

Computer Engineering Department
Program: Sem VI
Course: Cloud Computing Lab(CSL605)
PART A

(PART A: TO BE COMPLETED BY STUDENTS)

Experiment No.5

A.1 Aim:

To demonstrate and implement Storage as a service using AWS S3 Service

A.2 Prerequisite:

Knowledge of Networking, Distributed Computing and knowledge of Software architectures.

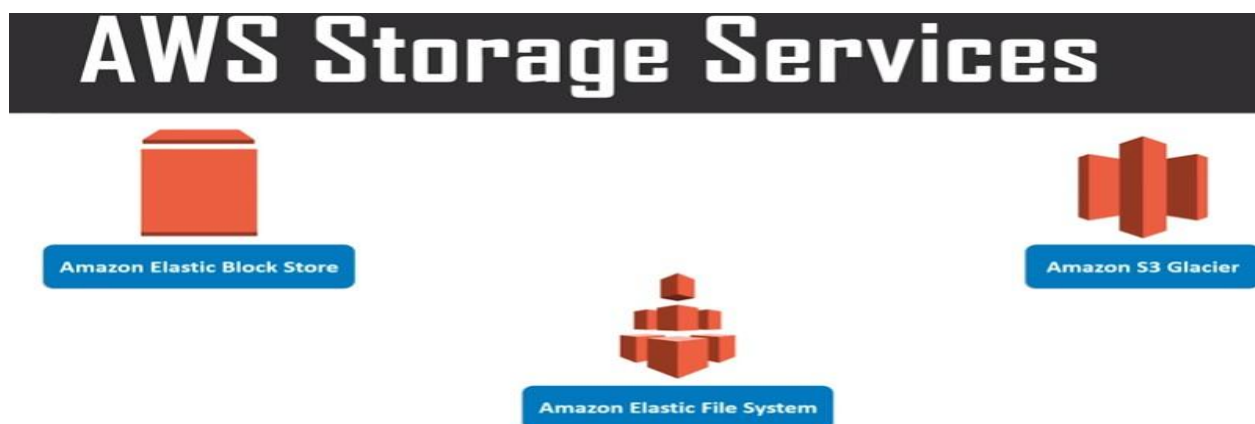
A.3 Objective:

Objectives this experiment is to provide students overview of AWS storages, its Features and Services.

A.4 Outcome: (L03)

After successful completion of this experiment student will be able to; Implement IAAS deployment model of cloud.

A.5 Theory:



Amazon Elastic Block storage (EBS): Storage in the form of blocks, each of these blocks associated with one particular instance, so when to access this block make sure that you have an instant connected to it and storage is accessed through that instant only.

Amazon Elastic File System (EFS): It is shared file system; hence it is not attached to a particular system or operating system.

Amazon S3 Glacier: Need to store archive data, certain data that we would not want to retrieve or access on daily basis or frequently, such data you can put on **S3 Glacier**. That is it locks data for certain time during which you cannot access it, once you clear that duration you are free to access that data.

This storage is very cheap as compared to other data storage.

Example: Hospital System (birth certificate data): Birth certificate given by hospital once

baby takes birth, so once you get it you are not requesting it again and again and hospital need to maintain it. That is data which is important but not needed on daily basis.

AWS Storage Gateway: Act as middle ware to move data from one system to another system.

+ What is Amazon S3 (Simple Storage Service)

- 1) Amazon s3 has a simple web service interface that you can use to store and retrieve any amount of data, at anytime from anywhere on the web.
- 2) **S3 works on objects and buckets** (Bucket is an container and an object (doc, image, file etc) is an file, which you can stored in container)
- 3) Online cloud storage

Pre-S3

Online Cloud Storage

- Online Storage – Upload files, folders, images, songs, videos from a machine and access it from anywhere in the world.



Dropbox



iCloud



OneDrive

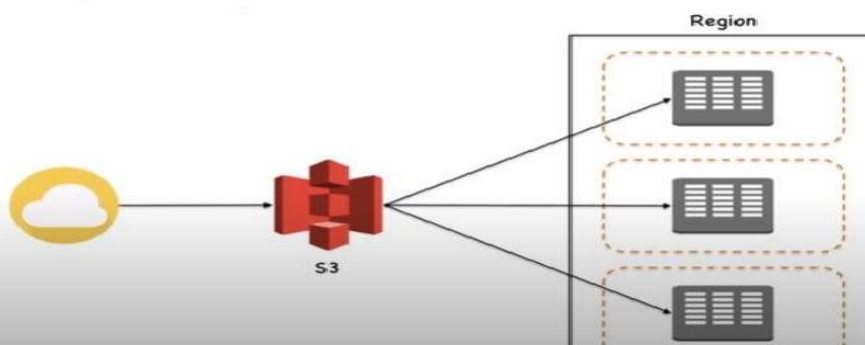


Google Drive

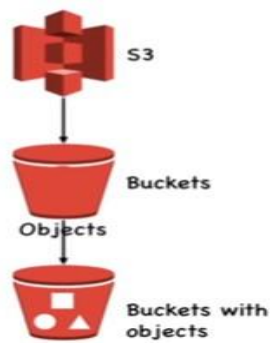
S3 Concepts

Simple Storage Service

- Simple Storage Service is storage for internet.
- S3 provides web service which can be used to store and retrieve unlimited amount of data. Same can be done programmatically using Amazon provided APIs.
- **S3 Data Consistency Model**
 - S3 provides highly durable and available solution by replicating all data in multiple data centers in a region.
 - Data uploaded in a particular region never leaves it.
 - Read-after-write consistency.
 - Eventual consistency.



- S3 follows a storage hierarchy in keeping data (documents, images, videos, files etc.).



- Management console or S3 APIs can be used to manage buckets and objects.
- Bucket names have to be Globally unique irrespective of which region they are created in.
- Max 100 buckets can be created per account.

Amazon S3 (Simple Storage Service) provides object storage which is built for storing and recovering any amount of information or data from anywhere over the internet



4)

- ✓ Amazon S3 provides storage through web services interface
- ✓ It is designed for developers where web-scale computing can be easier for them
- ✓ It provides 99.999999999% durability and 99.99% availability of objects
- ✓ It can store computer files up to 5 terabytes in size

5)

Object & Bucket in Amazon S3

An object consists of data, key(assigned name) and metadata

A bucket stores objects

When data is added to the bucket, Amazon S3 creates a unique version ID and allocates it to the object

For Example:



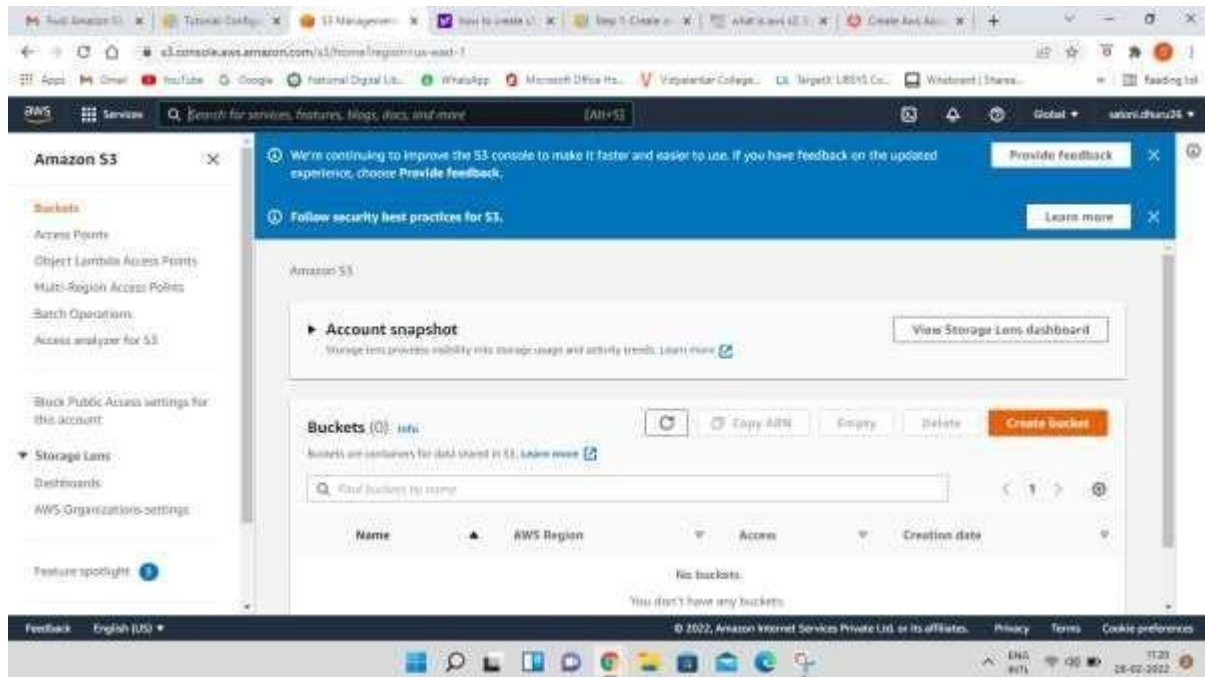
Object: folder/Penguins.jpg
 Bucket: simplilearn
 Link Address: <https://s3.amazonaws.com/simplilearn/folder/Penguins.jpg>

Key(name)

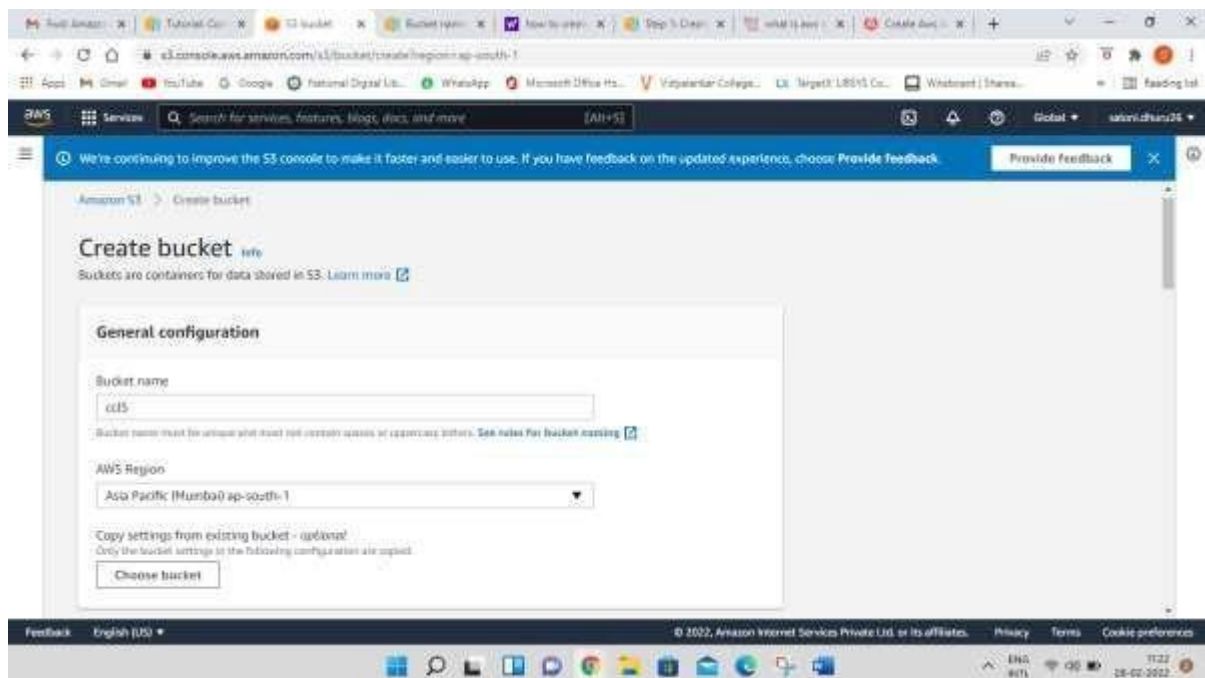
Version ID

Steps to Implement Storage as a Service using Own Cloud/ AWS, Glaciers

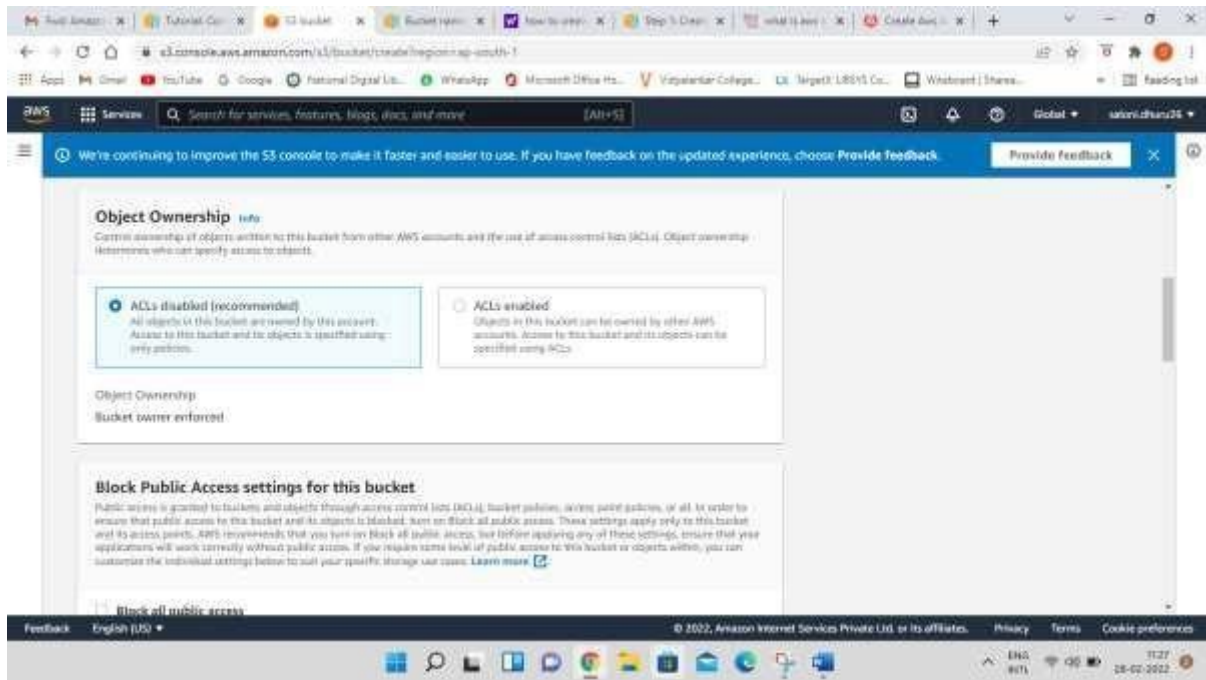
Step-1: click on create bucket



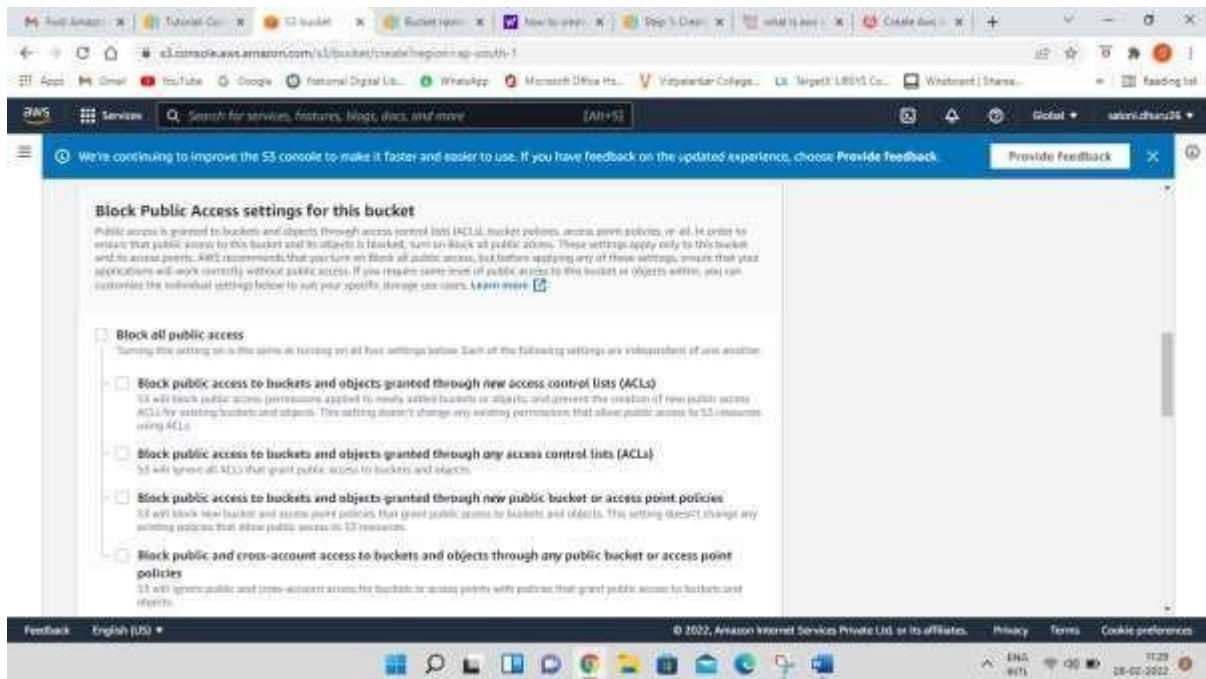
Step-2: Give Bucket name & select region for storage



