optimizes costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. It moves objects that have not been accessed for 30 consecutive days to the infrequent access tier.

If the object is accessed then it is automatically moved back to the frequent access tier. No retrieval fees or additional tiering fees are using the S3 Intelligent-Tiering storage class.

It is ideal for storing long-lived data where the access patterns are unknown.

**Key Points:**

* Low latency and high throughput performance
* Automatically moves the data between two access tiers. (Infrequent Access and Frequent Access)
* The durability of 99.999999999% and availability of 99.99% availability over a given year
* Small monthly monitoring and auto-tiering fee

**S3 One Zone-IA**

S3 One Zone- Infrequent Access is for the data that is accessed less frequently but available for millisecond access. Since the other S3 storage class store data in a minimum of 3 Availability Zones (AZ), S3 One Zone-IA stores data in only one AZ which makes the costs 20% lesser than the S3 Standard-IA.

It offers the same high durability, high throughput, and low latency. It can be considered as a good choice for storing secondary backup copies or easily re-creatable data if an AZ fails.

**Key Points**:

* Low Latency and High throughput performance
* The durability of 99.999999999% and availability of 99.5% availability over a given year
* Data will be lost if the Availability Zone where the data is stored is destroyed.
* Suitable for larger objects greater than 128 KB kept for at least 30 days (charged minimum for 30 days)

**S3 Glacier**

S3 Glacier is a low-cost storage class for data archiving where data access is infrequent. It provides a configurable retrieval time for the data from minutes to hours.

This storage class uses a very low-cost Glacier storage service but the objects are still managed through S3.

**Key Points**:

* Low-cost design for long-term archiving
* Data will be available in case of entire Availability Zone destruction
* The durability of 99.999999999% and availability of 99.9% availability over a given year
* It has a minimum storage duration period of 90 days.

**S3 Glacier Deep Archive**

The S3 Glacier Deep Archive provides the lowest-cost storage class and supports long-term retention and digital preservation for data that may be accessed only once or twice in a year.

It is ideal for those industries which store data for 5-10 years or longer like healthcare, finance, etc. It can also be used for backup and disaster recovery.

**Key Points**:

* Lowest cost storage option in S3
* The durability of 99.999999999% and availability of 99.9% availability over a given year
* Retrieval costs can be reduced by using bulk retrieval
* It has a minimum storage duration period of 180 days

**S3 Outposts**

S3 on Outposts provides object storage to our on-premises AWS outposts environment.

S3 on Outposts makes it easy to store, retrieve, secure, control access, tag, and report on the data. It is ideal for workloads with local data residency requirements, and to satisfy demanding performance needs by keeping data close to on-premises.

**Key Points**:

* S3 Object compatibility and bucket management is through S3 SDK
* For durable and redundant storage of data on Outposts
* S3 on Outposts will give users 48TB or 96TB of S3 storage capacity, with up 100 buckets on each Outpost.
* Hope the blog will help you in creating & understanding the concept behind an S3 bucket on AWS. Now it’s your turn to post your doubts in the comment section.

4. How to host a static website in S3? (steps with screen shots)

: configure an Amazon S3 bucket to function like a website.

**Topics**

Step 1: Create a bucket

Step 2: Enable static website hosting

Step 3: Edit Block Public Access settings

Step 4: Add a bucket policy that makes your bucket content publicly available

Step 5: Configure an index document

Step 6: Configure an error document

Step 7: Test your website endpoint

Step 8: Clean up

Step 1: **Create a bucket**

The following instructions provide an overview of how to create your buckets for website hosting. For detailed, step-by-step instructions on creating a bucket, see Creating a bucket.

**To create a bucket**

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose Create bucket.
3. Enter the Bucket name (for example, example.com).
4. Choose the Region where you want to create the bucket.
5. Choose a Region that is geographically close to you to minimize latency and costs, or to address regulatory requirements. The Region that you choose determines your Amazon S3 website endpoint. For more information, see Website endpoints.
6. To accept the default settings and create the bucket, choose Create.

Step 2: **Enable static website hosting**

After you create a bucket, you can enable static website hosting for your bucket. You can create a new bucket or use an existing bucket.

**To enable static website hosting**

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets list, choose the name of the bucket that you want to enable static website hosting for.
3. Choose Properties.
4. Under Static website hosting, choose Edit.
5. Choose Use this bucket to host a website.
6. Under Static website hosting, choose Enable.
7. In Index document, enter the file name of the index document, typically index.html.

* The index document name is case sensitive and must exactly match the file name of the HTML index document that you plan to upload to your S3 bucket. When you configure a bucket for website hosting, you must specify an index document.
* Amazon S3 returns this index document when requests are made to the root domain or any of the subfolders. For more information, see Configuring an index document.

1. To provide your own custom error document for 4XX class errors, in Error document, enter the custom error document file name.

* The error document name is case sensitive and must exactly match the file name of the HTML error document that you plan to upload to your S3 bucket.
* If you don't specify a custom error document and an error occurs, Amazon S3 returns a default HTML error document. For more information, see Configuring a custom error document.

1. (Optional) If you want to specify advanced redirection rules, in Redirection rules, enter JSON to describe the rules.

For example

you can conditionally route requests according to specific object key names or prefixes in the request. For more information, see Configure redirection rules to use advanced conditional redirects.

1. Choose Save changes.

* Amazon S3 enables static website hosting for your bucket. At the bottom of the page, under Static website hosting, you see the website endpoint for your bucket.

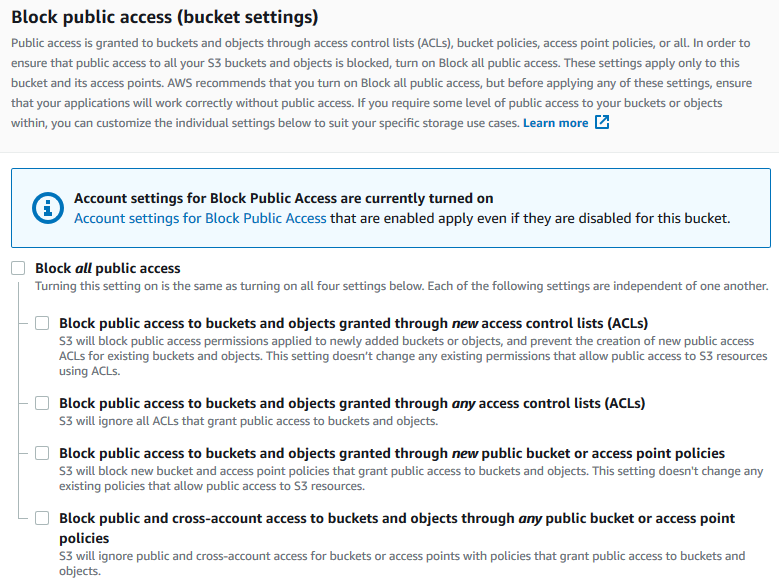
1. Under Static website hosting, note the Endpoint.

* The Endpoint is the Amazon S3 website endpoint for your bucket. After you finish configuring your bucket as a static website, you can use this endpoint to test your website.

Step 3: **Edit Block Public Access settings**

By default, Amazon S3 blocks public access to your account and buckets. If you want to use a bucket to host a static website, you can use these steps to edit your block public access settings.

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose the name of the bucket that you have configured as a static website.
3. Choose Permissions.
4. Under Block public access (bucket settings), choose Edit.
5. Clear Block all public access, and choose Save changes.



Amazon S3 turns off Block Public Access settings for your bucket. To create a public, static website, you might also have to [edit the Block Public Access settings](https://docs.aws.amazon.com/AmazonS3/latest/user-guide/block-public-access-account.html) for your account before adding a bucket policy. If account settings for Block Public Access are currently turned on, you see a note under **Block public access (bucket settings)**.

Step 4: **Add a bucket policy that makes your bucket content publicly available**

After you edit S3 Block Public Access settings, you can add a bucket policy to grant public read access to your bucket. When you grant public read access, anyone on the internet can access your bucket.

1. Under **Buckets**, choose the name of your bucket.
2. Choose **Permissions**.
3. Under **Bucket Policy**, choose **Edit**.
4. To grant public read access for your website, copy the following bucket policy, and paste it in the **Bucket policy editor**.
5. {
6. "Version": "2012-10-17",
7. "Statement": [
8. {
9. "Sid": "PublicReadGetObject",
10. "Effect": "Allow",
11. "Principal": "\*",
12. "Action": [
13. "s3:GetObject"
14. ],
15. "Resource": [
16. "arn:aws:s3:::*Bucket-Name*/\*"
17. ]
18. }
19. ]
20. }
21. Update the Resource to your bucket name.

In the preceding example bucket policy, *Bucket-Name* is a placeholder for the bucket name.

To use this bucket policy with your own bucket, you must update this name to match your bucket name.

1. Choose **Save changes**.

A message appears indicating that the bucket policy has been successfully added.

If you see an error that says Policy has invalid resource, confirm that the bucket name in the bucket policy matches your bucket name.

