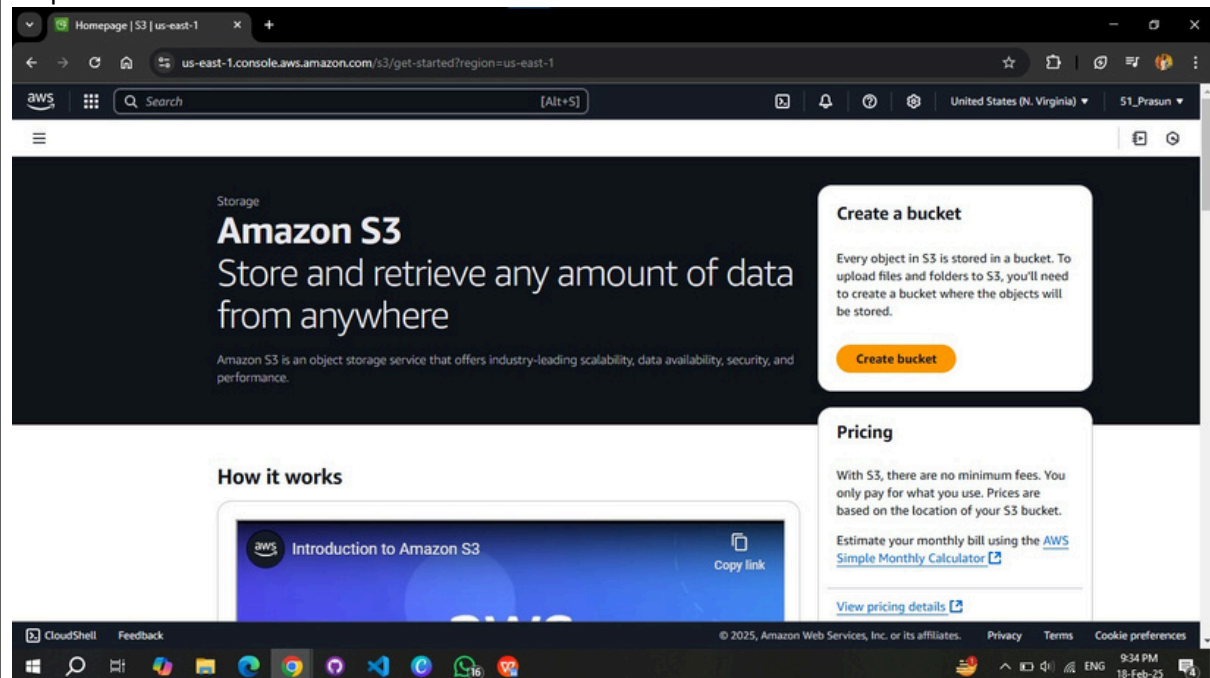


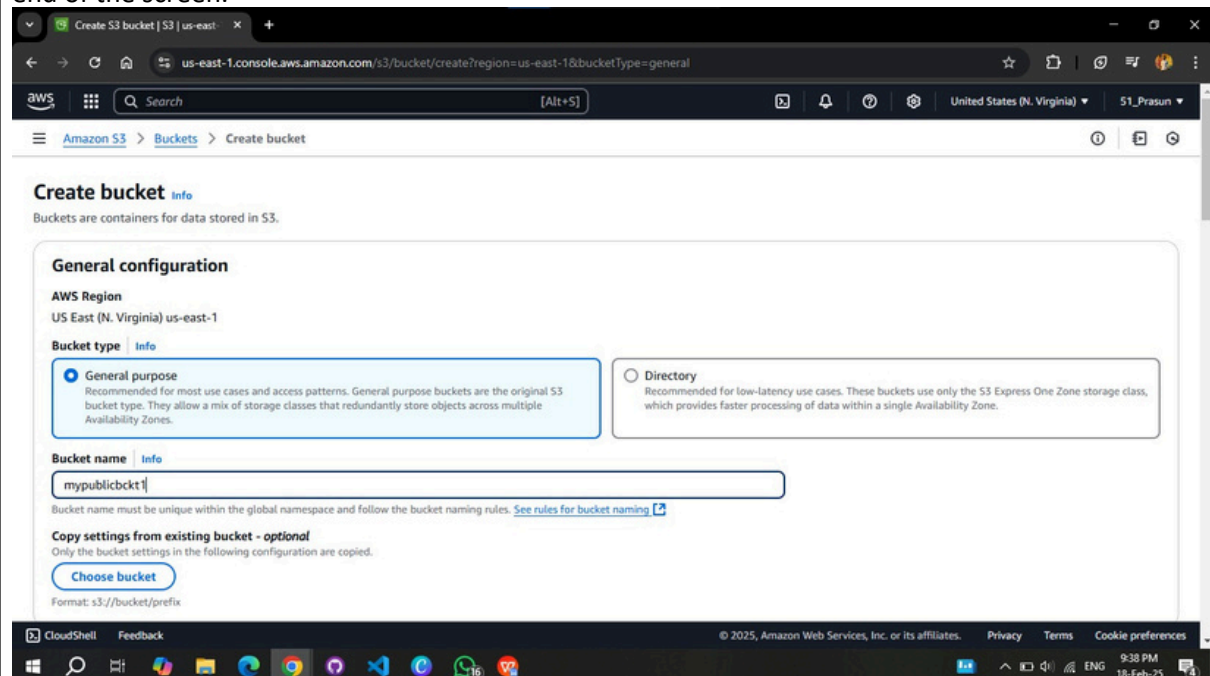
Assignment : 5

Create a public bucket in AWS. Upload a file and give the necessary permission to check whether the file URL is working.

Step 1: Visit Amazon S3 and click on "Create Bucket"



Step 2: In the Create bucket section, firstly name the bucket, in 'Object ownership' heading, select ACLs enabled option and keep 'Block all public access' box unchecked to keep the bucket public check the acknowledgement box. Leave all other options as default and click on Create Bucket at the end of the screen.



Create S3 bucket | S3 | us-east-1

us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general

Amazon S3 > Buckets > Create bucket