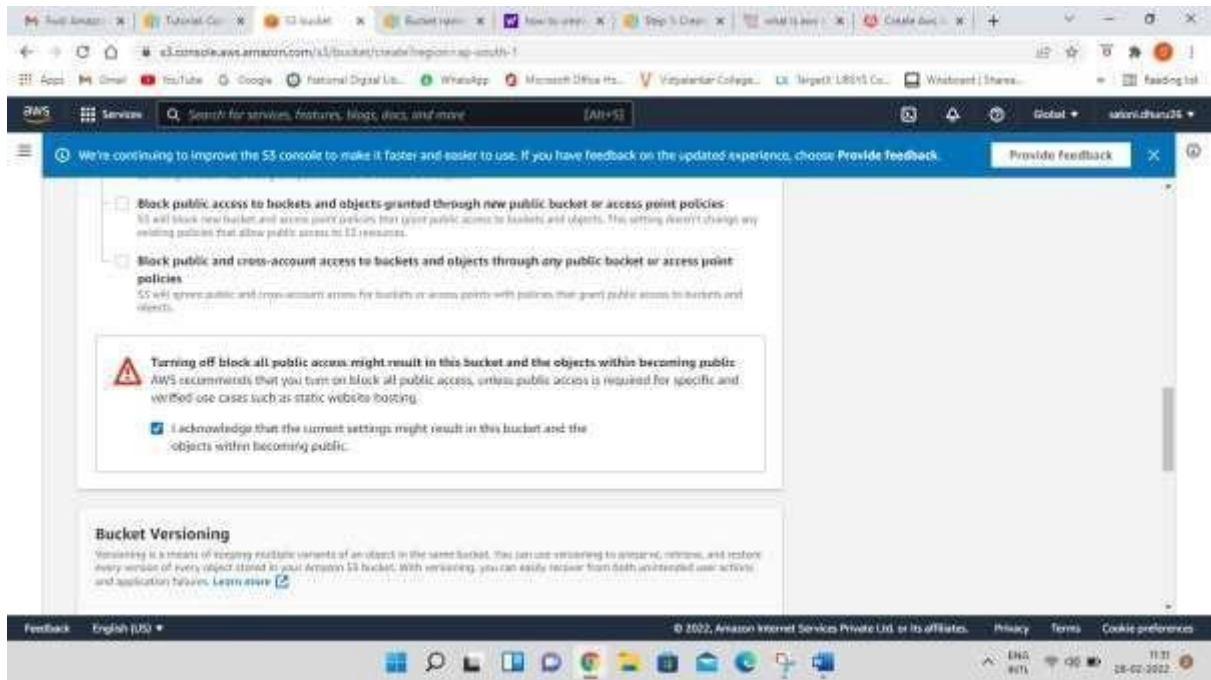
Step-3: Keep object ownership setting as ACLs Disabled as by-default



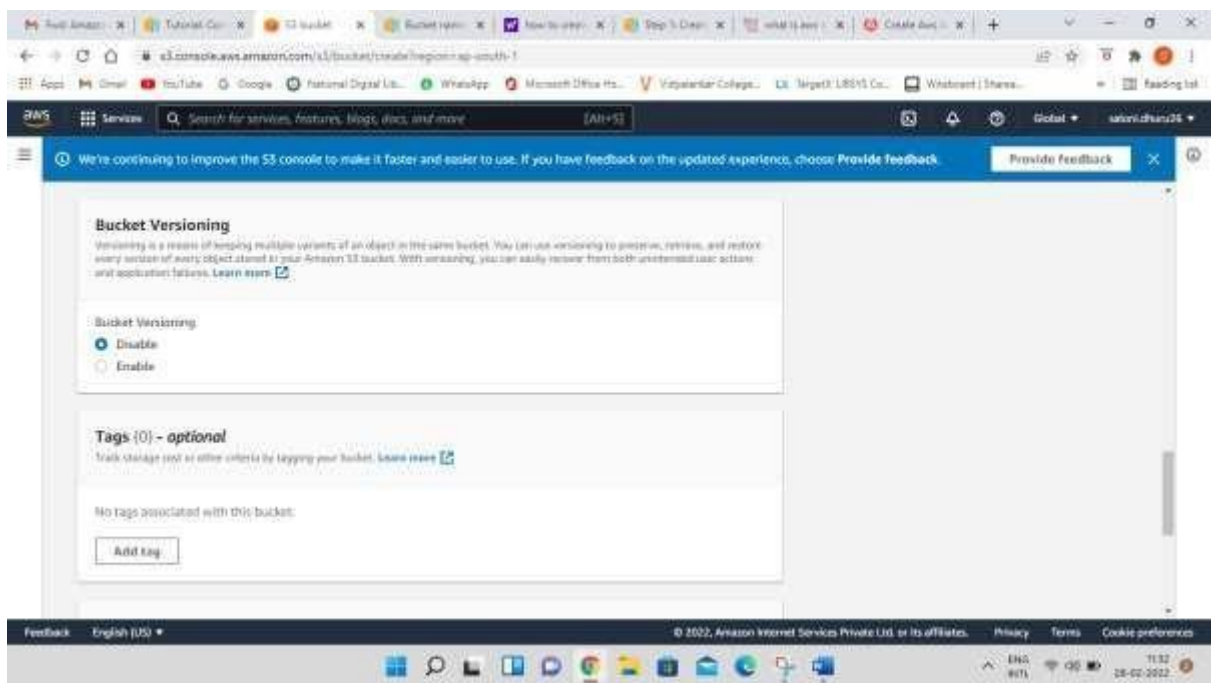
Step-4: Disable block all public access checkbox



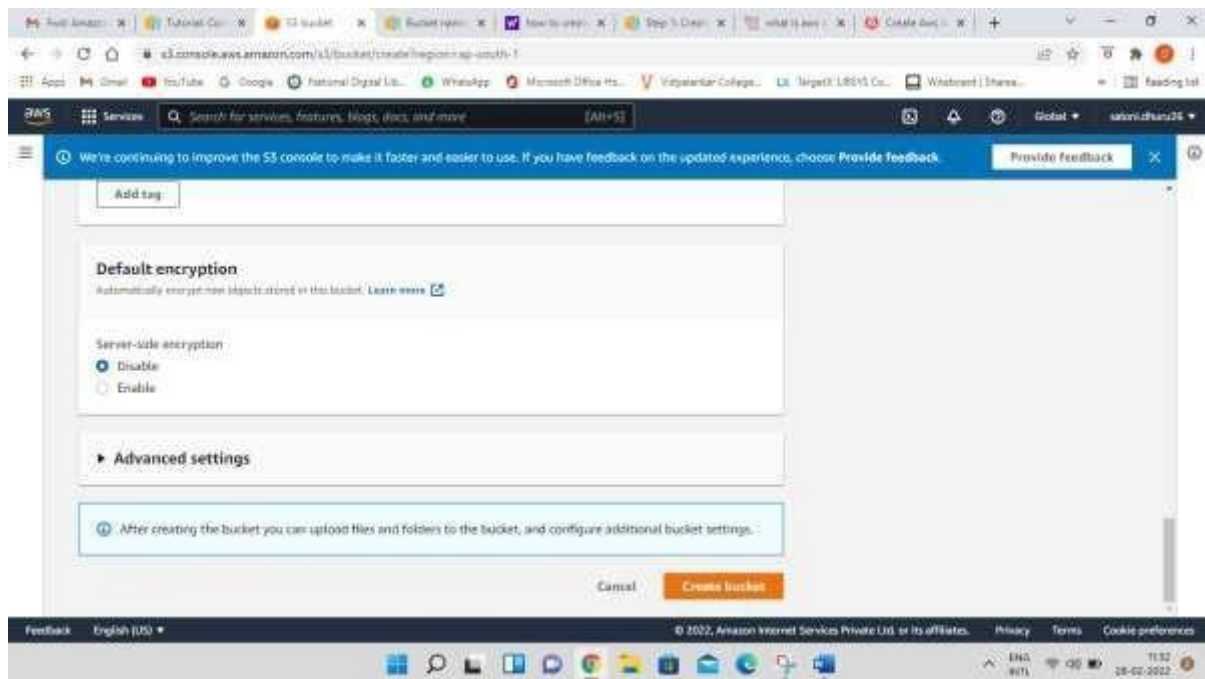
Step-5: Select the checkbox for Turning off block all public access might result in this bucket and the objects within becoming public



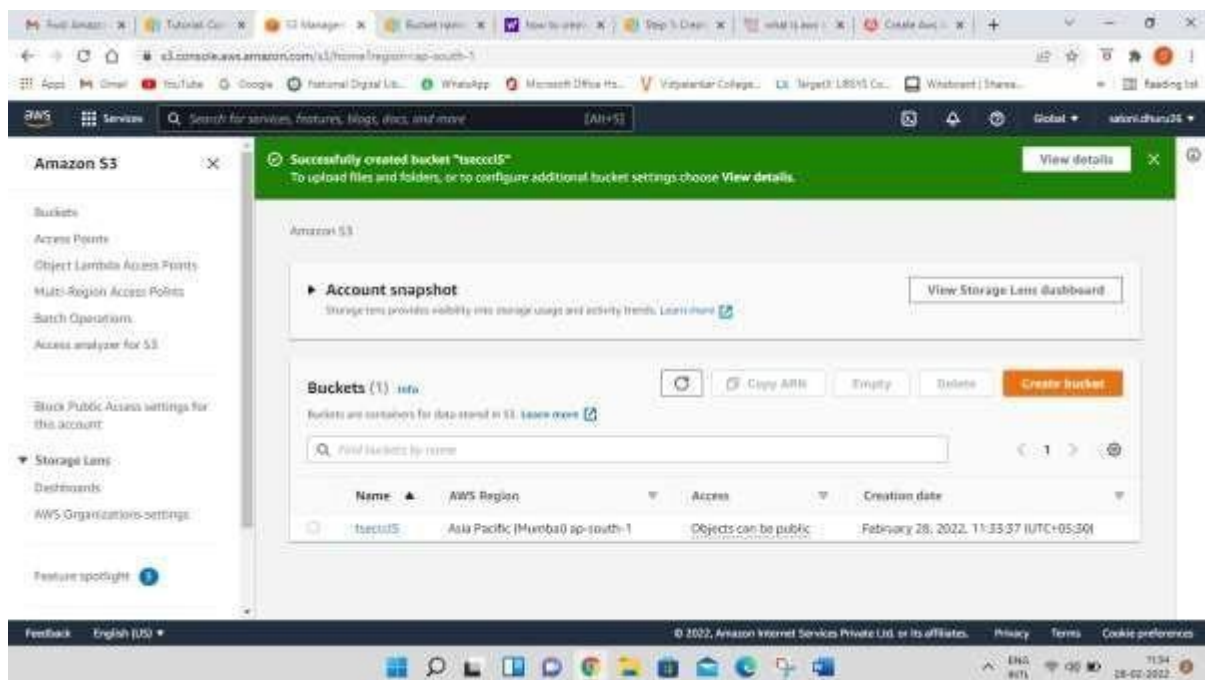
Step-6: Keep bucket versioning as disabled and add tags if required.



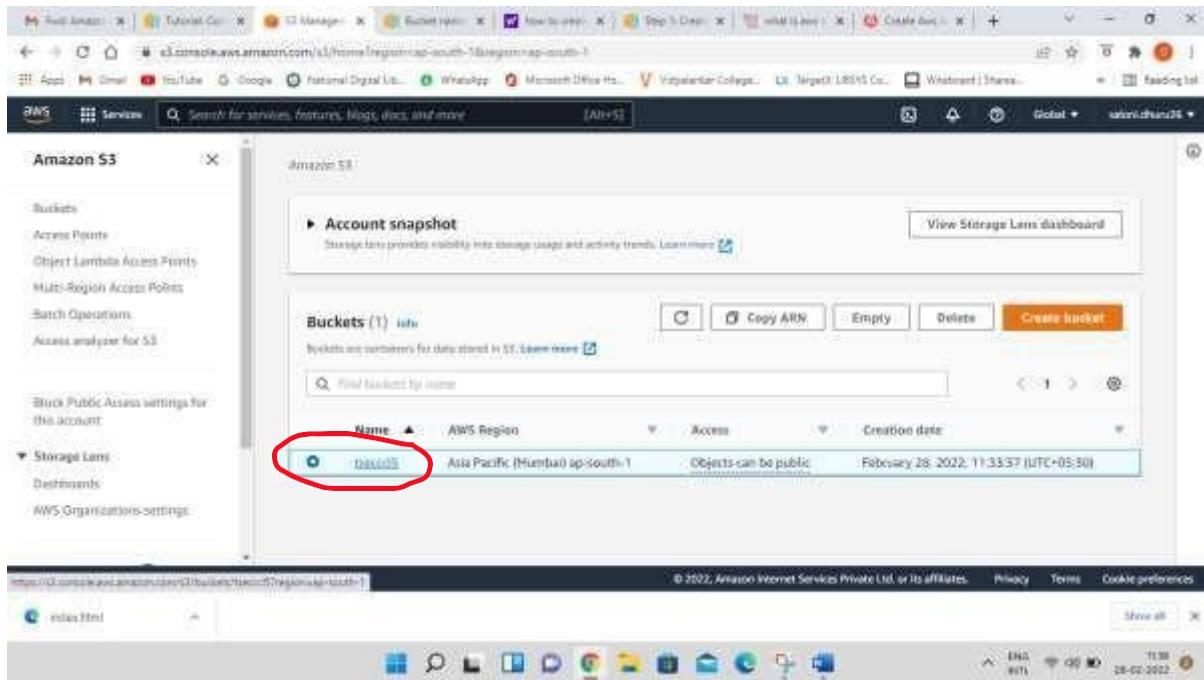
Step-7: Keep default encryption disabled and click on create bucket button