For information about adding a bucket policy, see [How do I add an S3 bucket policy?](https://docs.aws.amazon.com/AmazonS3/latest/user-guide/add-bucket-policy.html)

If you get an error message and cannot save the bucket policy, check your account and bucket Block Public Access settings to confirm that you allow public access to the bucket.

Step 5: **Configure an index document**

When you enable static website hosting for your bucket, you enter the name of the index document (for example, **index.html**).

After you enable static website hosting for the bucket, you upload an HTML file with this index document name to your bucket.

**To configure the index document**

1. Create an index.html file.

If you don't have an index.html file, you can use the following HTML to create one:

<html xmlns="http://www.w3.org/1999/xhtml" >

<head>

<title>My Website Home Page</title>

</head>

<body>

<h1>Welcome to my website</h1>

<p>Now hosted on Amazon S3!</p>

</body>

</html>

1. Save the index file locally.

The index document file name must exactly match the index document name that you enter in the **Static website hosting** dialog box. The index document name is case sensitive.

For example, if you enter index.html for the **Index document** name in the **Static website hosting** dialog box, your index document file name must also be index.html and not Index.html.

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Buckets** list, choose the name of the bucket that you want to use to host a static website.
3. Enable static website hosting for your bucket, and enter the exact name of your index document (for example, index.html). For more information, see [Enabling website hosting](https://docs.aws.amazon.com/AmazonS3/latest/userguide/EnableWebsiteHosting.html).
4. To upload the index document to your bucket, do one of the following:
   * Drag and drop the index file into the console bucket listing.
   * Choose **Upload**, and follow the prompts to choose and upload the index file.

For step-by-step instructions, see [Uploading objects](https://docs.aws.amazon.com/AmazonS3/latest/userguide/upload-objects.html).

1. (Optional) Upload other website content to your bucket.

Step 6: **Configure an error document**

When you enable static website hosting for your bucket, you enter the name of the error document (for example, **404.html**).

After you enable static website hosting for the bucket, you upload an HTML file with this error document name to your bucket.

**To configure an error document**

1. Create an error document, for example 404.html.
2. Save the error document file locally.

The error document name is case sensitive and must exactly match the name that you enter when you enable static website hosting. For example, if you enter 404.html for the **Error document** name in the **Static website hosting** dialog box, your error document file name must also be 404.html.

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Buckets** list, choose the name of the bucket that you want to use to host a static website.
3. Enable static website hosting for your bucket, and enter the exact name of your error document (for example, 404.html). For more information, see [Enabling website hosting](https://docs.aws.amazon.com/AmazonS3/latest/userguide/EnableWebsiteHosting.html).

After enabling static website hosting, proceed to step 6.

1. To upload the error document to your bucket, do one of the following:
   * Drag and drop the error document file into the console bucket listing.
   * Choose **Upload**, and follow the prompts to choose and upload the index file.

For step-by-step instructions, see [Uploading objects](https://docs.aws.amazon.com/AmazonS3/latest/userguide/upload-objects.html).

Step 7: **Test your website endpoint**

After you configure static website hosting for your bucket, you can test your website endpoint.

1. Under **Buckets**, choose the name of your bucket.
2. Choose **Properties**.
3. At the bottom of the page, under **Static website hosting**, choose your **Bucket website endpoint**.

Your index document opens in a separate browser window.

You now have a website hosted on Amazon S3.

Step 8: **Clean up**

If you created your static website only as a learning exercise, delete the AWS resources that you allocated so that you no longer accrue charges.

After you delete your AWS resources, your website is no longer available. For more information, see [Deleting a bucket](https://docs.aws.amazon.com/AmazonS3/latest/userguide/delete-bucket.html).

1. How to create a bucket policy by using generate policy?

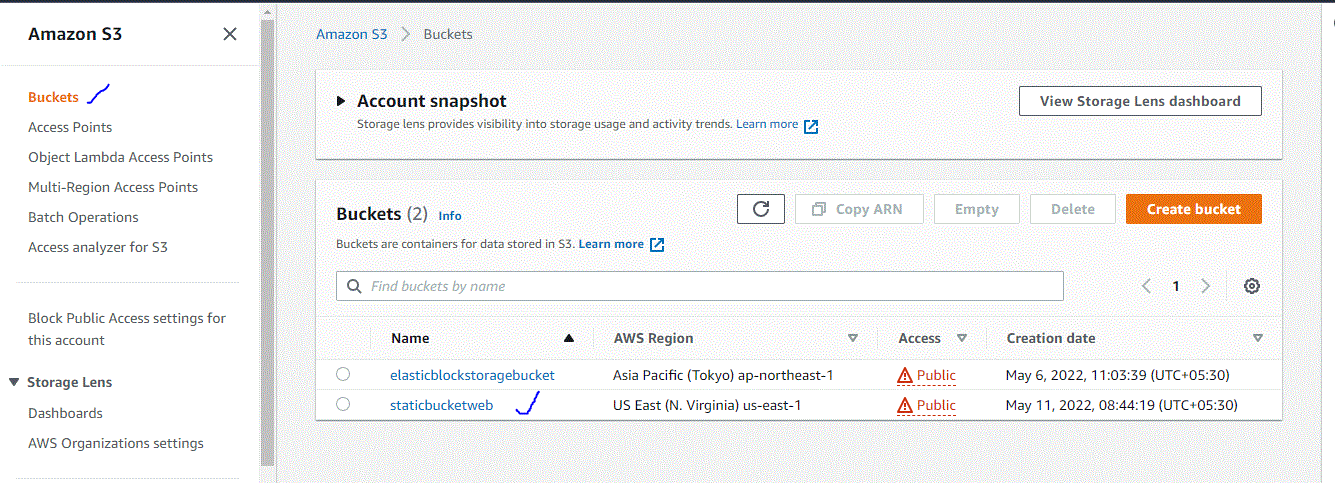
: **To create or edit a bucket policy**

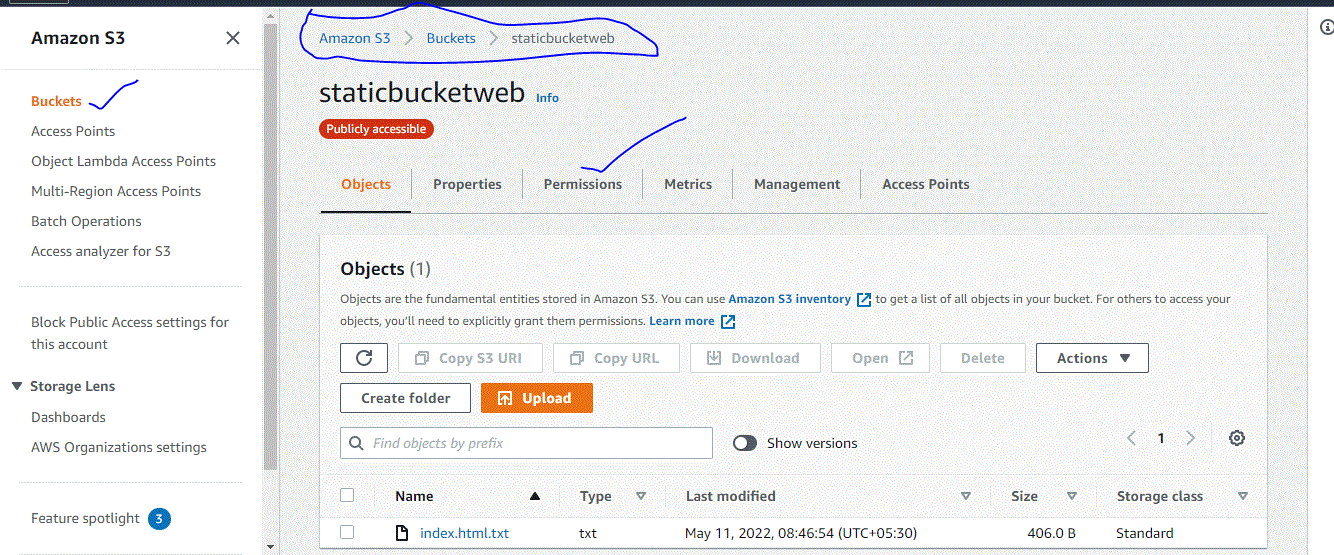
1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Buckets** list, choose the name of the bucket that you want to create a bucket policy for or whose bucket policy you want to edit.
3. Choose **Permissions**.
4. Under **Bucket policy**, choose **Edit**. This opens the Edit bucket policy page.
5. On the **Edit bucket policy**page, explore **Policy examples** in the *Amazon S3 User Guide*, choose **Policy generator** to generate a policy automatically, or edit the JSON in the **Policy** section.