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.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

☒ Bucket owner preferred

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ Object writer

The object writer remains the object owner.

📘 If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

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

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

9:35 PM 18-Feb-25

Create S3 bucket | S3 | us-east-1

us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general

Amazon S3 > Buckets > Create bucket

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

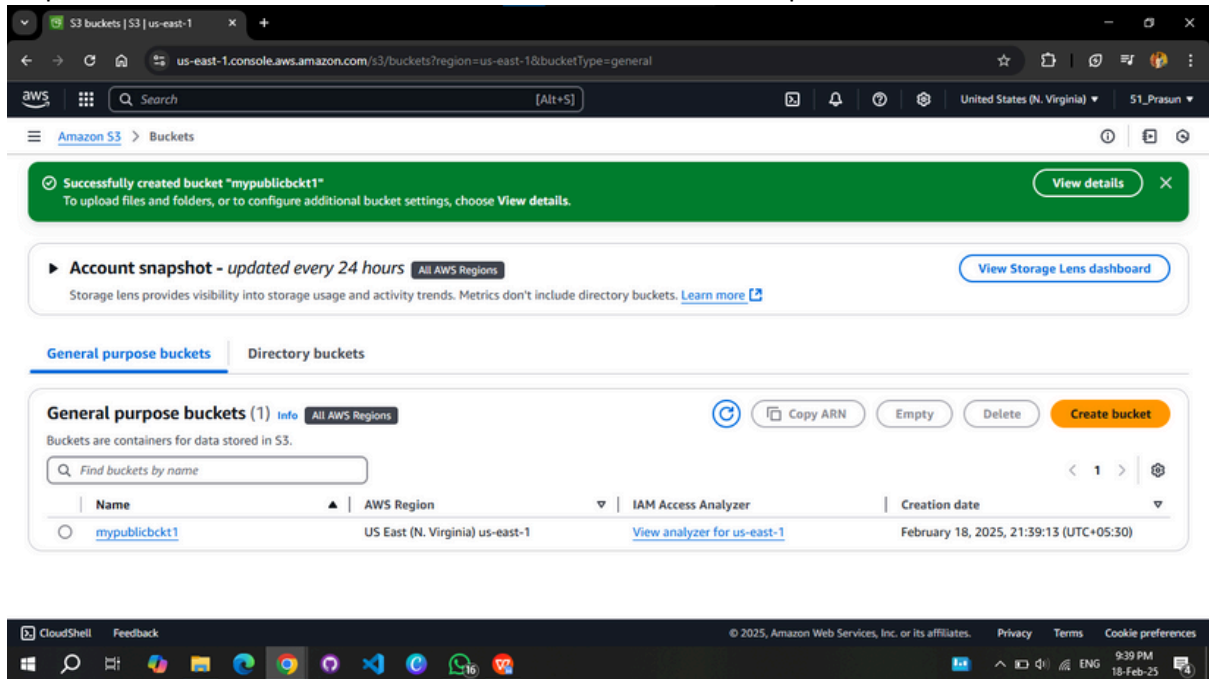
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