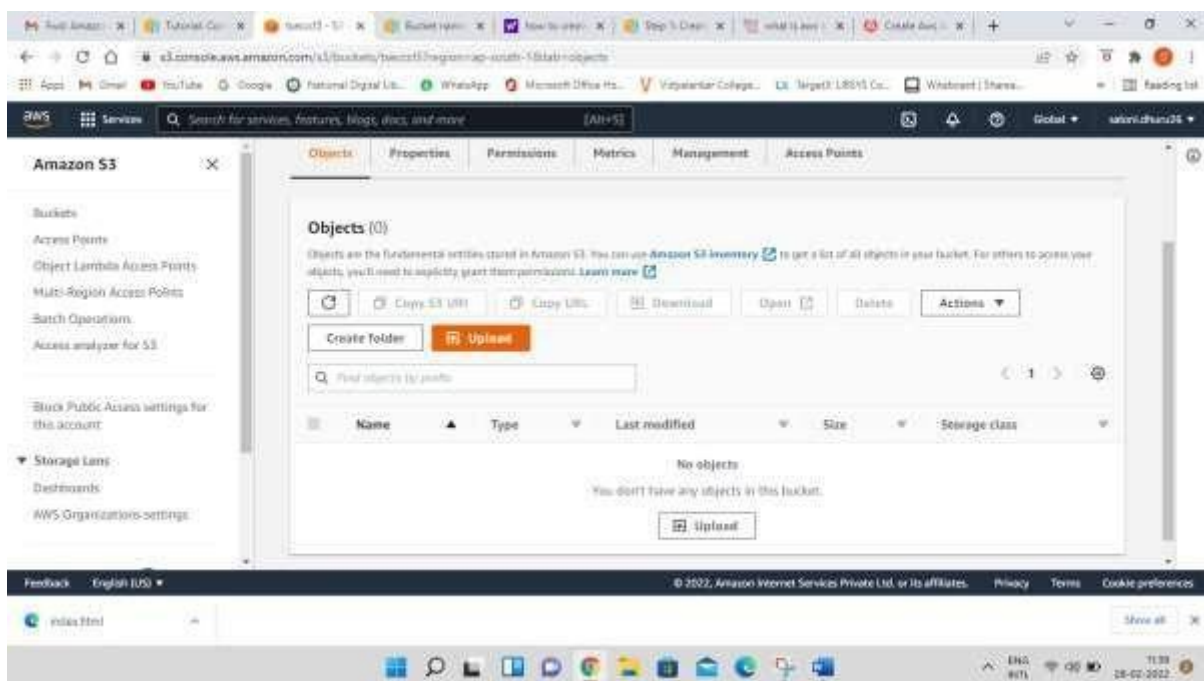
You can now see the successful creation of your bucket



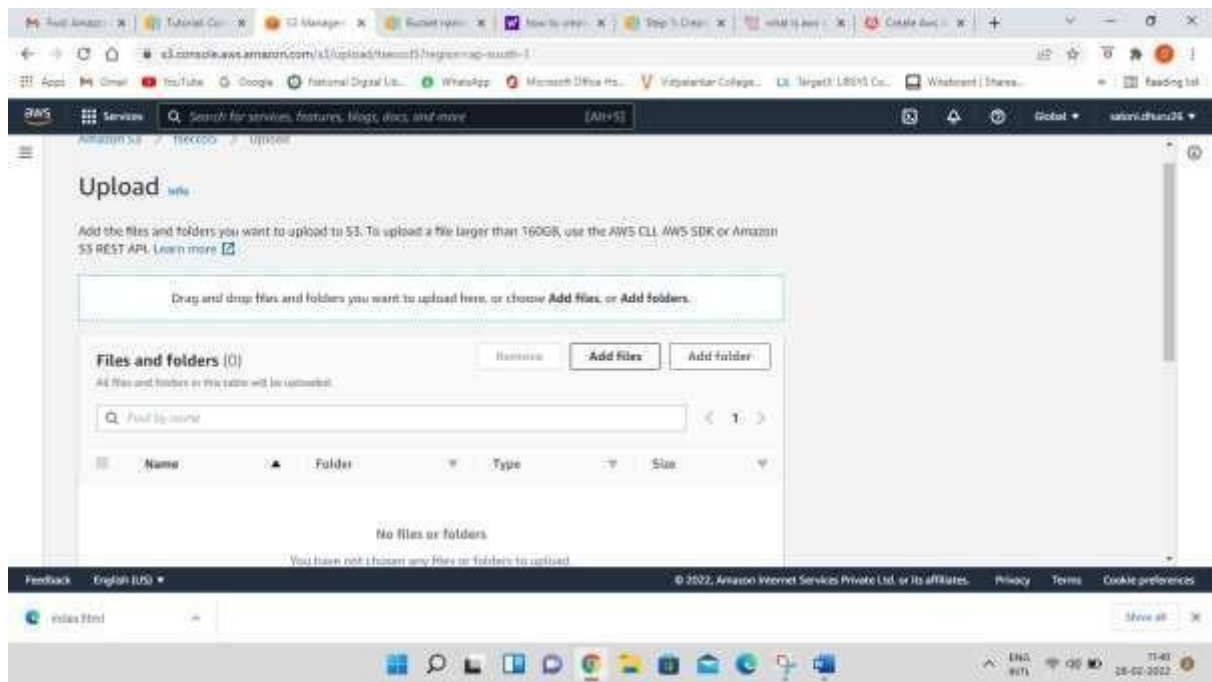
Step-8: now click on the bucket that you have created



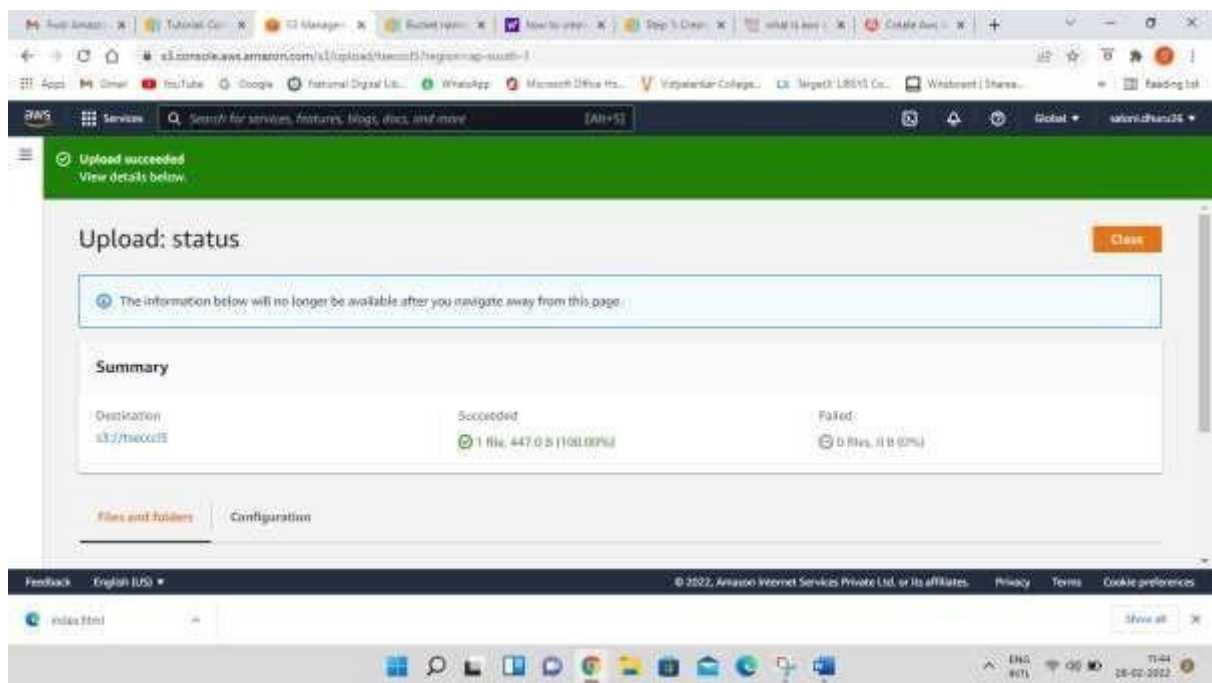
Step-9: You can either create a folder here or upload an existing file in the bucket



Step-10: now click on upload button and click on add files button browse your local machine and select which file you need to upload on S3 next click on upload button at bottom right end

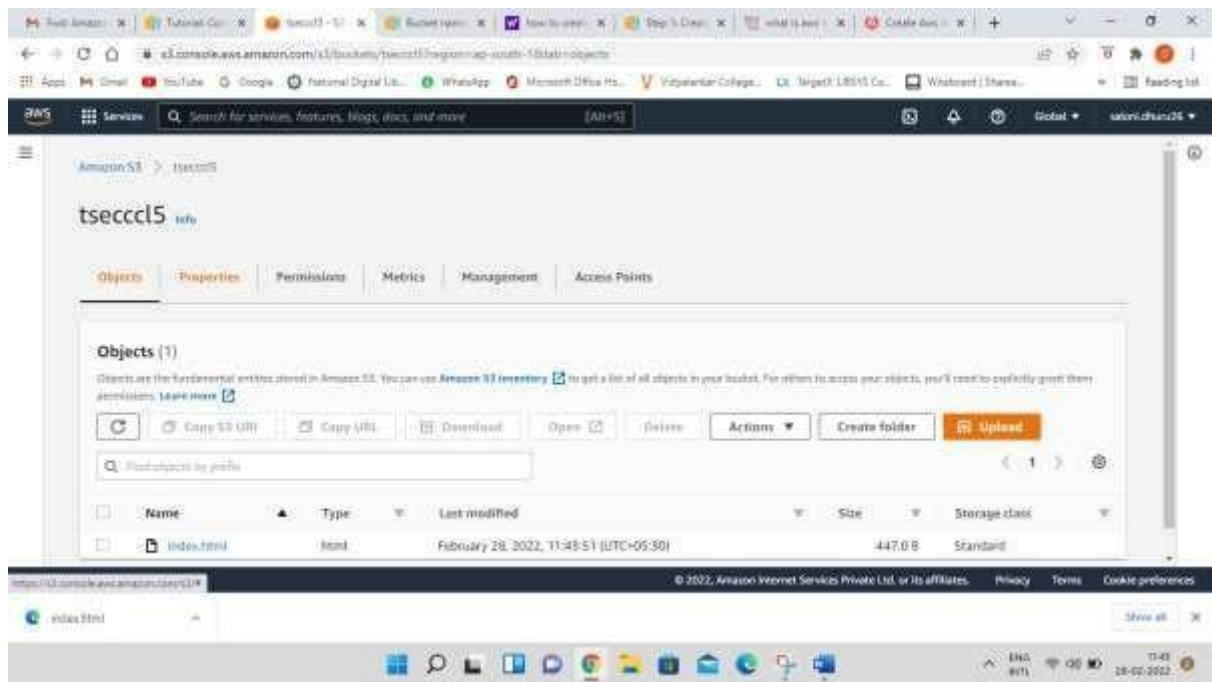


Now you can check the upload status screen

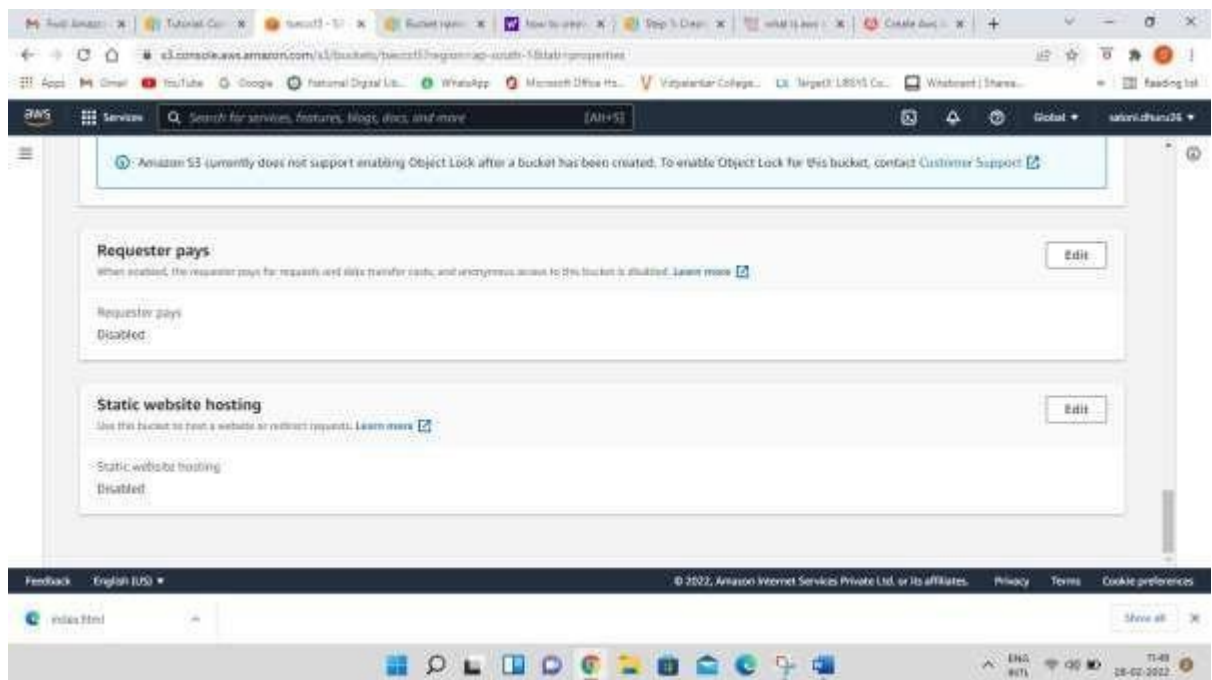


Now click on close button

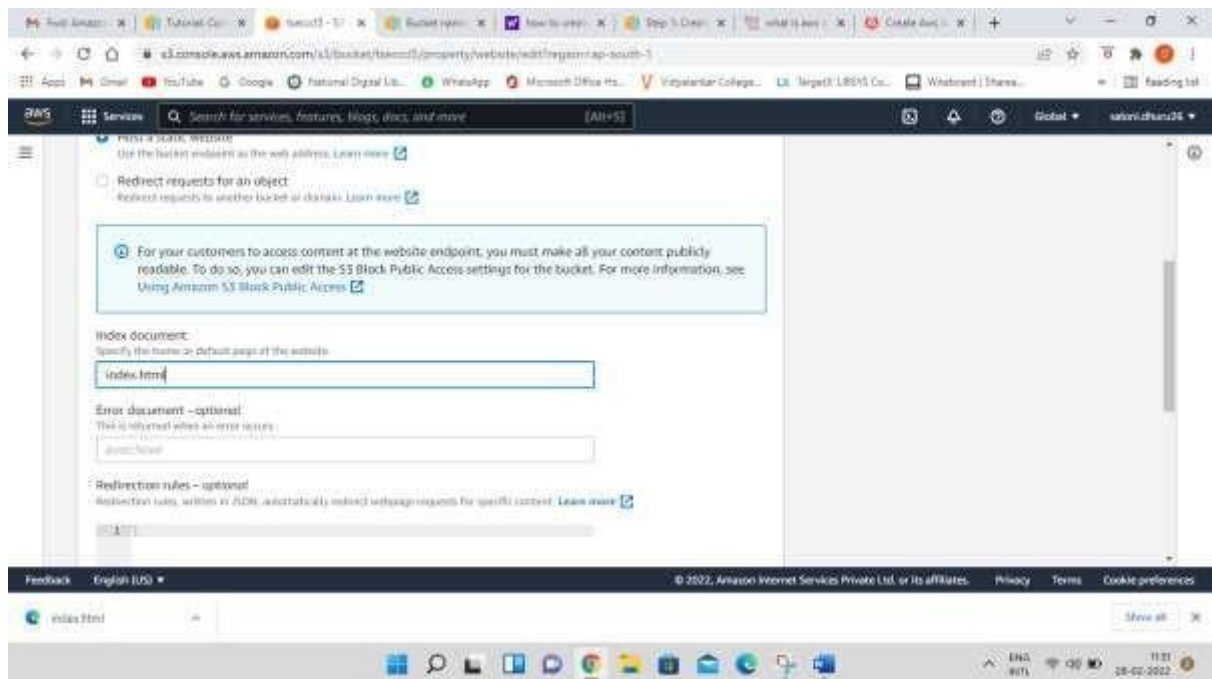
The screen will appear as below



Step-11: Select properties and scroll down to **Static website hosting** option which is disabled now click on Edit option on right side



Step-12: Enable the radio button and specify the file name in **Index document** which you have added in S3



Edit static website hosting [Info](#)