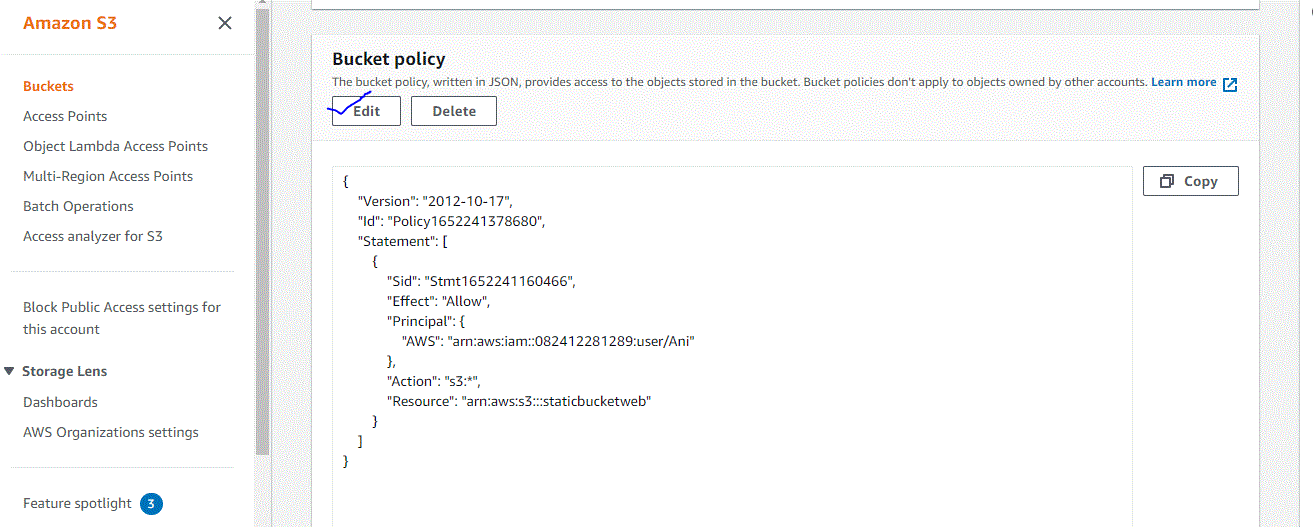
If you choose **Policy generator**, the AWS Policy Generator opens in a new window:

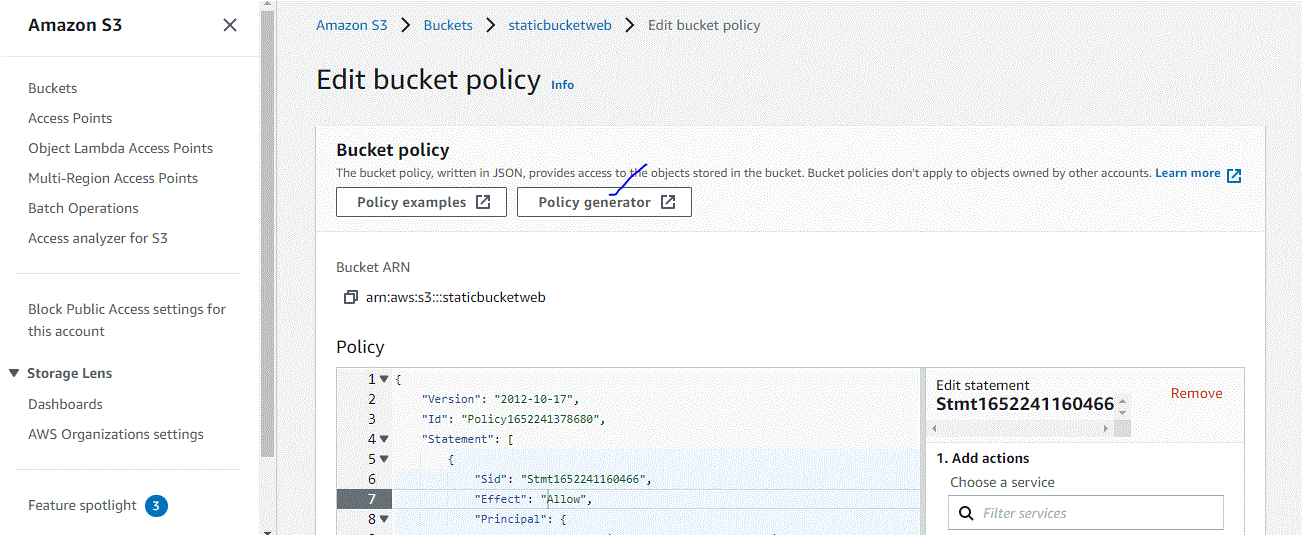
* + On the **AWS Policy Generator** page, in **Select Type of Policy**, choose **S3 Bucket Policy**.
  + Add a statement by entering the information in the provided fields, and then choose **Add Statement**. Repeat for as many statements as you would like to add. For more information about these fields, see the [IAM JSON policy elements reference](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements.html) in the *IAM User Guide*.

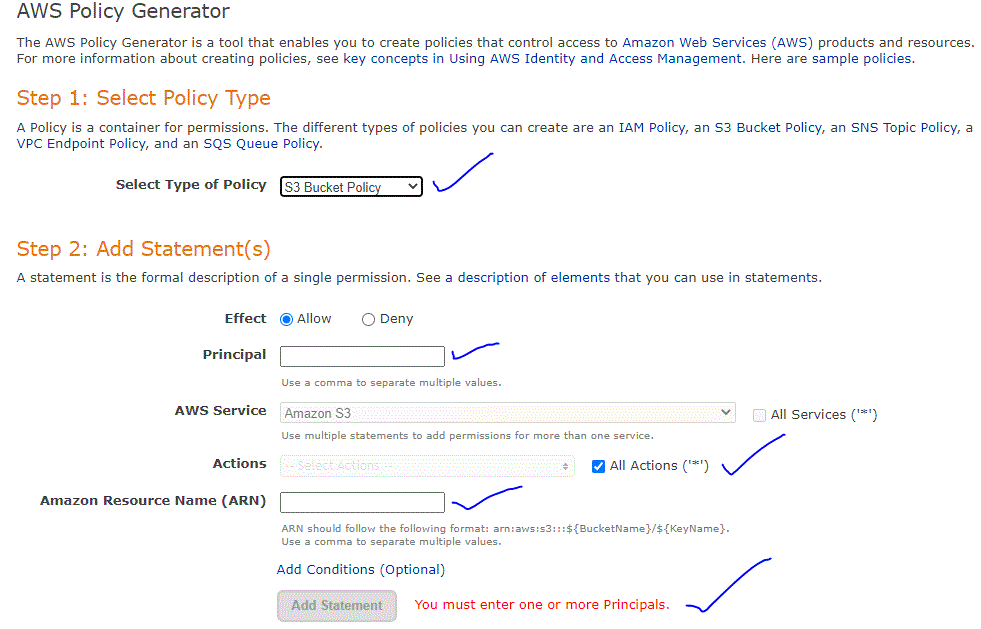
1. After you finish adding statements, choose **Generate Policy**.
   * Copy the generated policy text, choose **Close**, and return to the **Edit bucket policy** page in the Amazon S3 console.
2. In the **Policy** box, edit the existing policy or paste the bucket policy from the Policy generator. Make sure to resolve security warnings, errors, general warnings, and suggestions before you save your policy.
3. (Optional) Preview how your new policy affects public and cross-account access to your resource. Before you save your policy, you can check whether it introduces new IAM Access Analyzer findings or resolves existing findings. If you don’t see an active analyzer, [create an account analyzer](https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-getting-started.html#access-analyzer-enabling) in IAM Access Analyzer. For more information, see [Preview access](https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-access-preview.html) in the *IAM User Guide*.
4. Choose **Save changes**, which returns you to the Bucket Permissions page.