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

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

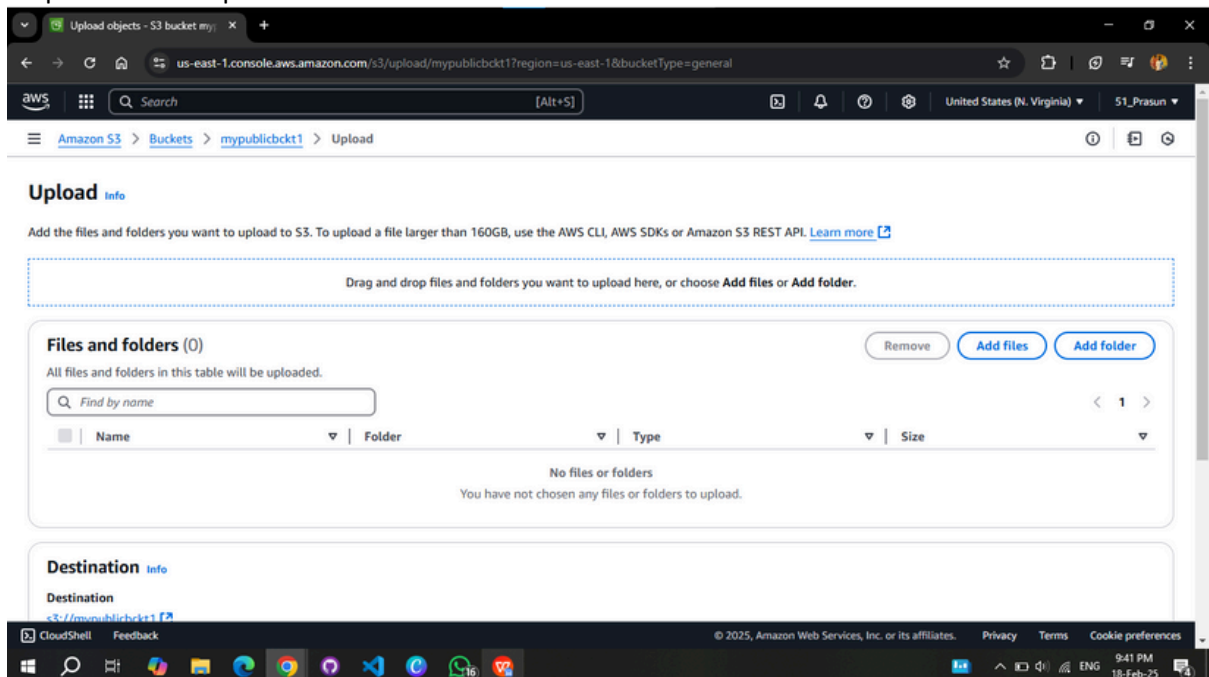
9:37 PM 18-Feb-25

Step 3: After creation of the bucket click on the bucket name to open it.