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

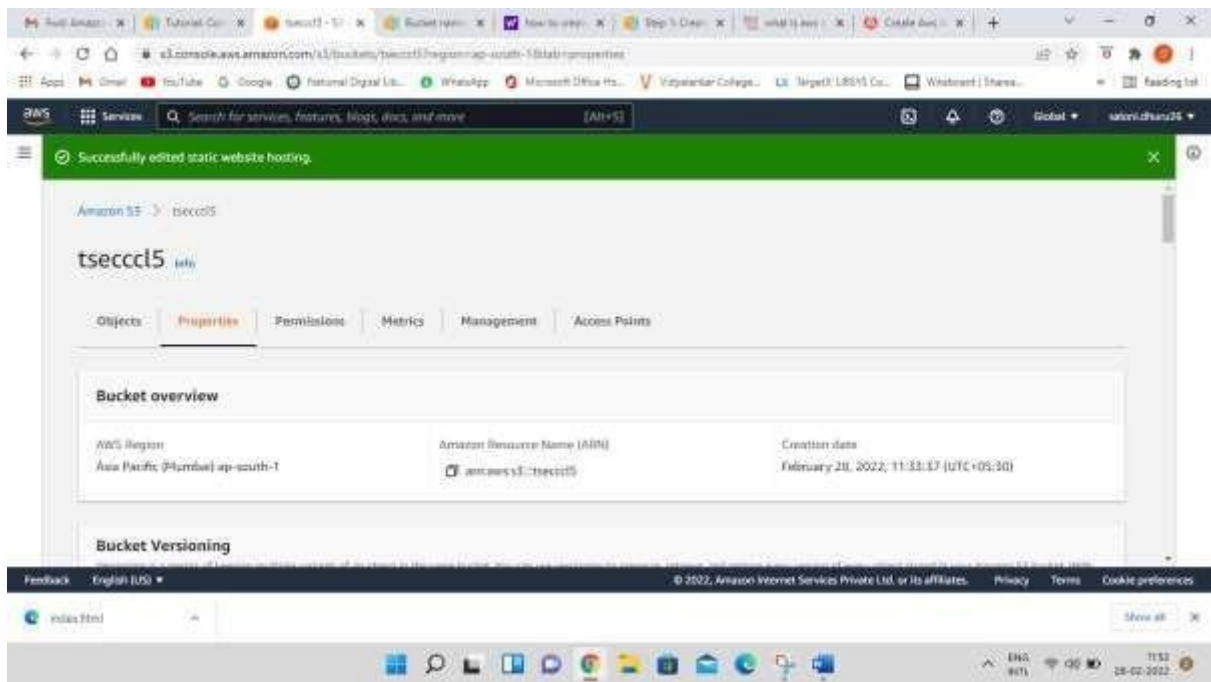
- ☐ Disable
- ☒ Enable

Hosting type

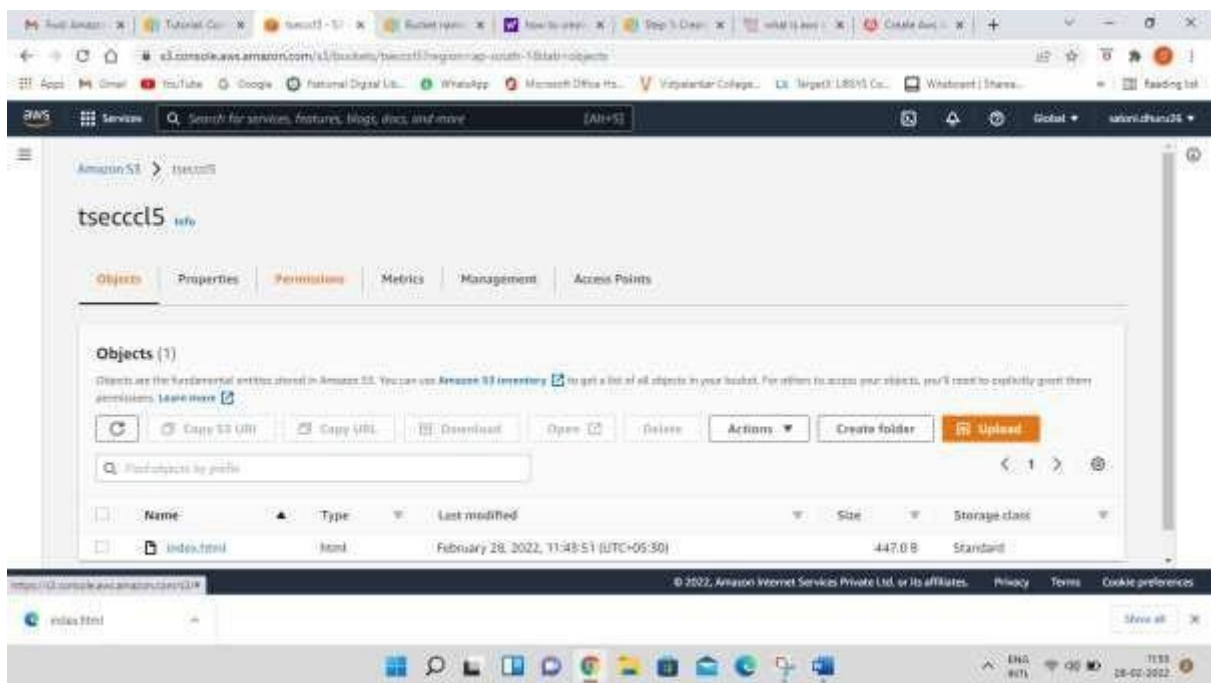
- ☒ Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
- ☐ Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Scroll down and save the changes at bottom right

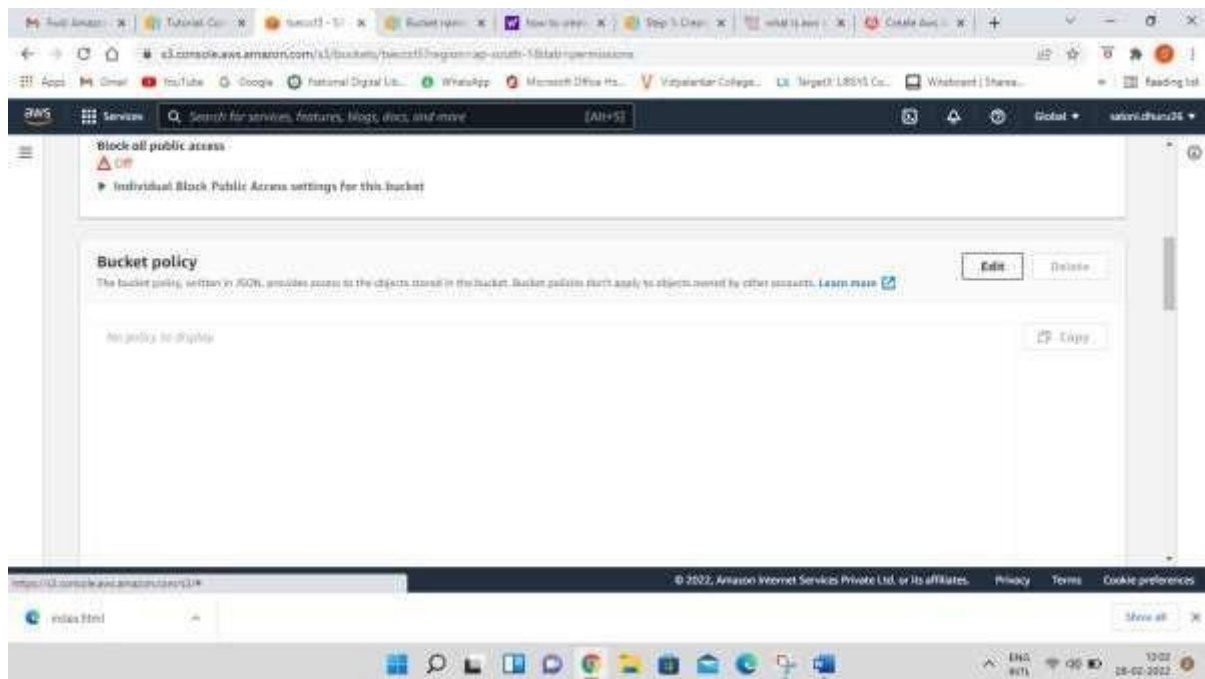
Following screen will appear



Step-13: Click on Permissions Tab



Step-14: In **bucket policy** click on Edit option

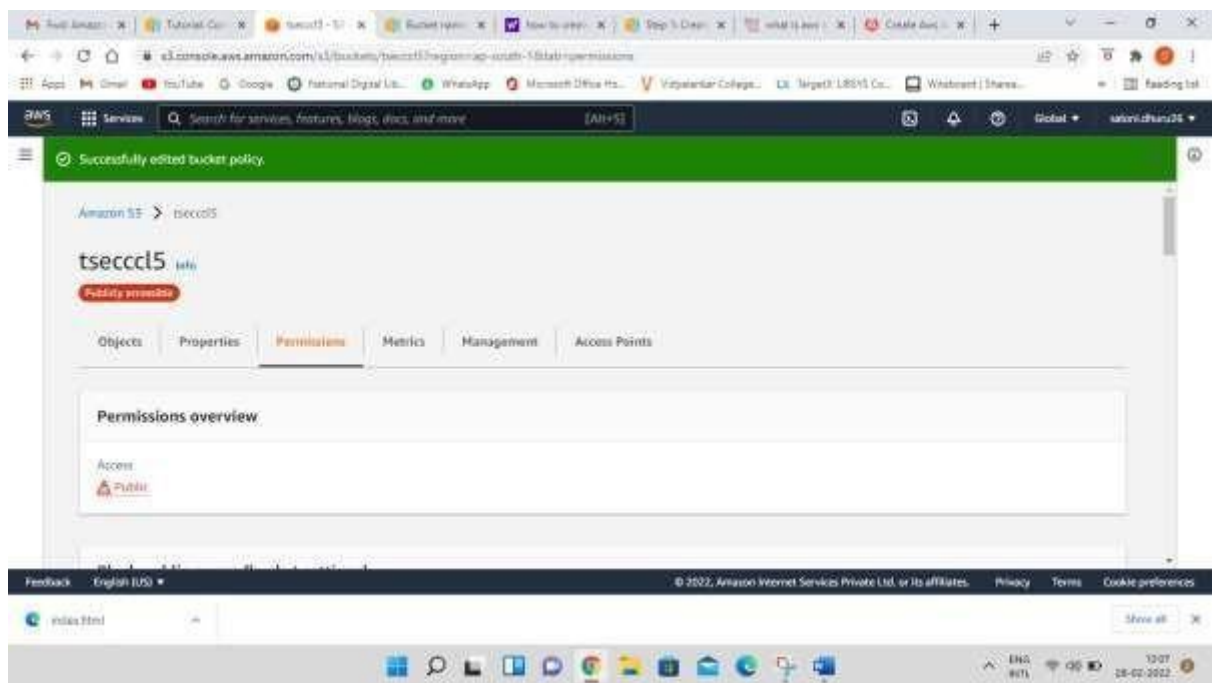
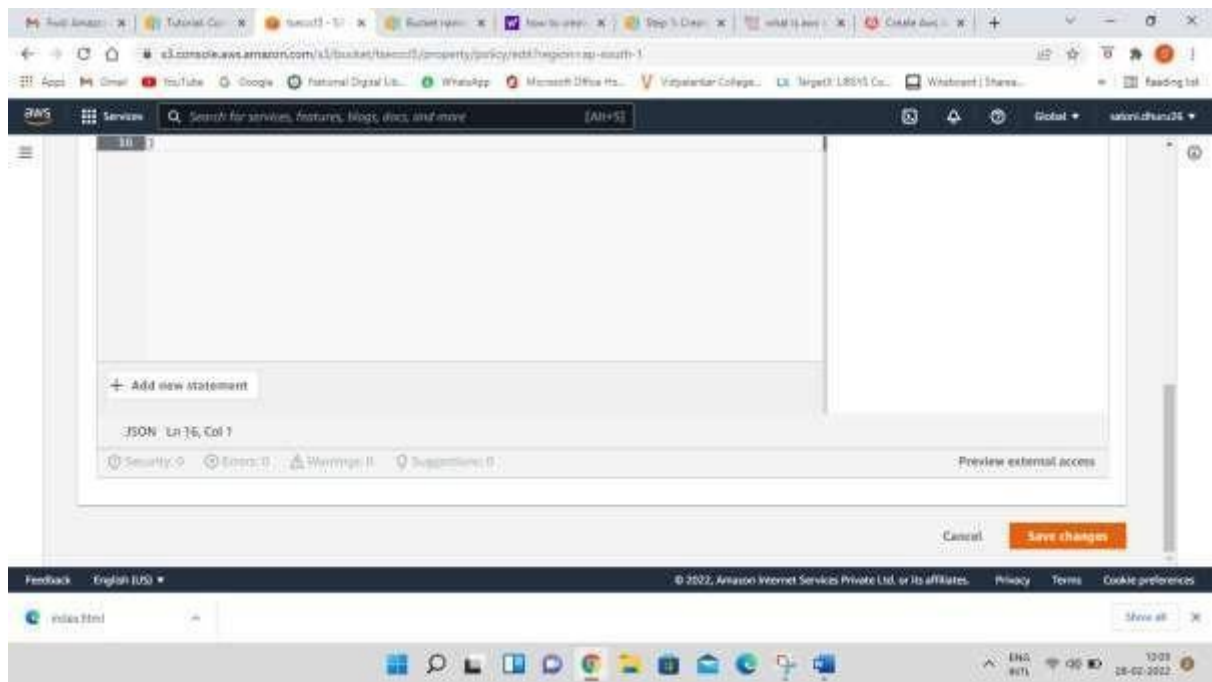