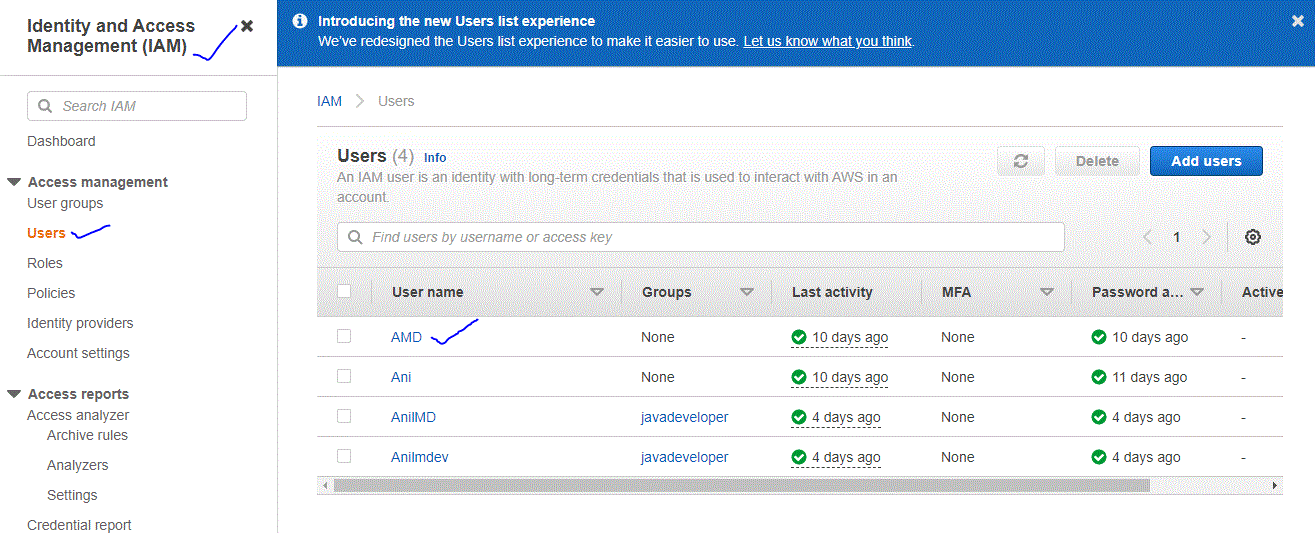


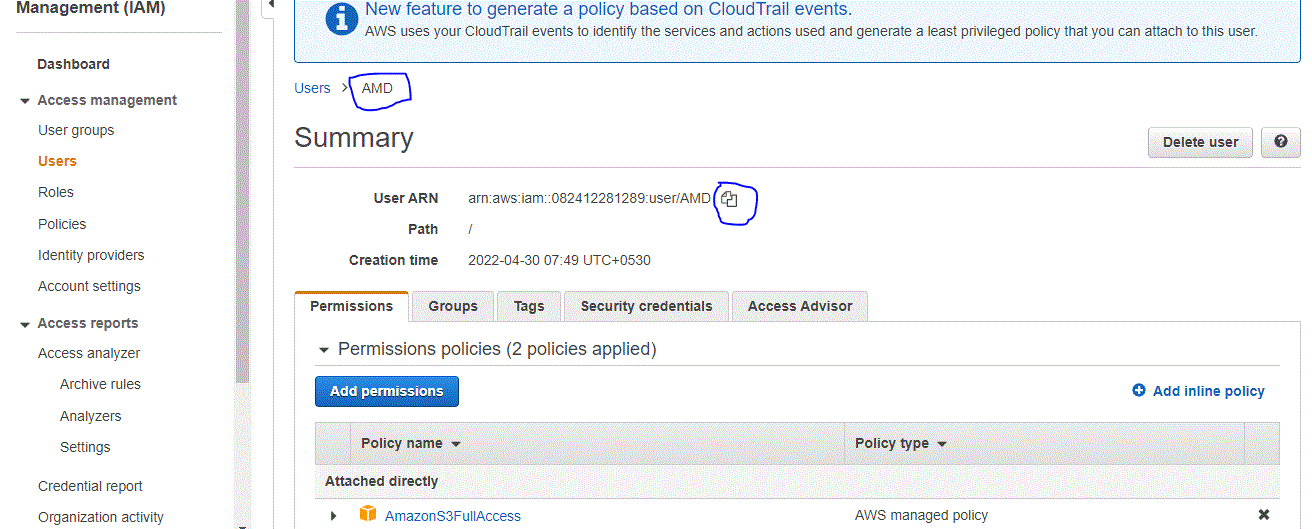


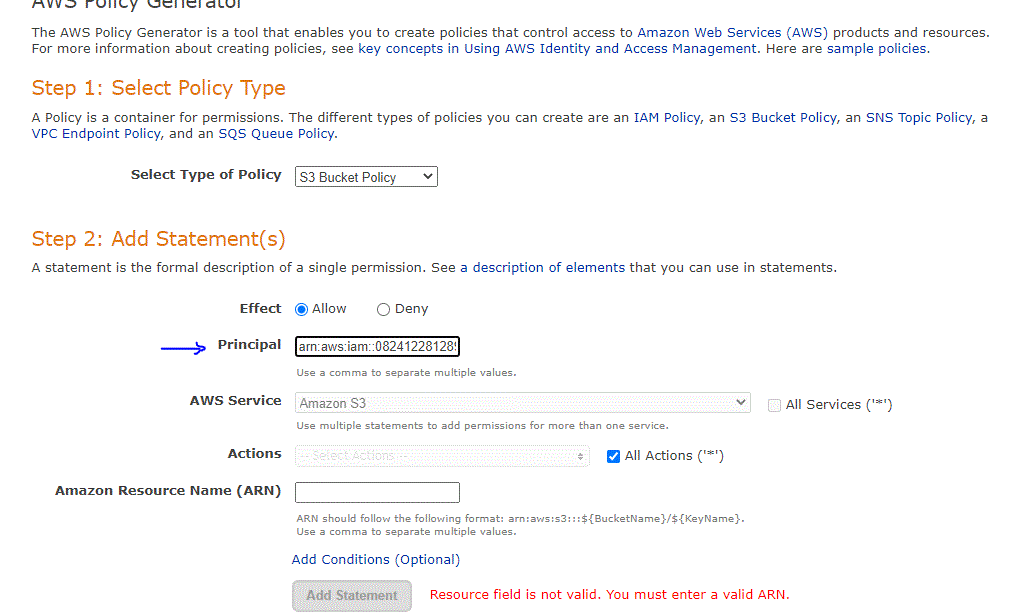


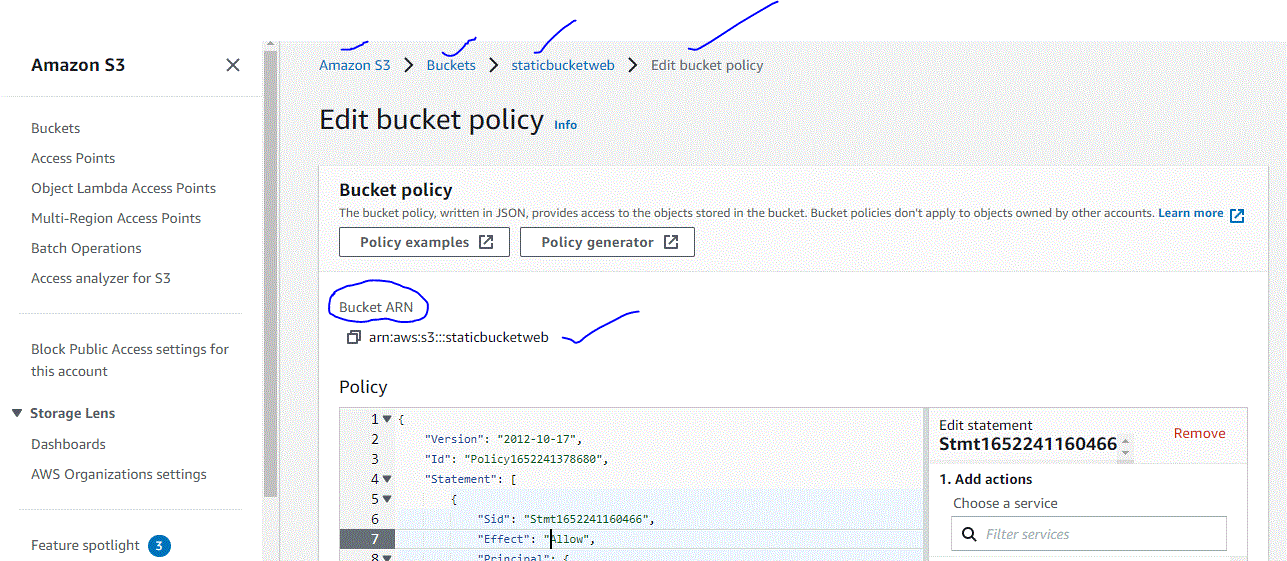


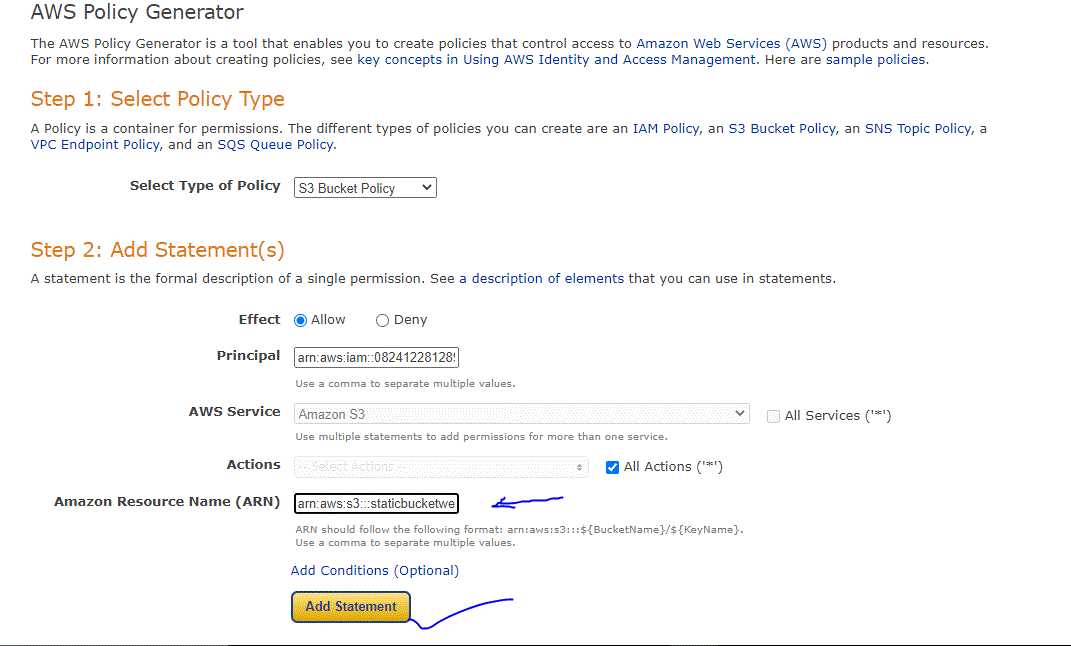


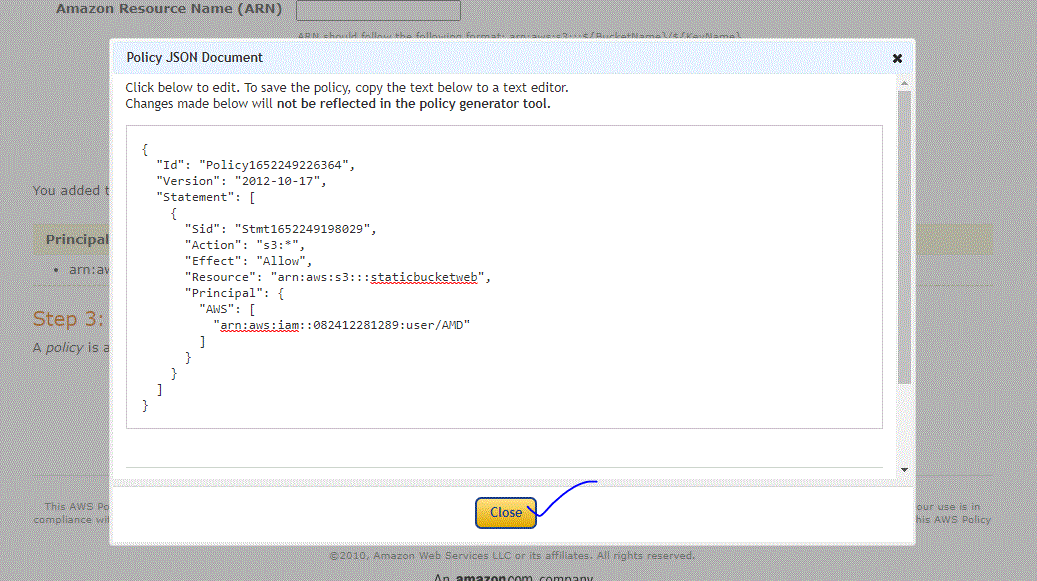


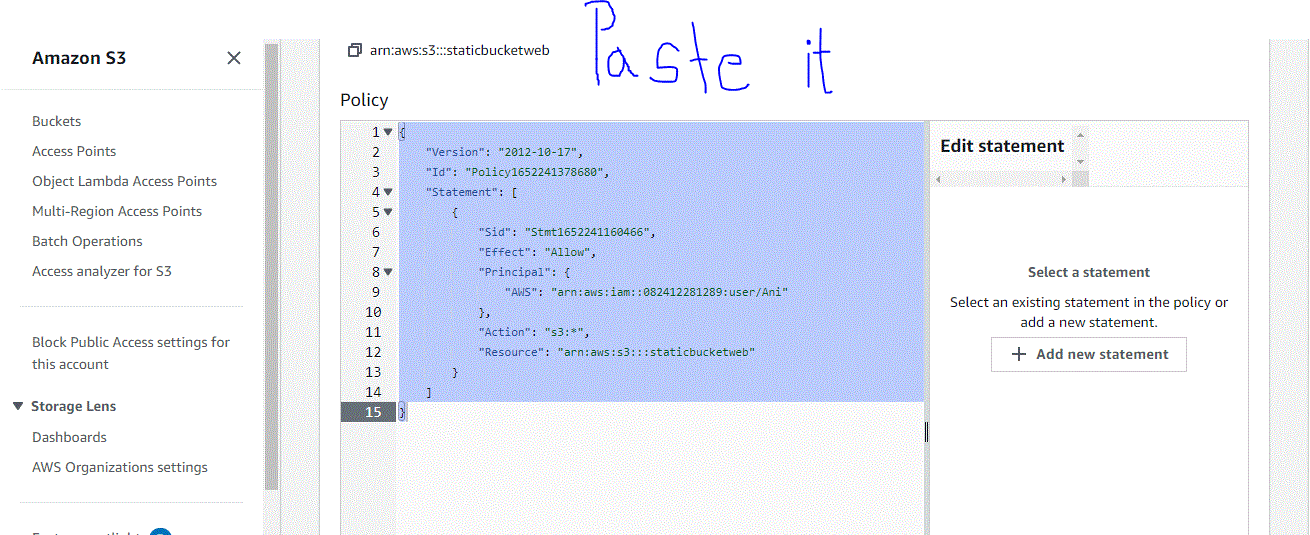












Assignment - 12th May

1) What is lifecycle rule? write the steps with screen shots (any one like transition from one class to other)

: (a) Move current versions of objects b/w storage classes

(b) Move non-current versions of object b/w storage classes

(c) Expire current versions of object

(d) Permanently delete non-current versions of object

(e) Delete expired object delete markets or incomplete multipart uploads

2) What is S3 object lock?