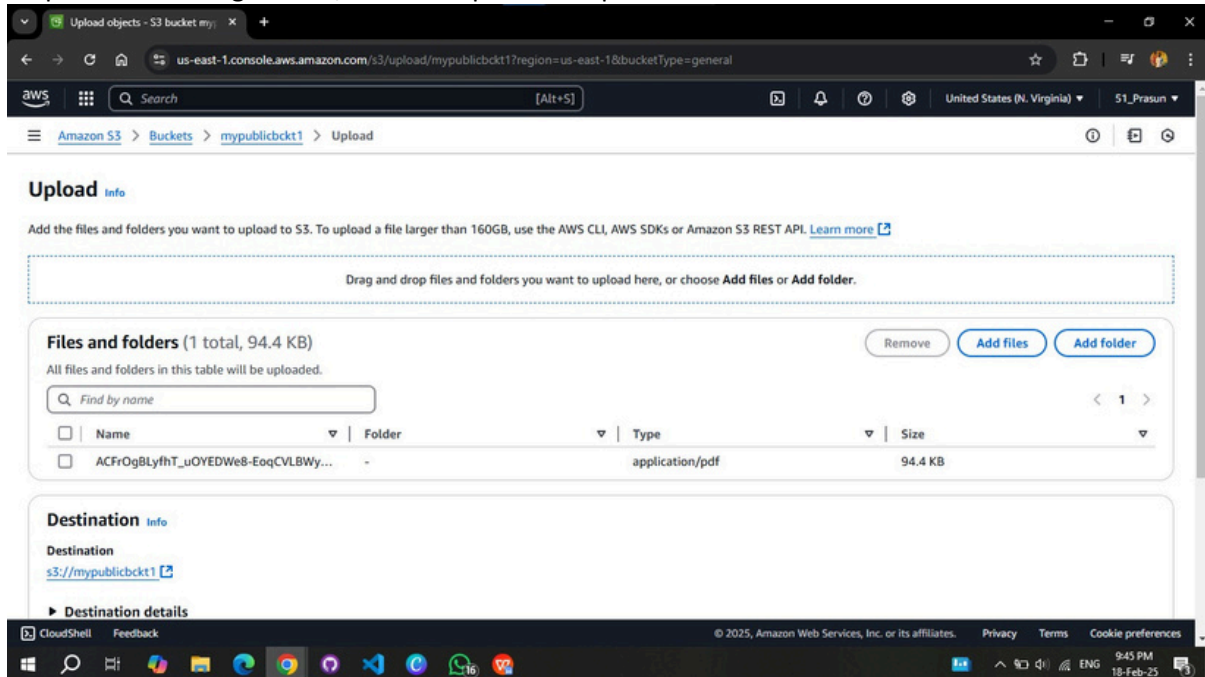
The screenshot shows the AWS Management Console interface for the 'Buckets' section. A green notification banner at the top indicates that the bucket 'mypublicbckt1' was successfully created. Below this, there's a section for 'General purpose buckets' with a table listing the bucket 'mypublicbckt1' in the 'US East (N. Virginia) us-east-1' region, created on February 18, 2025. The bucket name 'mypublicbckt1' is highlighted with a blue link.

Step 4: Click on 'Upload' button. And then click on 'Add files' button and add a file.



The screenshot shows the AWS Management Console interface for the 'Upload' page of the bucket 'mypublicbckt1'. The page has a section for 'Files and folders (0)' with a table that is currently empty. Below this, there's a 'Destination' section showing the path 's3://mypublicbckt1'.

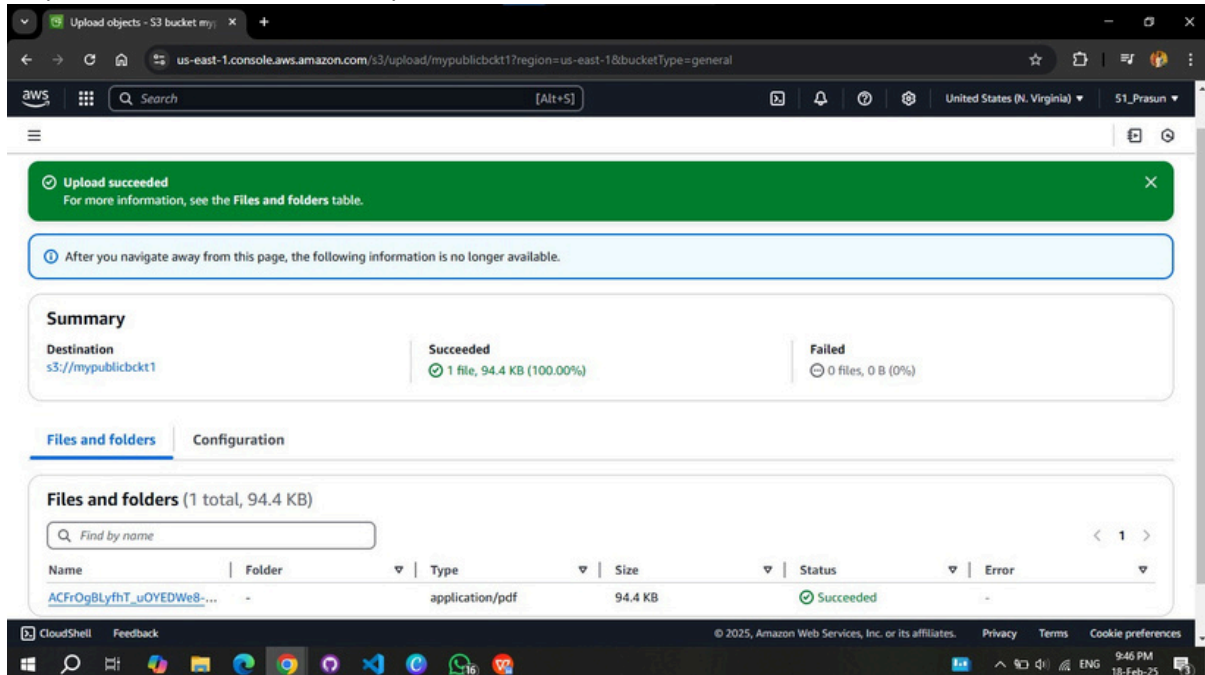
Step 5: After adding the file, click on 'Upload' to upload the selected file.