Step 15- after clicking on edit button paste the following code in bucket policy

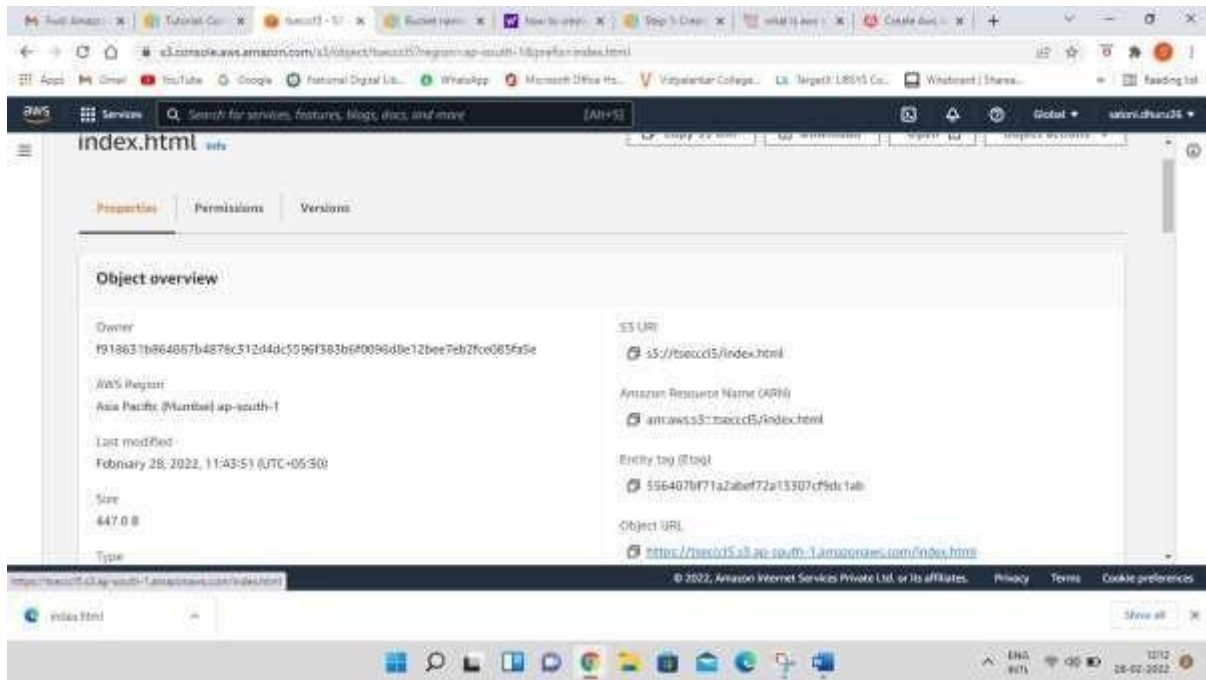
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::Bucket-Name/*"
      ]
    }
  ]
}
```

Note-Make sure that you add your bucket name in the code above

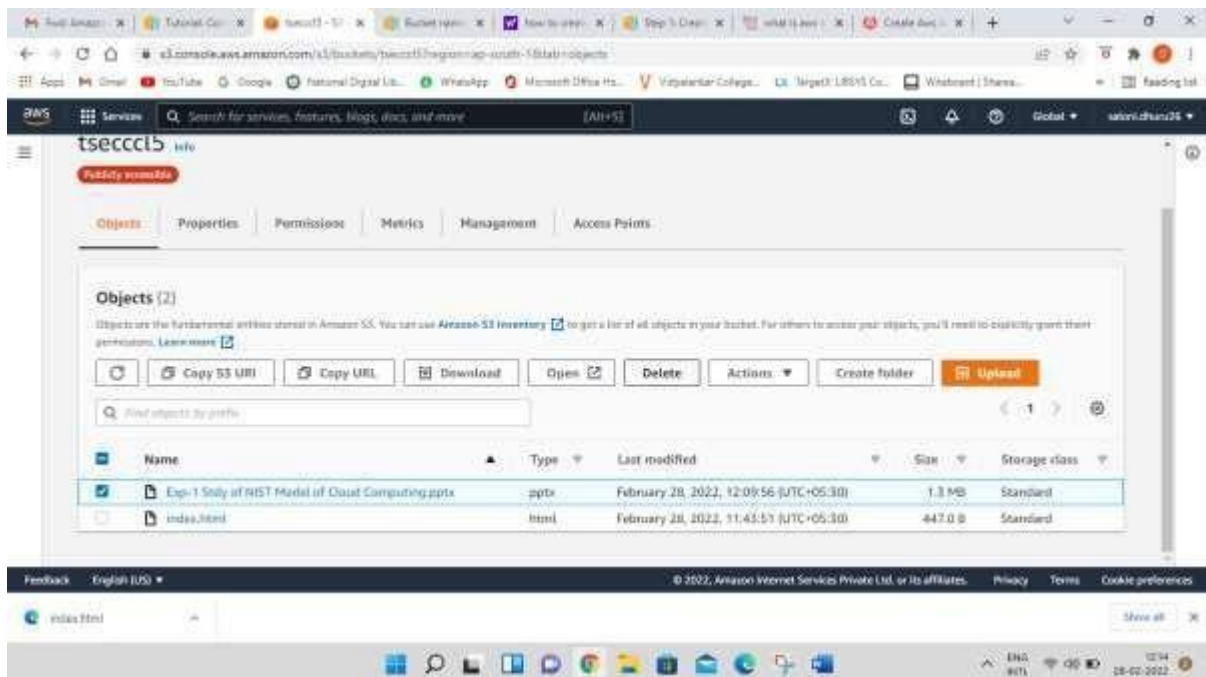
Scroll down and click on Save Changes button



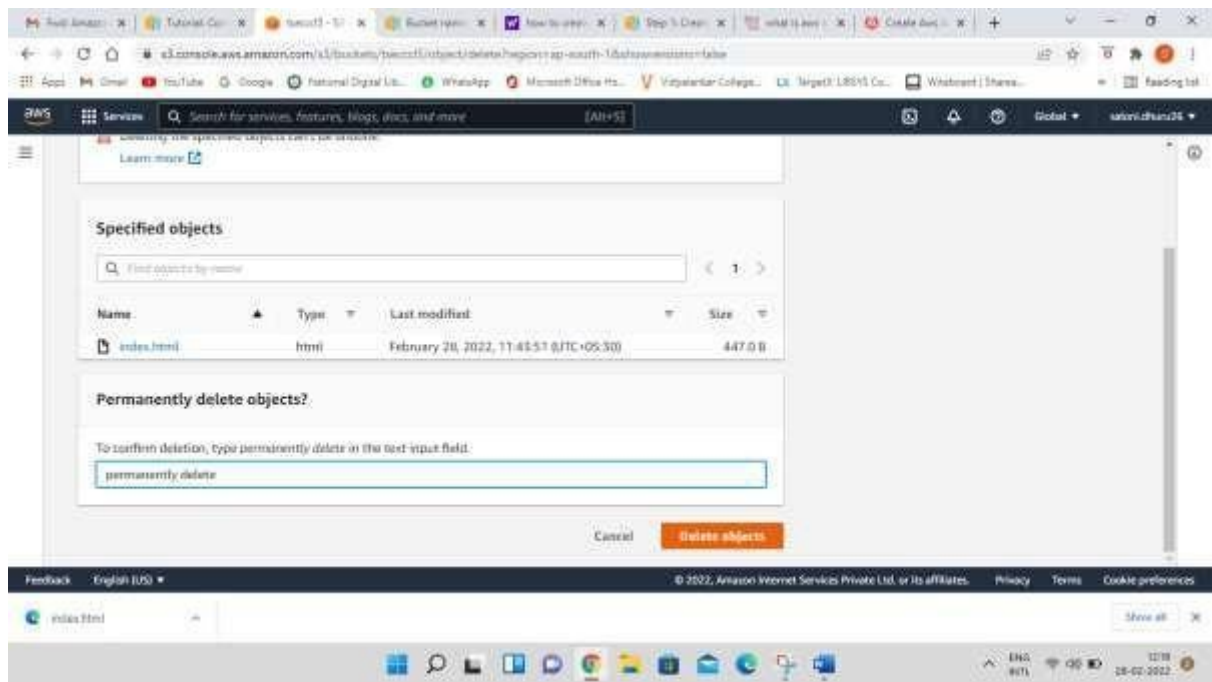
Step-16: open your html file and click on Object URL



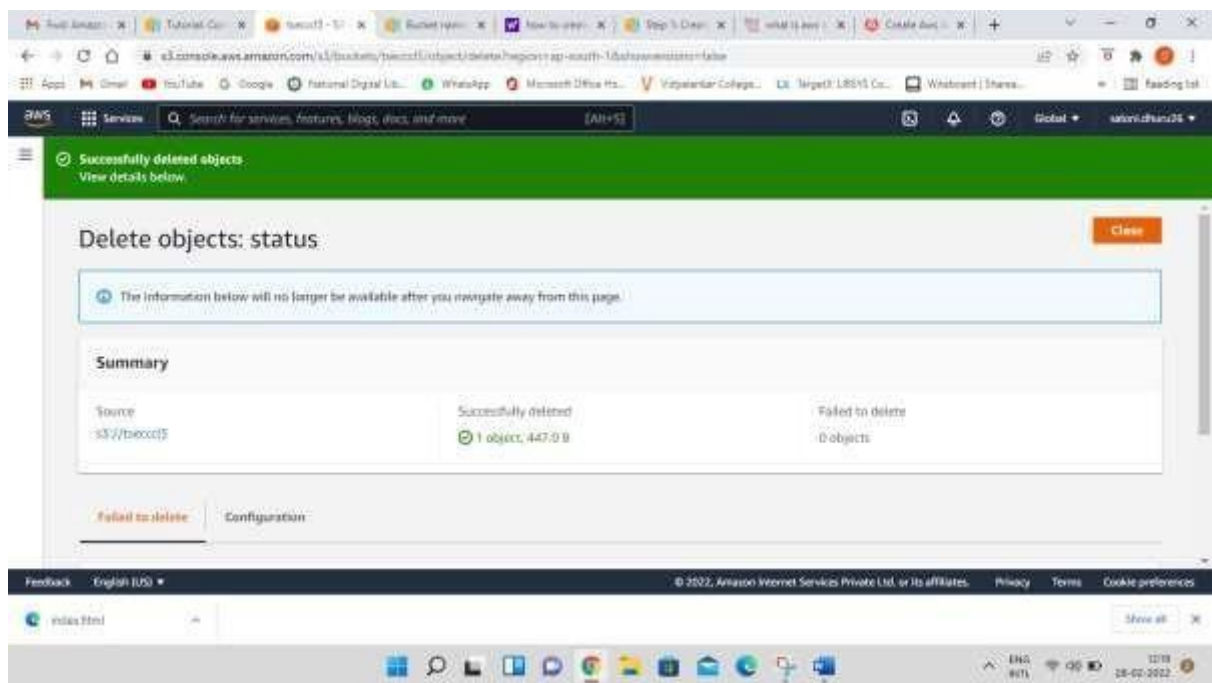
Step-17: Now for delete files click on checkbox of your file and then click on **Delete** Button



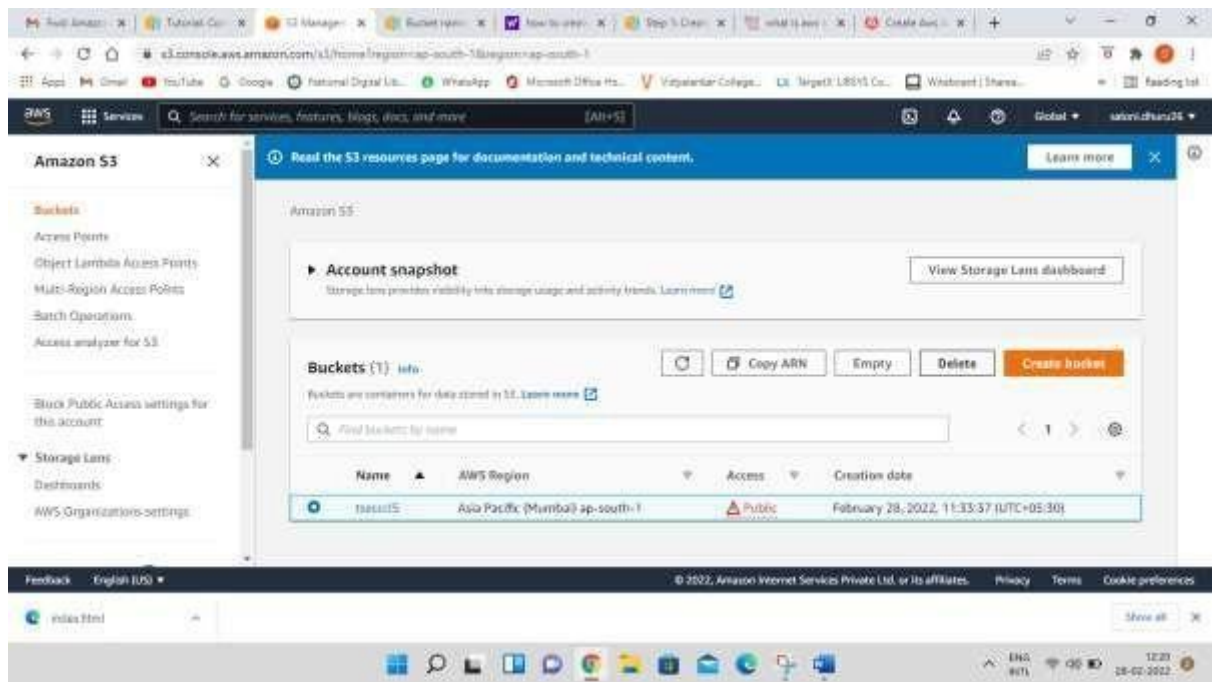
Write permanently delete and click on delete object button



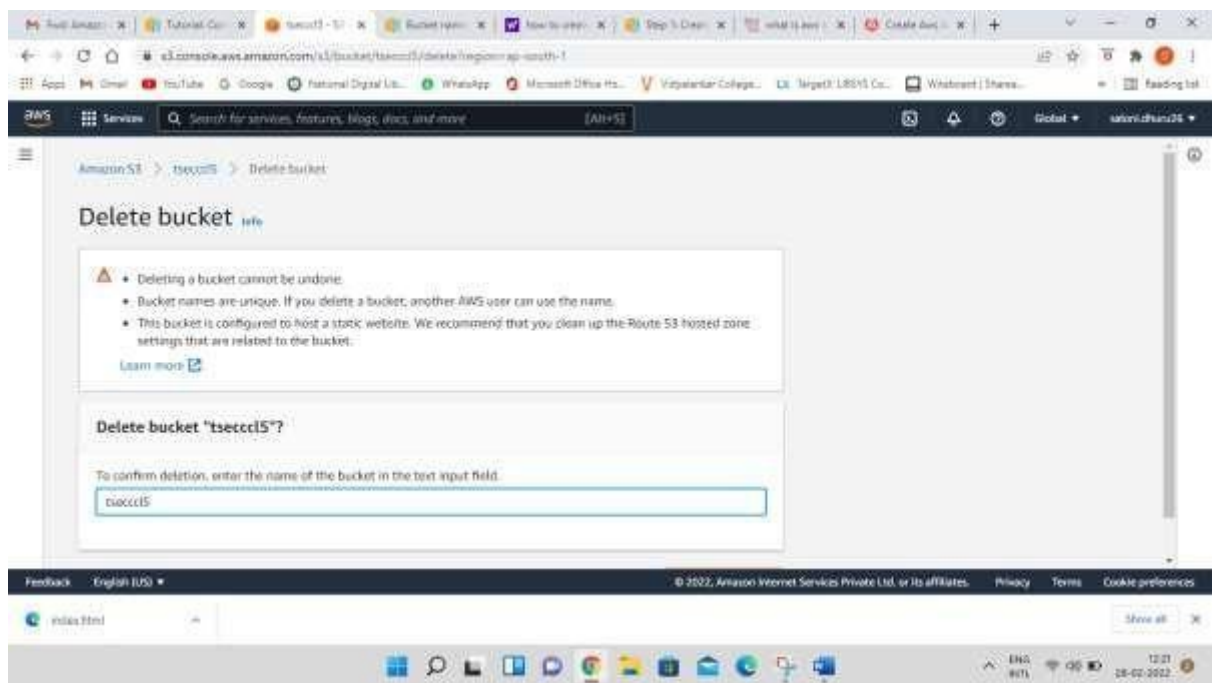
Now click on close button



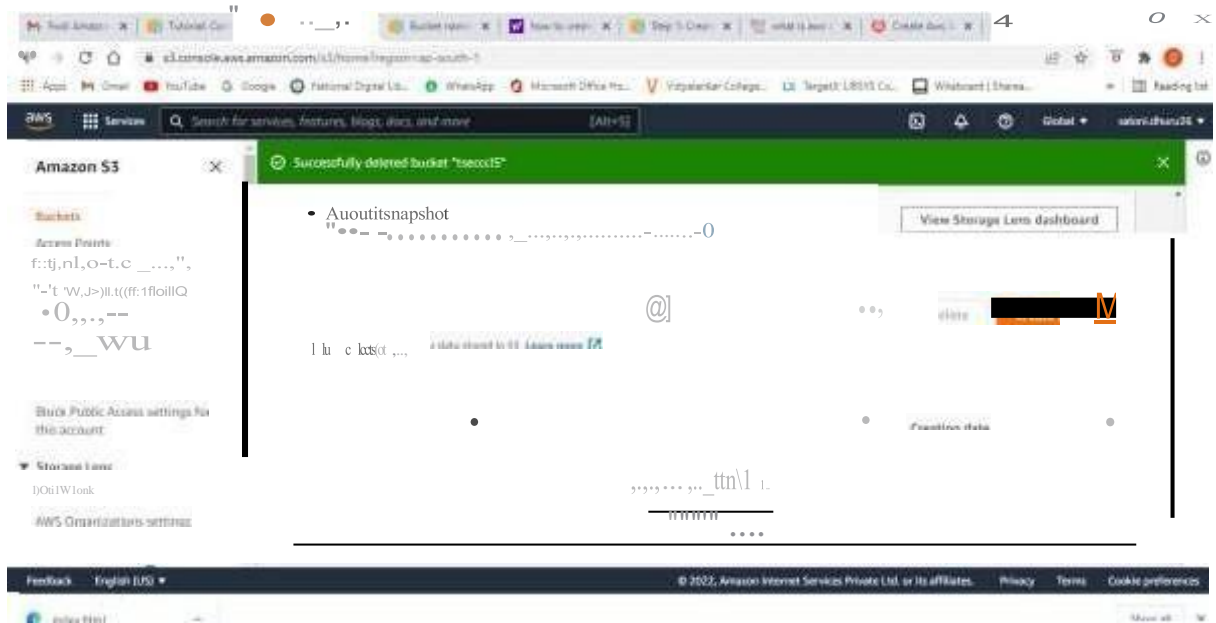
Step-18: now come to Amazon S3 tab and select your bucket and then click on delete button



Write down your bucket name in delete bucket tab and click on delete button at bottom right



You can see that the bucket is deleted



i1P1.CII0V



PART B

(PART B: TO BE COMPLETED BY STUDENTS)

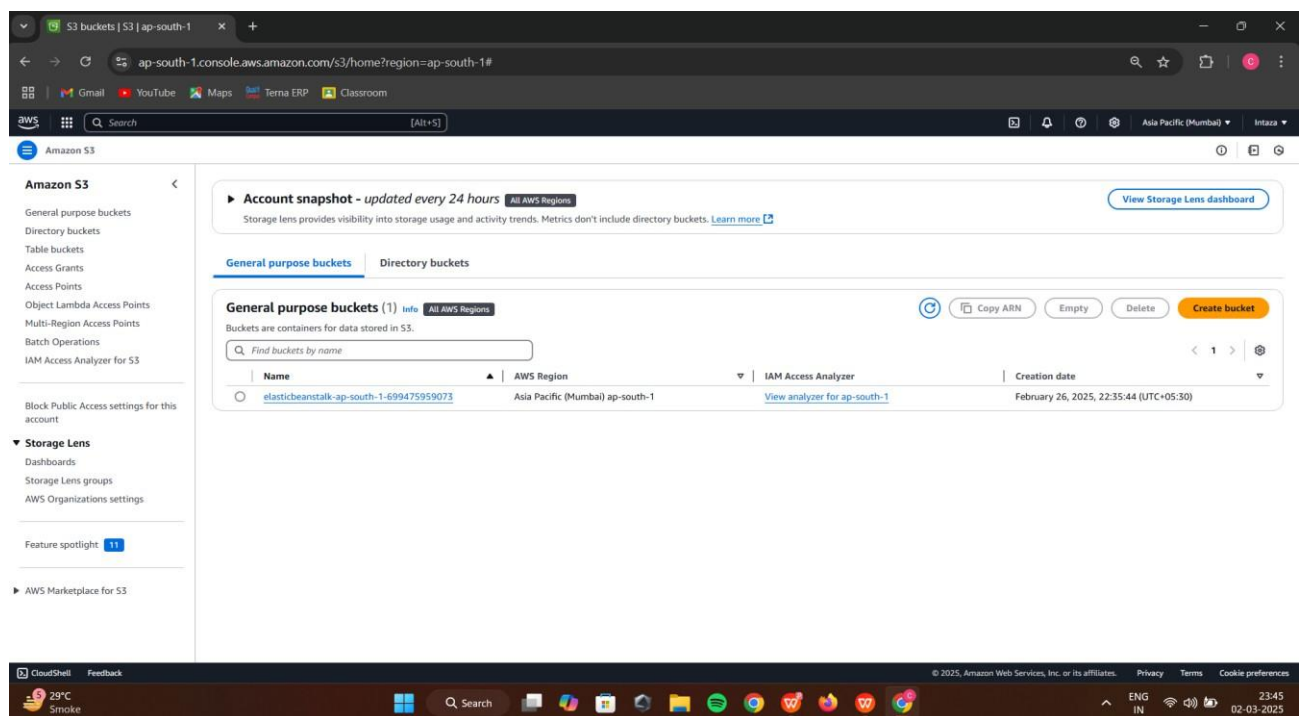
(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)

Roll No.B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch :B2
Date of Experiment:	Date of Submission:
Grade :	

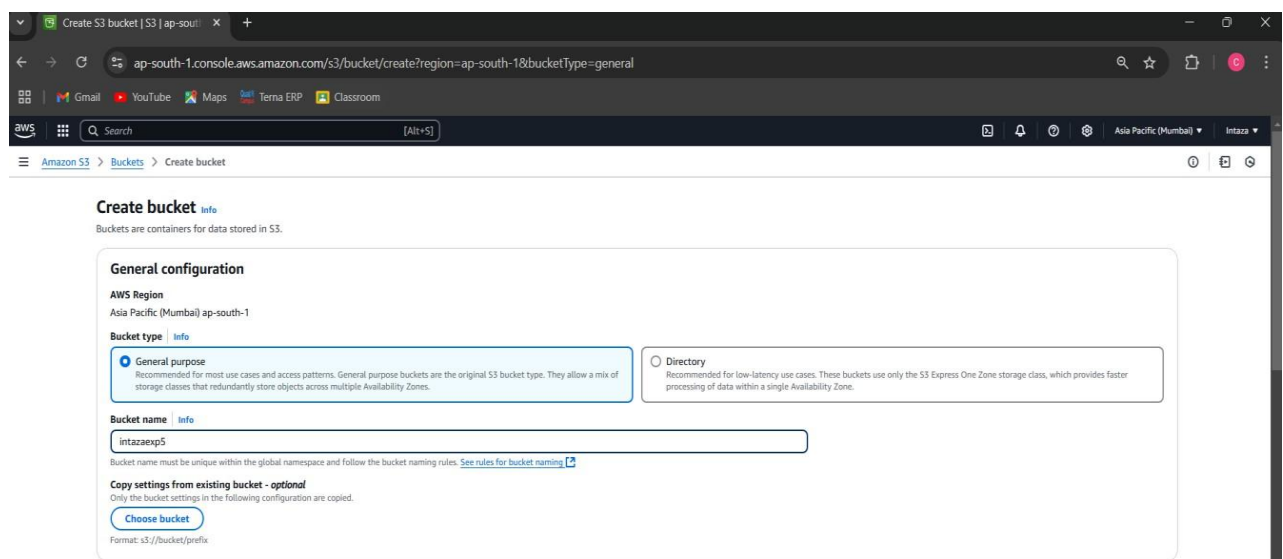
B.1 Question of Curiosity:

Q.1: Create Bucket using AWS S3 service (Add stepwise screenshots of the same)

Step-1: click on create bucket



Step-2: Give Bucket name & select region for storage



Step-3: Keep object ownership setting as ACLs Disabled as by-default

Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Step-4: Disable block all public access checkbox

Step-5: Select the checkbox for Turning off block all public access might result in this bucket and the objects within becoming public

Step-6: Keep bucket versioning as disabled and add tags if required.

Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
☒ **Disable**
☐ **Enable**

Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