: Amazon S3 Object Lock is an Amazon S3 feature that allows you to store objects using a write once, read many (WORM) model. You can use WORM protection for scenarios where it is imperative that data is not changed or deleted after it has been written

3) What is S3 Glacier vault lock?

: S3 Glacier Vault Lock allows you to easily deploy and enforce compliance controls for individual S3 Glacier vaults with a vault lock policy. You can specify controls such as “write once read many” (WORM) in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.

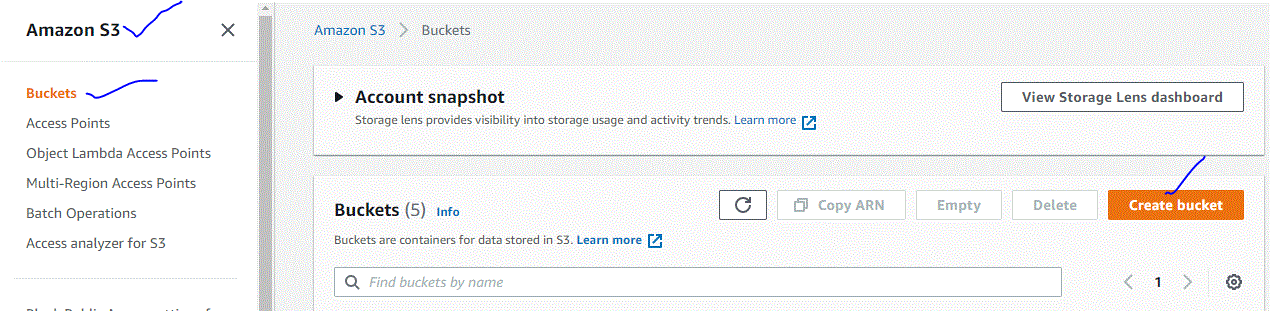
4) What is SRR? explain with hands on steps

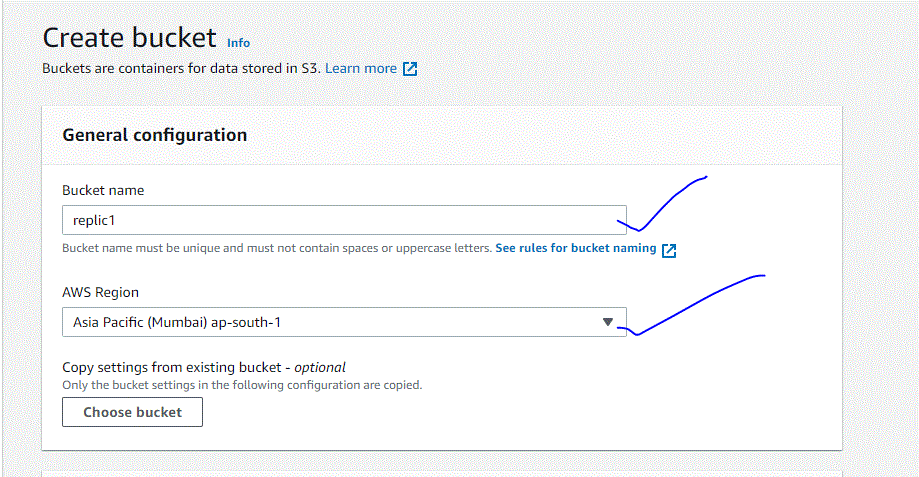
: Same-Region Replication (SRR) is used to copy objects across Amazon S3 buckets in the same AWS Region.