The screenshot shows the AWS S3 Upload console. At the top, there's a breadcrumb trail: Amazon S3 > Buckets > mypublicbckt1 > Upload. Below this, the 'Upload' section has an 'Info' tab and instructions: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)'. A dashed box indicates where to drag and drop files. Below that, a section titled 'Files and folders (1 total, 94.4 KB)' shows a table with one file: 'ACFrOgBlyfhT_uOYEDWe8-EoqCVLBWy...' of type 'application/pdf' and size '94.4 KB'. There are 'Remove', 'Add files', and 'Add folder' buttons. The 'Destination' section shows 's3://mypublicbckt1' and 'Destination details'.

Name	Folder	Type	Size
ACFrOgBlyfhT_uOYEDWe8-EoqCVLBWy...	-	application/pdf	94.4 KB

Step 6: Click on close after the upload is successful.



The screenshot shows the AWS S3 Upload console after a successful upload. A green banner at the top says 'Upload succeeded' with a close button. Below it, a blue box states: 'After you navigate away from this page, the following information is no longer available.' The 'Summary' section shows 'Destination: s3://mypublicbckt1', 'Succeeded: 1 file, 94.4 KB (100.00%)', and 'Failed: 0 files, 0 B (0%)'. The 'Files and folders' tab is active, showing a table with one file: 'ACFrOgBlyfhT_uOYEDWe8-EoqCVLBWy...' of type 'application/pdf', size '94.4 KB', and status 'Succeeded'.

Name	Folder	Type	Size	Status	Error
ACFrOgBlyfhT_uOYEDWe8-EoqCVLBWy...	-	application/pdf	94.4 KB	Succeeded	-

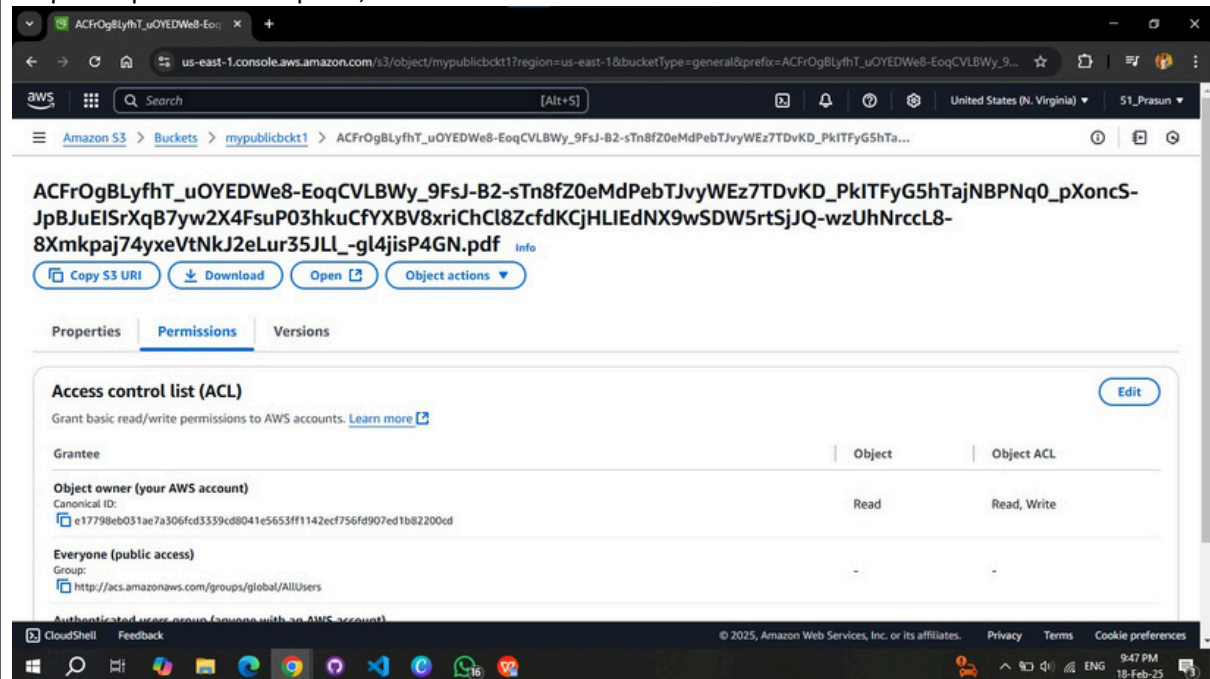
Step 7: Click the name of the selected file to edit its permissions.