Step-7: Keep default encryption disabled and click on create bucket button

Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
☒ **Server-side encryption with Amazon S3 managed keys (SSE-S3)**
☐ **Server-side encryption with AWS Key Management Service keys (SSE-KMS)**
☐ **Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)**
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing on the Amazon S3 pricing page](#).

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☒ **Disable**
☐ **Enable**

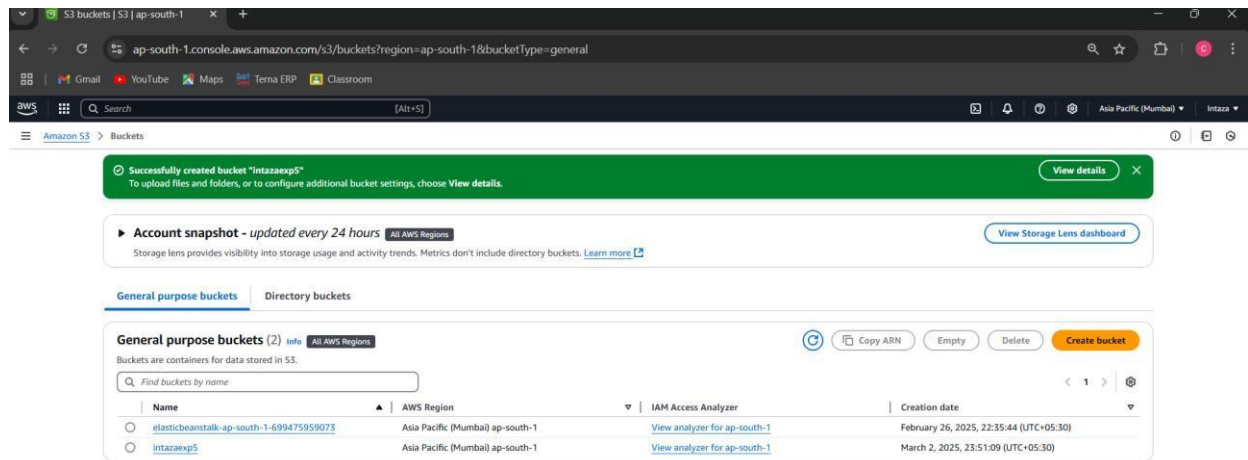
Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

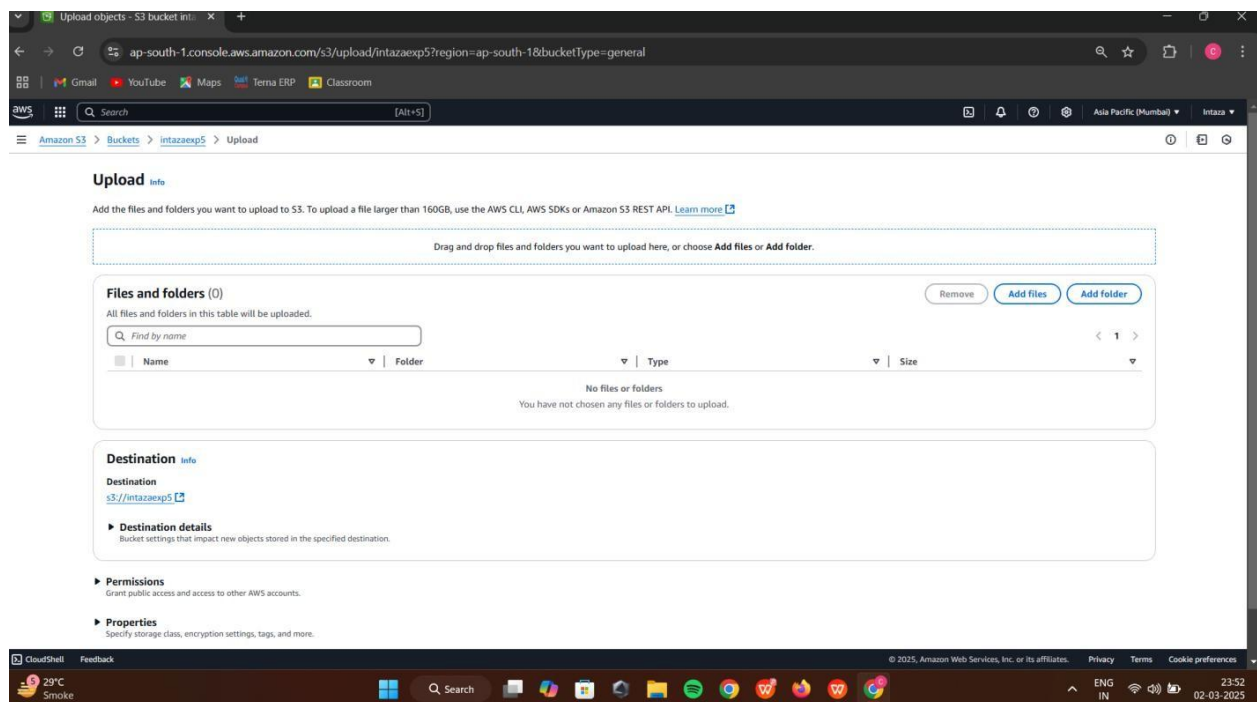
step-8: You can now see the successful creation of your bucket

Step-9: now click on the bucket that you have created

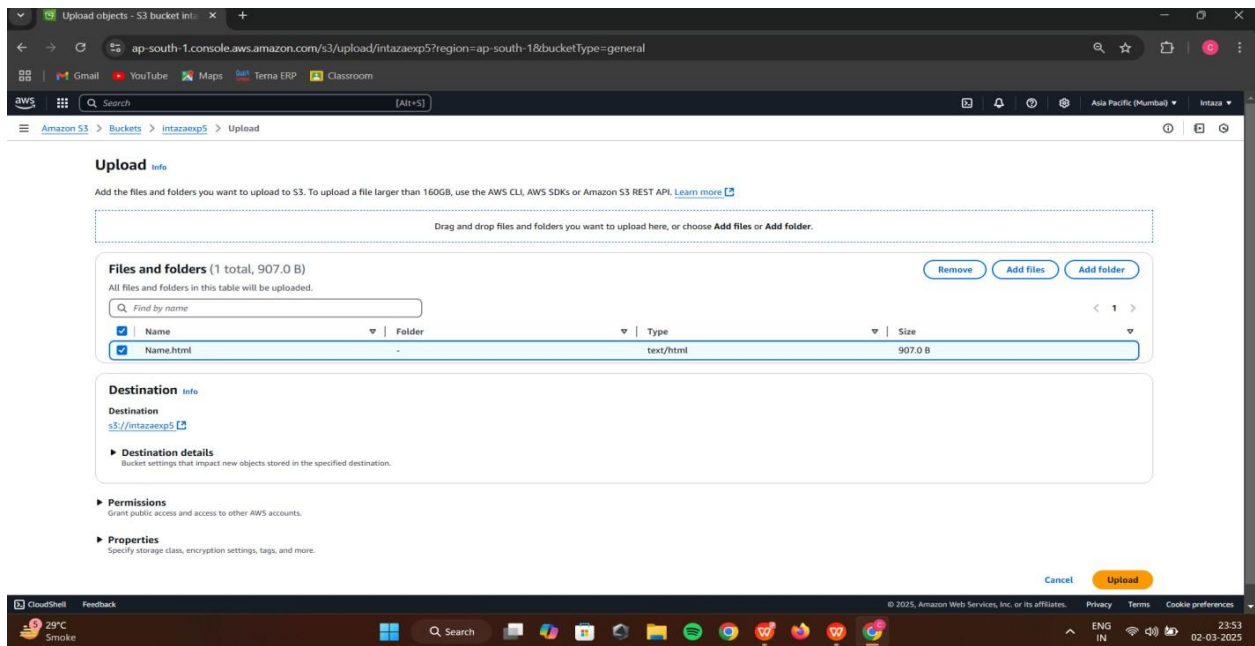


Q2: Add Objects to Bucket created (Add stepwise screenshots of the same)

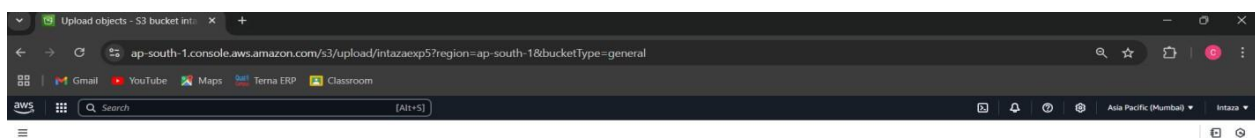
Step-10: You can either create a folder here or upload an existing file in the bucket



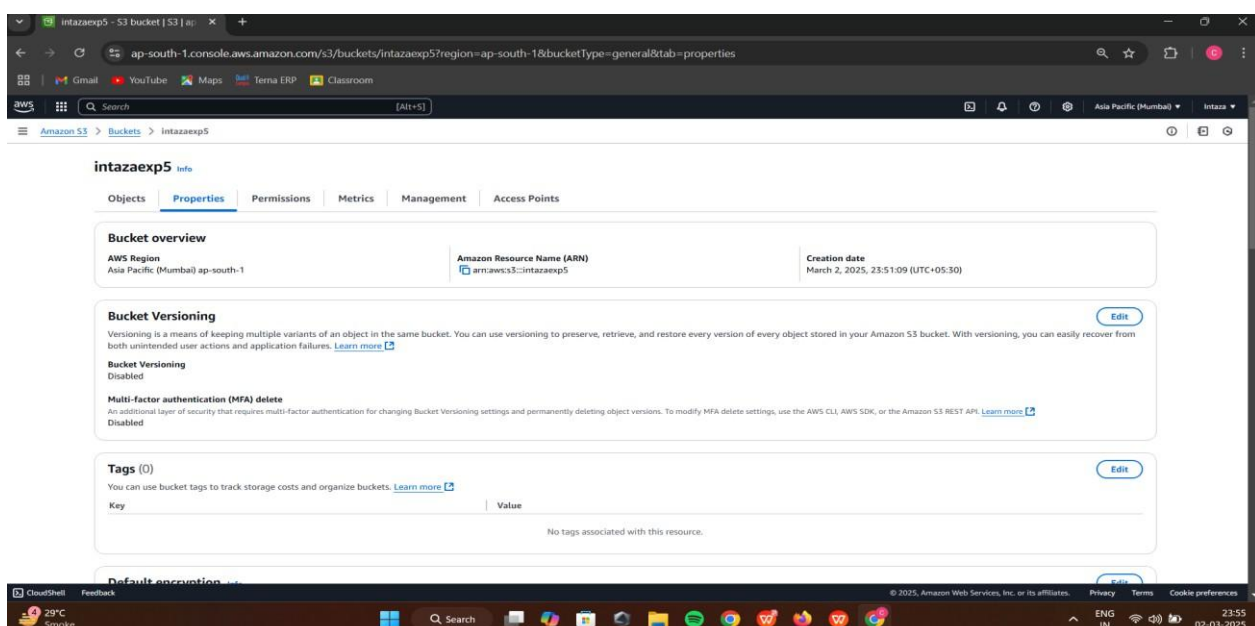
Step-10: now click on upload button and click on add files button browse your local machine and select which file you need to upload on S3 next click on upload button at bottom right end

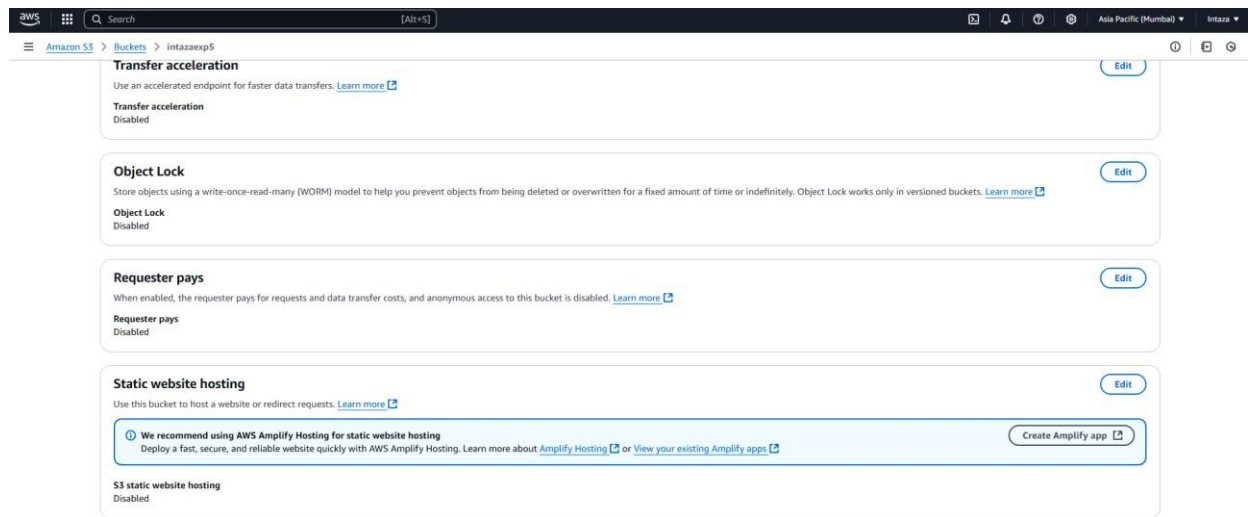


Step-11: Now you can check the upload status screen, Now click on close button.

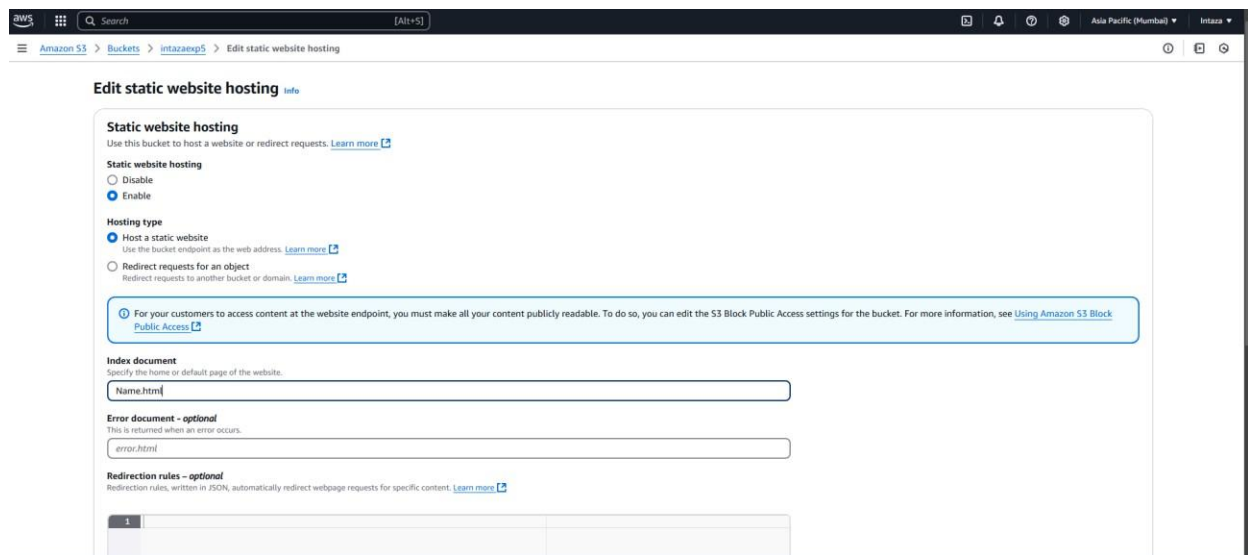


Step-12: Select properties and scroll down to Static website hosting option which is disabled now click on Edit option on right side

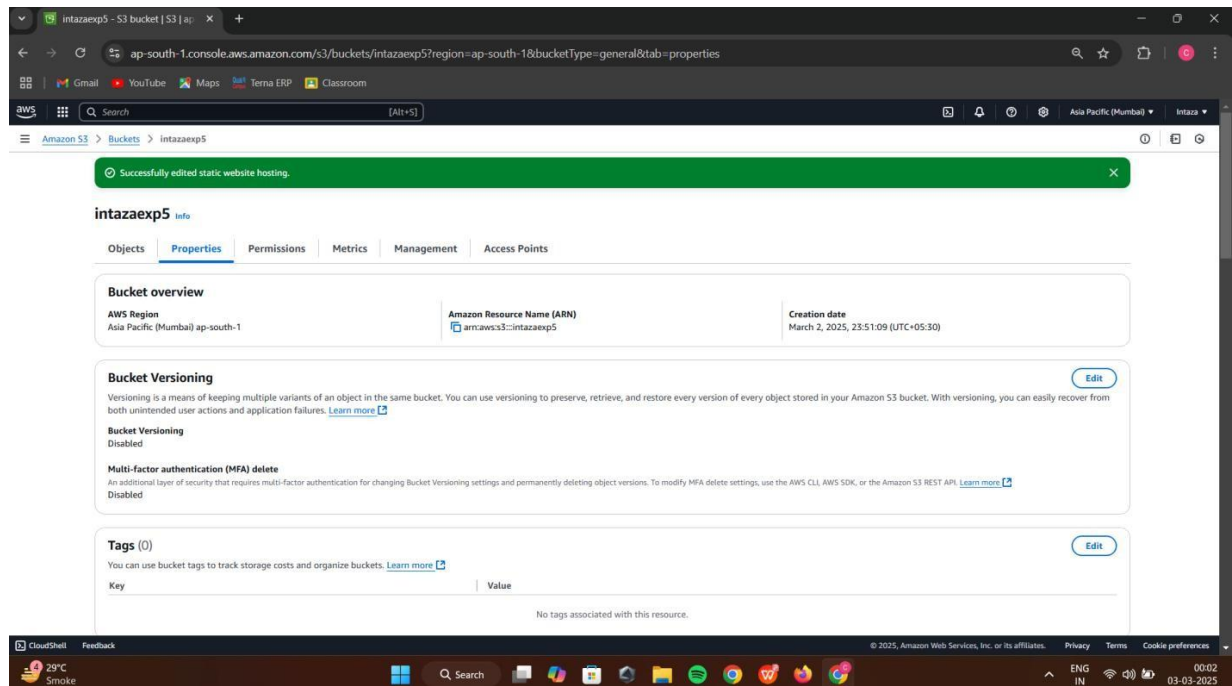




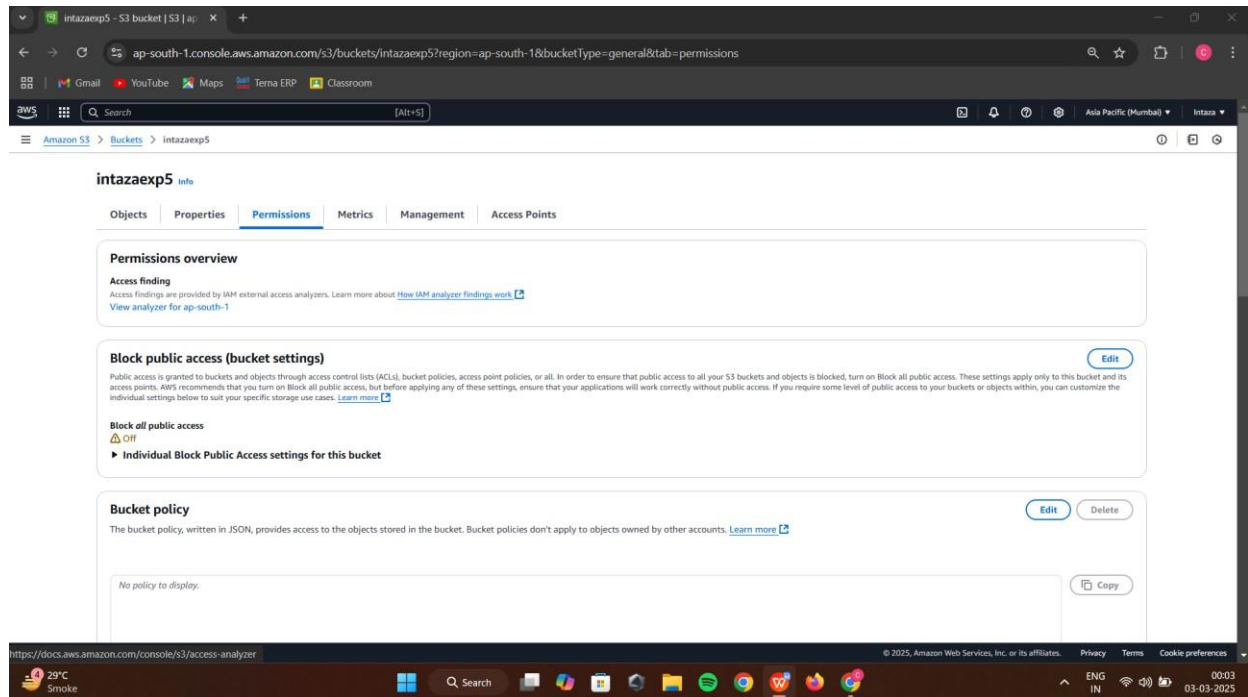
Step-13: Specify the file name in Index document which you have added in S3



Step-14: Scroll down and save the changes at bottom right, Following screen will appear



Step-15: Click on Permissions Tab

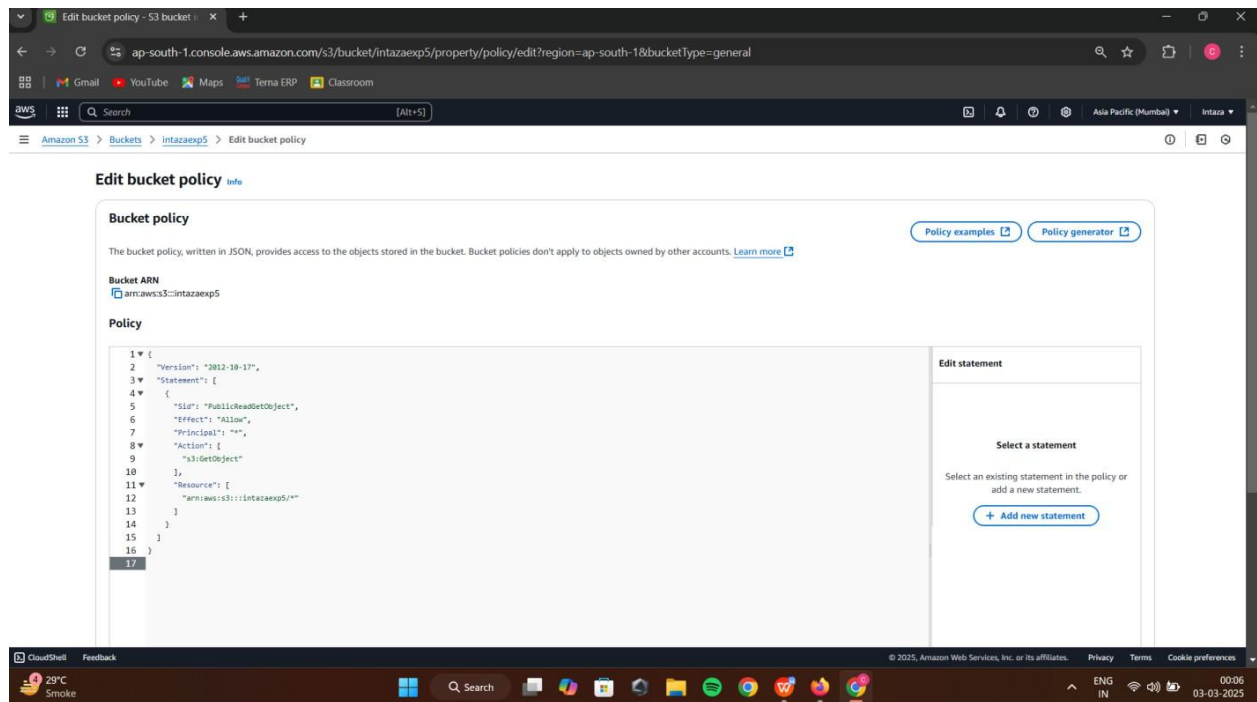


Step-16: In bucket policy click on Edit option

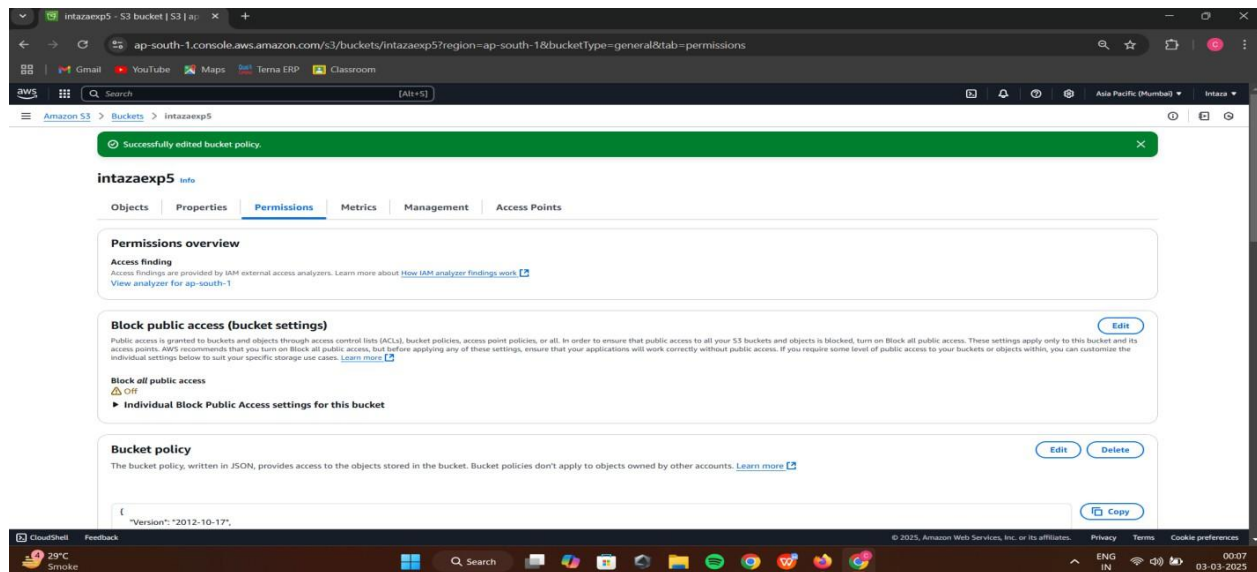
Step 17- after clicking on edit button paste the following code in bucket policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::Bucket-Name/*"
      ]
    }
  ]
}
```