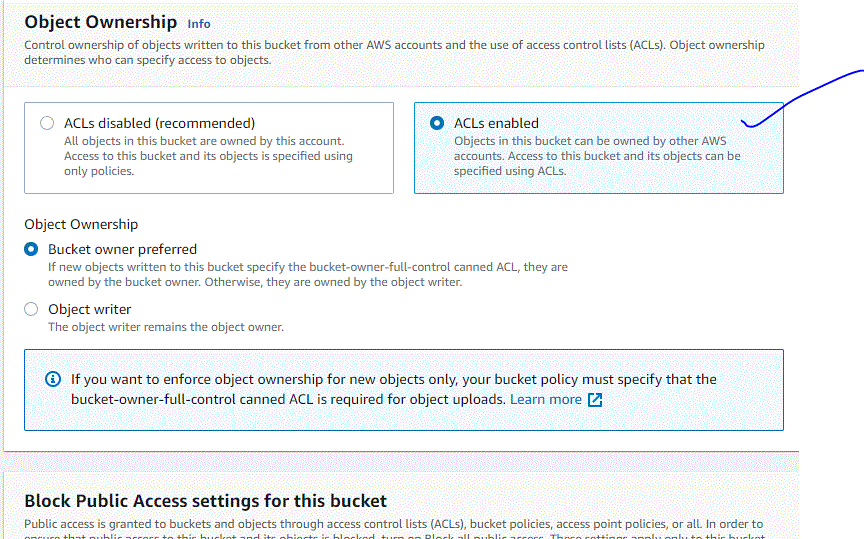
**SRR can help you do the following:** Aggregate logs into a single bucket – If you store logs in multiple buckets or across multiple accounts, you can easily replicate logs into a single, in-Region bucke

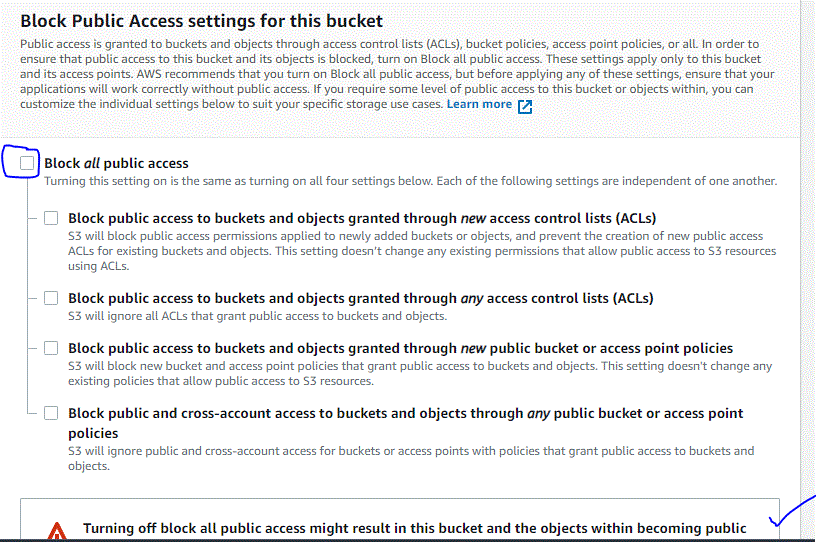
OR

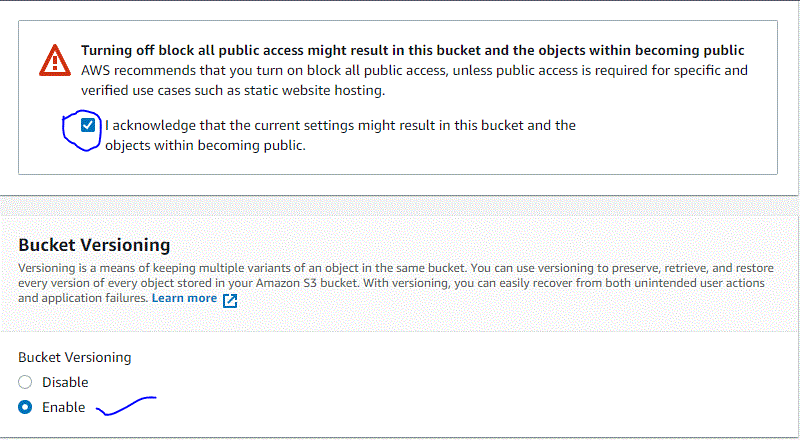
Amazon S3 now supports automatic and asynchronous replication of newly uploaded S3 objects to a destination bucket in the same AWS Region. Amazon S3 Same-Region Replication (SRR) adds a new replication option to Amazon S3, building on S3 Cross-Region Replication (CRR) which replicates data across different AWS Regions.

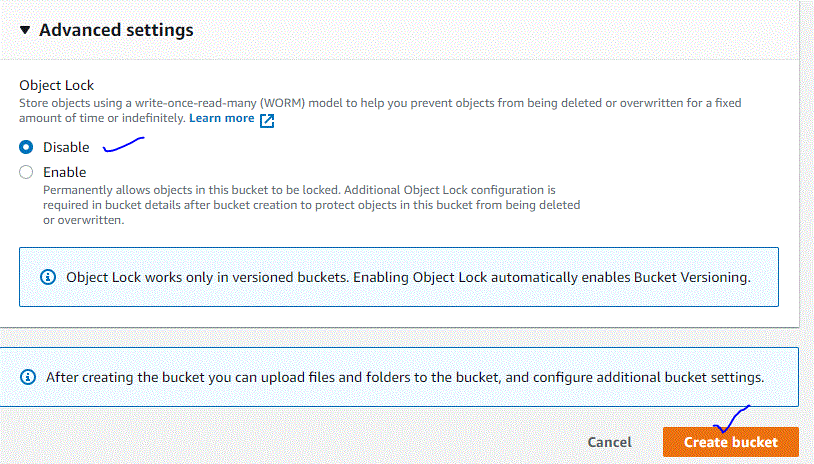


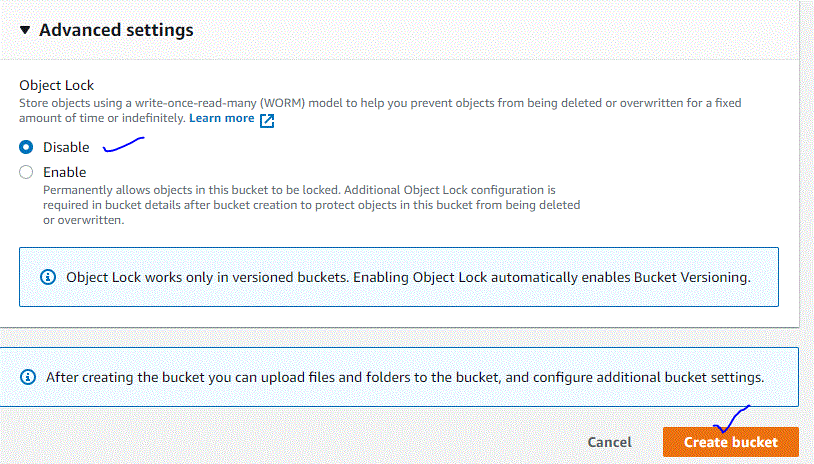


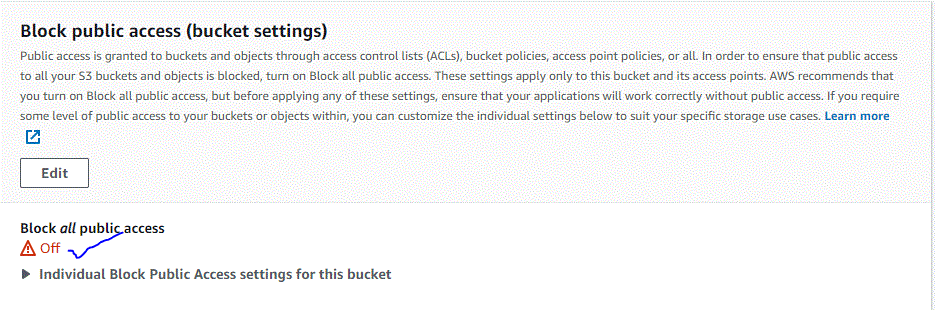


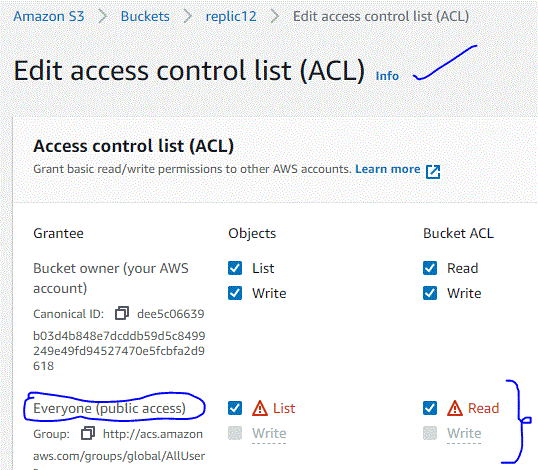


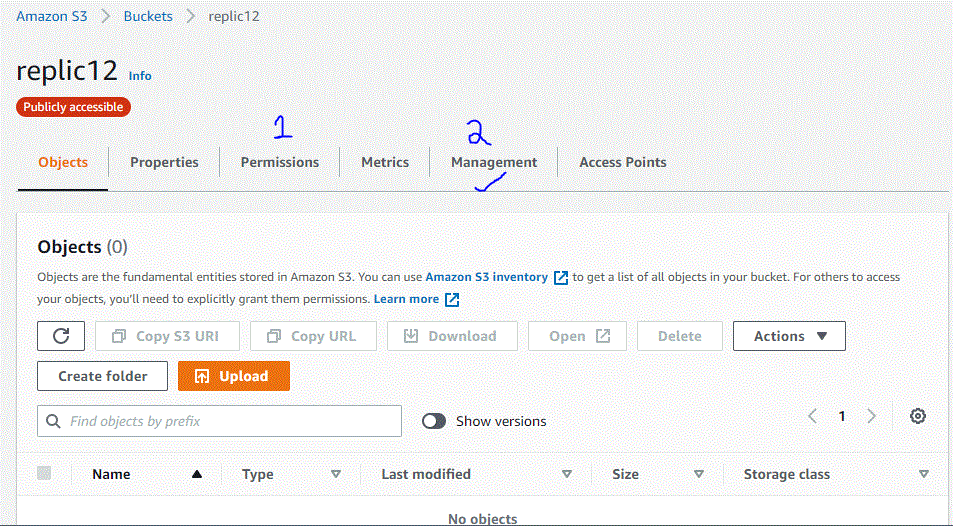


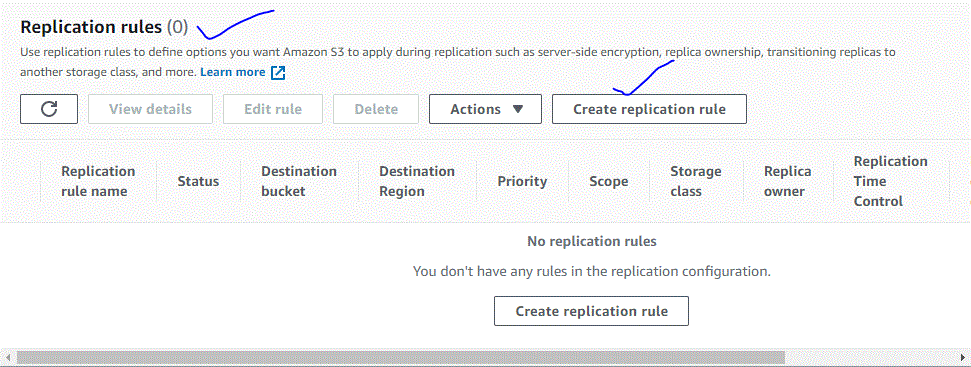


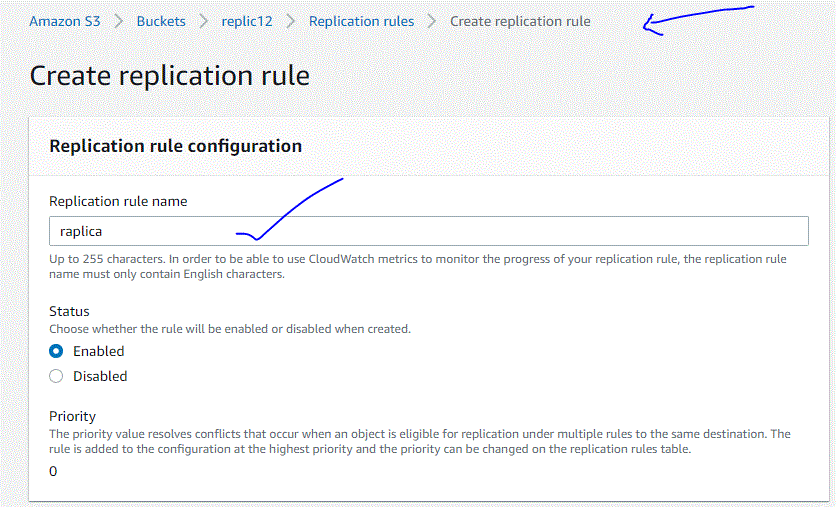


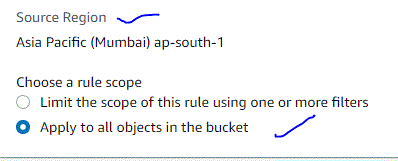


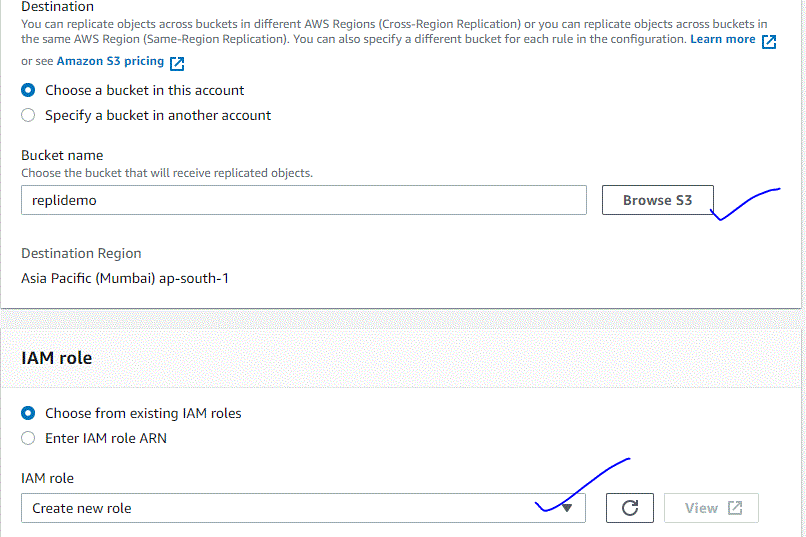


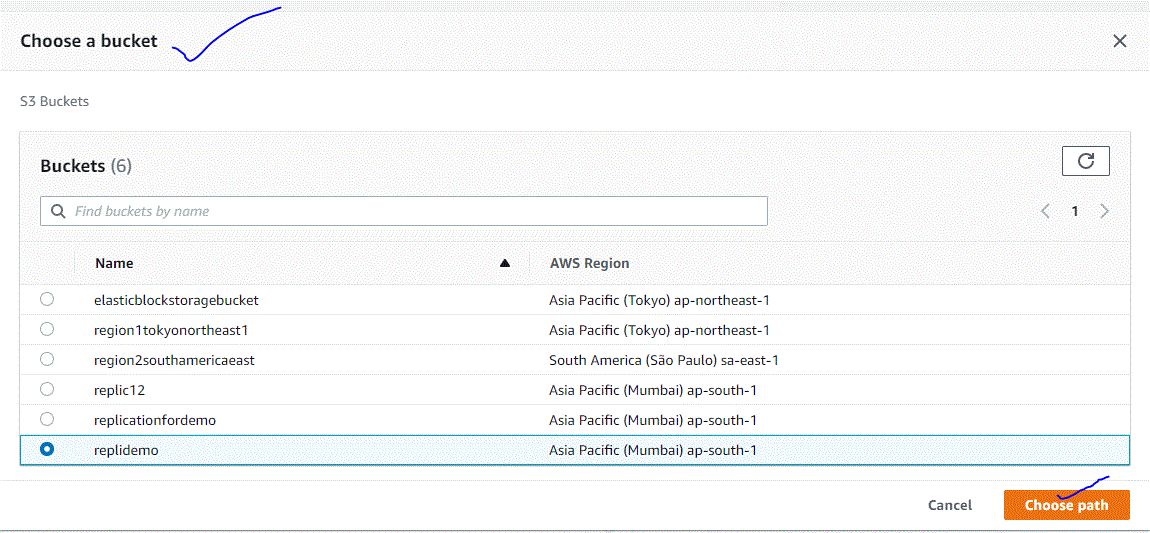


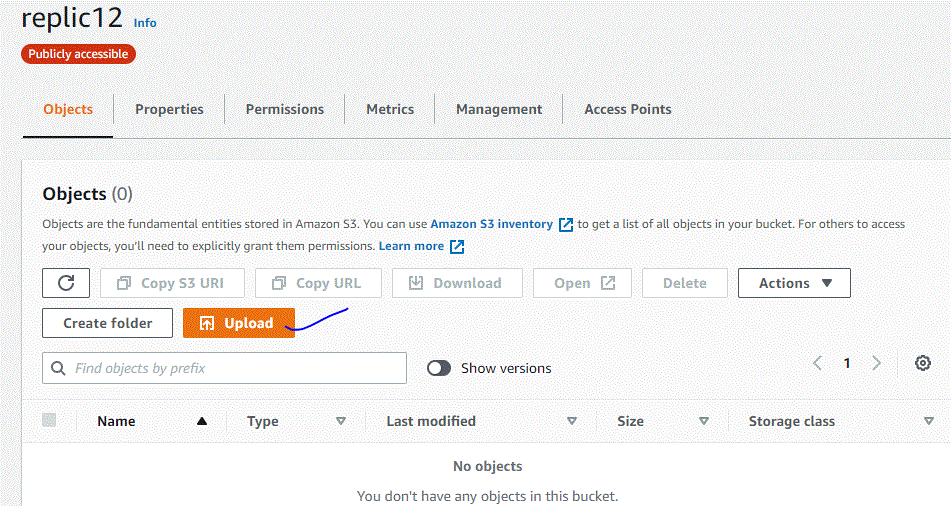


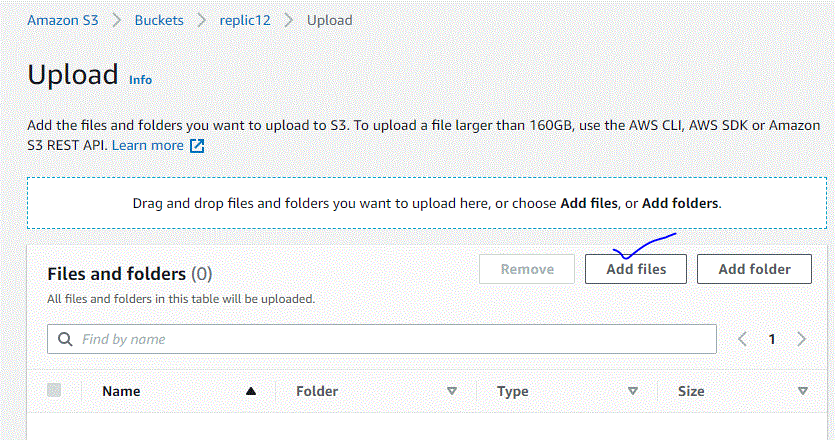




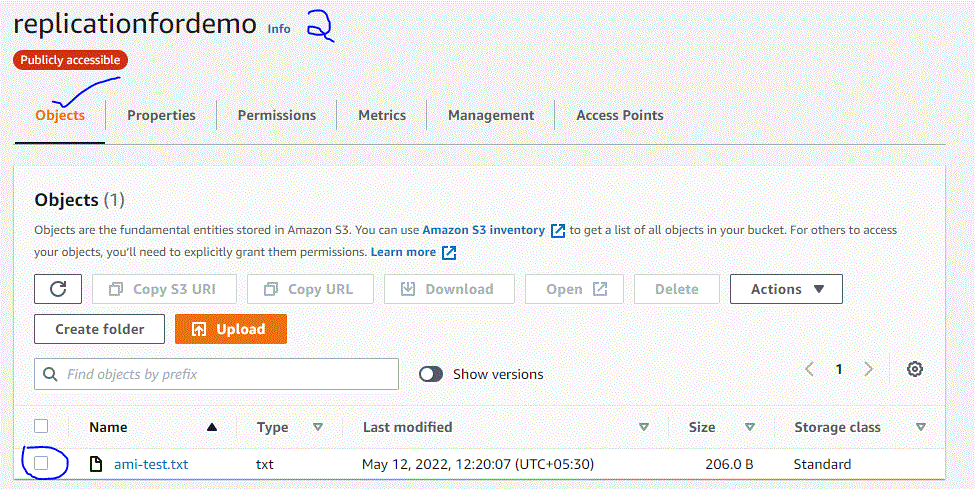












5) What is CRR?

: CRR is an Amazon S3 feature that automatically replicates every object uploaded to an S3 bucket to a destination bucket in a different AWS Region that you choose. S3 object tags are key-value pairs applied to S3 objects that allow you to better organize, secure, and manage your data stored in S3.

OR

Amazon S3 CRR automatically replicates data between buckets across different AWS Regions. With CRR, you can set up replication at a bucket level, a shared prefix level, or an object level using S3 object tags.

6) How to enable versioning? hands on steps

: Versioning is a means of keeping multiple versions of same file in the same S3 bucket. Ideally it lets you store, retrieve and restore each version of every object you store in S3 bucket.

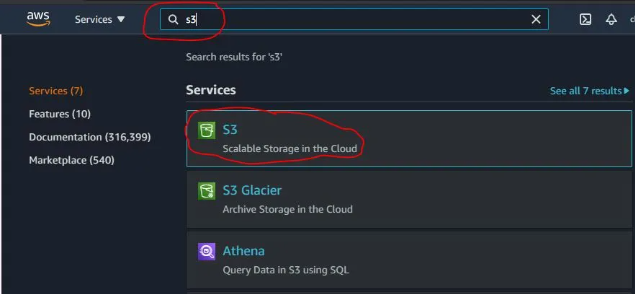
### Step 1: Login to AWS Management Console and open S3

Login to [**AWS Management Console**](https://aws.amazon.com/console/) and open S3 service.

You can either go to **Services -> Storage -> S3**

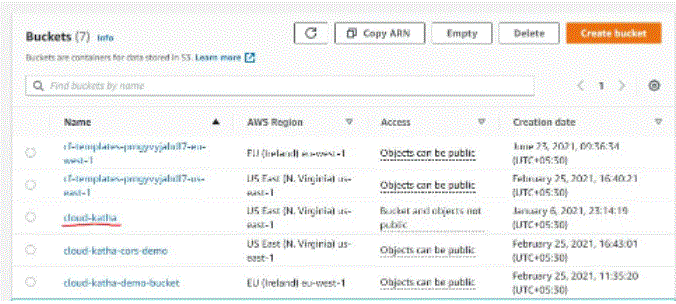
or

Type **s3** in the search bar and hit enter. Once you see S3 option click on that.