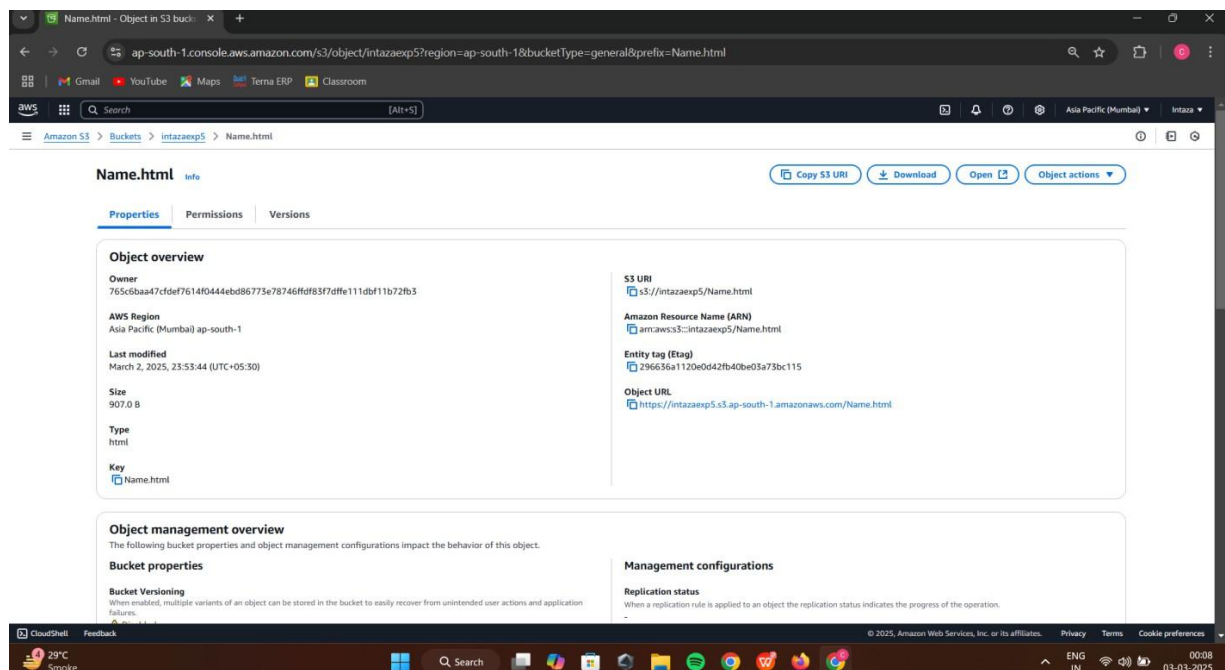
Note-Make sure that you add your bucket name in the code above