### Step 2: Choose the bucket on which you want to enable versioning

Once you click on S3, you will see the list of your buckets as you can see below.

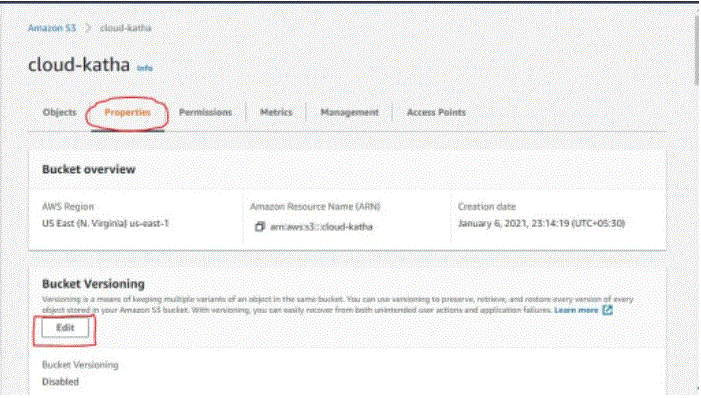


Click on your **bucket name** on which you want to enable versioning.

For this tutorial I will be enabling versioning on **cloud-katha** bucket using console. For CLI demo, I will create a separate bucket.

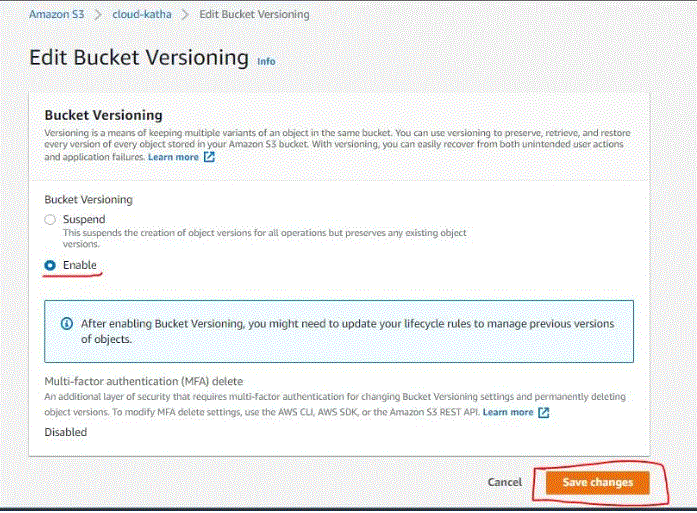
### Step 3: Go to Properties tab and Edit Versioning Status of Bucket

After you click on your bucket-name, you will see details like below. Click on **Properties**tab. Once you are there, you will see **Bucket Versioning**and an Edit button to edit the versioning status of the bucket.



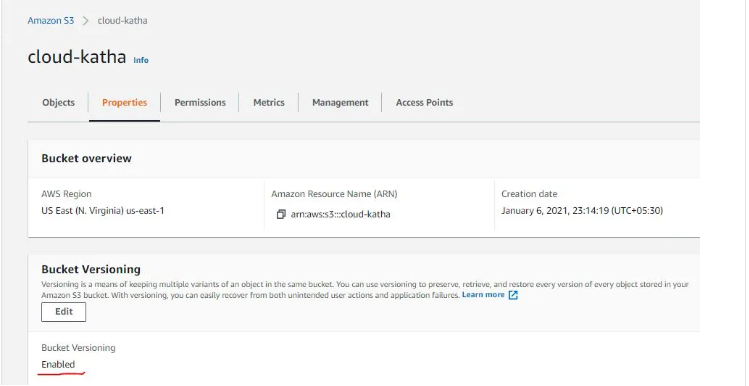
As you can see current status is **Disabled**

Let’s edit it to enable versioning on this bucket



Select **Enable** in bucket versioning and click on **Save changes**

One you click Save changes, versioning is enabled as you can see in below screenshot and is it effect.



### Step 4: Verify If Versioning is working fine

You can verify versioning by uploading two objects with same name/key. With versioning enabled, S3 will not override the object rather it will create a new version each time.

Create a text file with name versioning-demo.txt with below content

This is version 1 file

Save the file and upload into **cloud-katha** bucket

Now update the same text file with below content and upload it again to the same bucket.

This is version 2 file

Let’s observe what happened in console.

If you toggle **Show versions** button as highlighted below, you can see both the versions of file present in s3. So, object was not overwritten instead both the versions were saved, which verifies that versioning is working fine