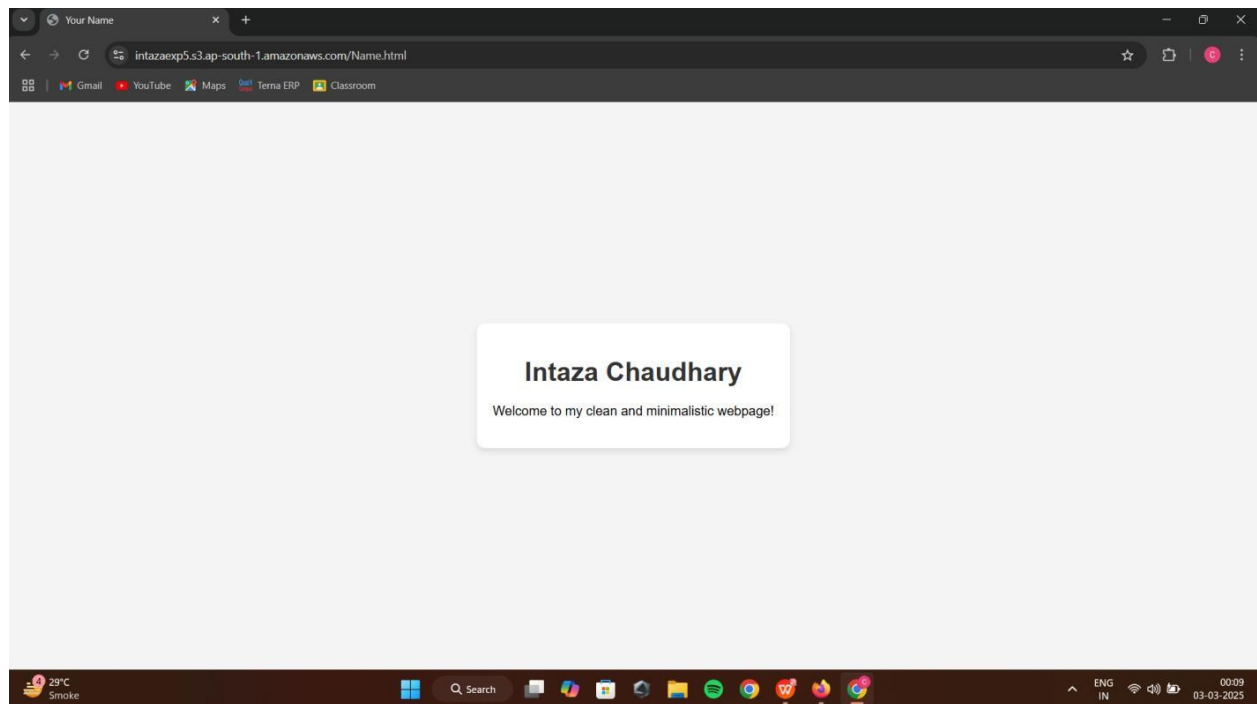


Scroll down and click on Save Changes button

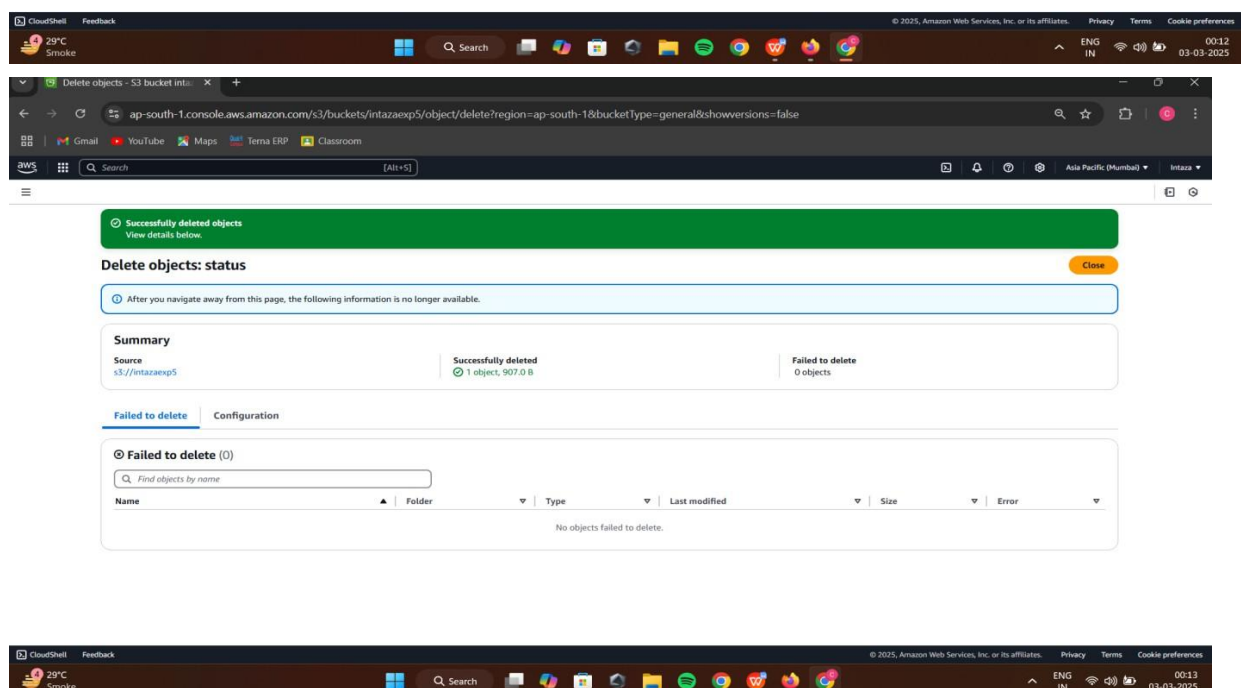
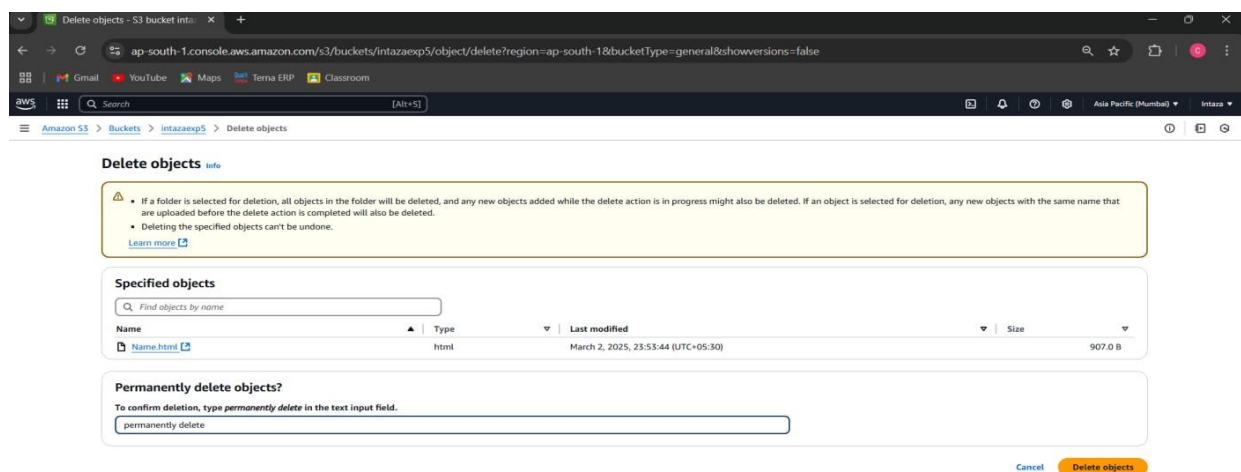


Step-18: open your html file and click on Object URL

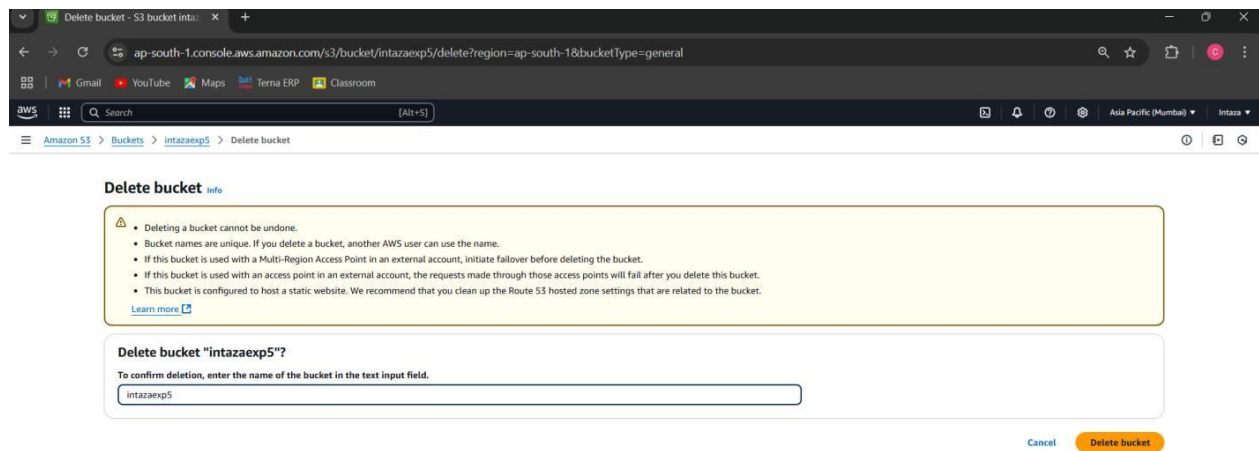




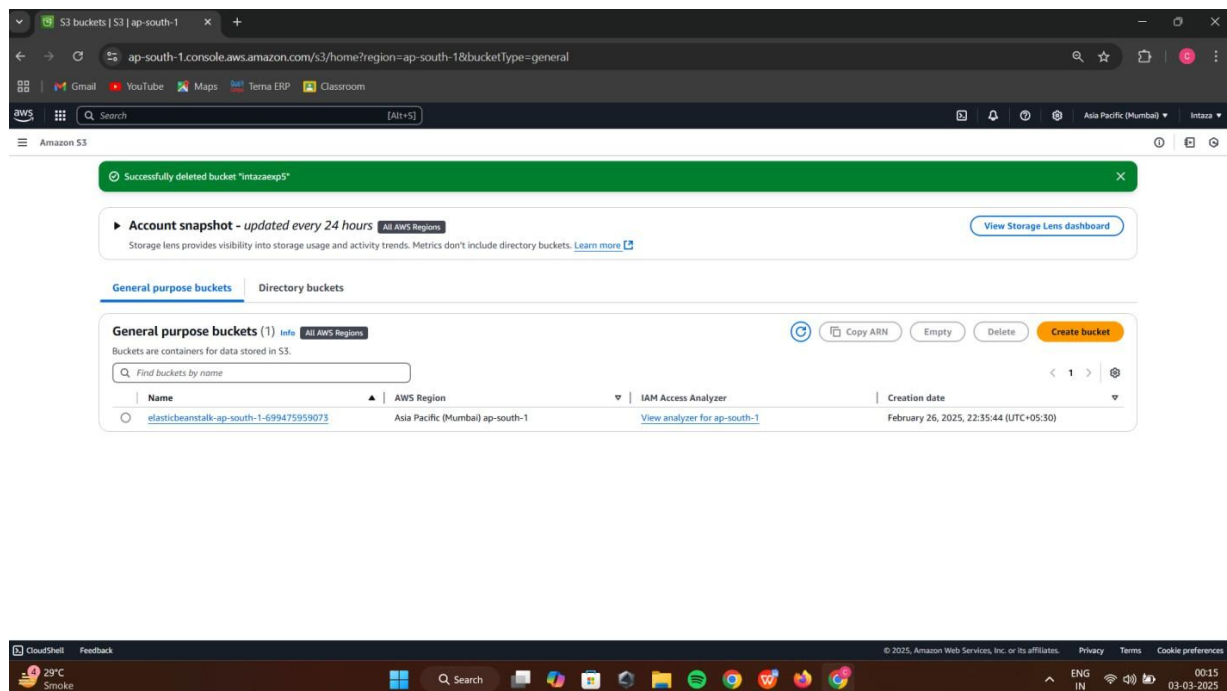
Step-19: Now for delete files click on checkbox of your file and then click on Delete Button, Write permanently delete and click on delete object button



Step-20: now come to Amazon S3 tab and select your bucket and then click on delete button, Write down your bucket name in delete bucket tab and click on delete button at bottom right



Step-21: You can see that the bucket is deleted



Q3: Compare Google drive with AWS S3

GOOGLE DRIVE	AMAZON S3
It is owned by Google LLC.	It is owned by Amazon.
It was launched in 2012.	It was launched in 2006.
It offers 15 GB free storage space.	It offers 5 GB free storage space.
It was developed by Google.	It was developed by Amazon Web Services(AWS).
The number of users using Google Drive is more.	The number of users using Amazon S3 is less.
It provides full security of data.	It also provides full security of data but comparatively less.
It has the maximum storage size of 30 TB.	It has the unlimited maximum storage size for paid users.
It does not support remote uploading.	Remote uploading is not supported here also.
Maximum file size in Google Drive is 5TB.	Here maximum file size is 5 TB.
It supports file versioning.	It also supports file versioning.

B.2 Conclusion:

In this experiment, we successfully demonstrated the implementation of **Storage as a Service (SaaS)** using **AWS S3**. Through this, students gained practical exposure to various AWS storage solutions, including **Amazon EBS, Amazon EFS, Amazon S3 Glacier, and AWS Storage Gateway**, understanding their use cases and benefits.