The screenshot shows the AWS console interface for the 'mypublicbckt1' bucket. The 'Objects' tab is active, displaying a list of objects. The selected object is a PDF file with the name 'ACFrOgBlyfhT_uOYEDWe8-EoqCVLBWY_9FsJ-B2-sTn8fZ0eMdPebTJvyWEz7TDvKD_PkITfyG5hTajNBPNq0_pXoncS-JpBJuEISrXqB7yw2X4FsuP03hkuCFYXBV8xriChCl8ZcfdKCjHLIEdNX9wSDW5rtSjJQ-wzUHNrccL8-8Xmkpaj74yxeVtNkJ2eLur35JLL_-gl4jisP4GN.pdf'. The 'Actions' button is highlighted, indicating the next step is to click the object name to edit its permissions.

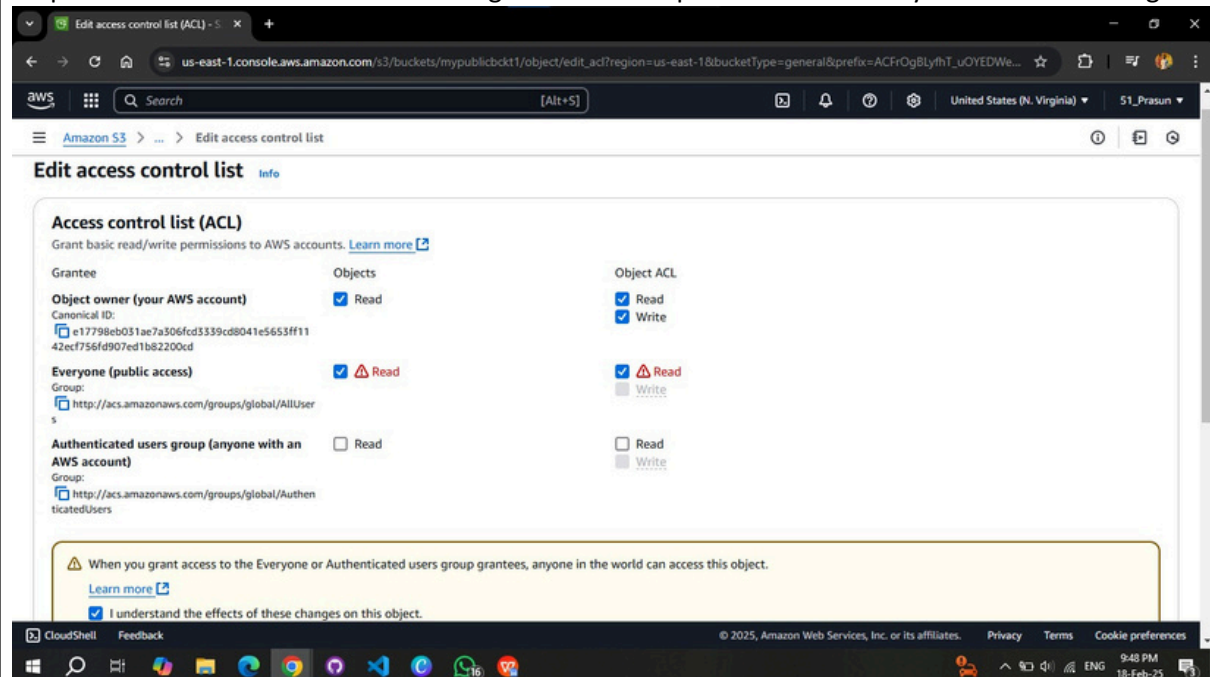
Step 8: Then go to Permissions option.

The screenshot shows the AWS console interface for the selected object. The 'Permissions' tab is active, displaying the 'Object overview' section. The details shown are: Owner: prasunkr.B43, AWS Region: US East (N. Virginia) us-east-1, Last modified: February 18, 2025, 21:46:30 (UTC+05:30), and Size: 94.4 KB. The 'Object overview' section is visible, showing details like Owner, AWS Region, Last modified, and Size.

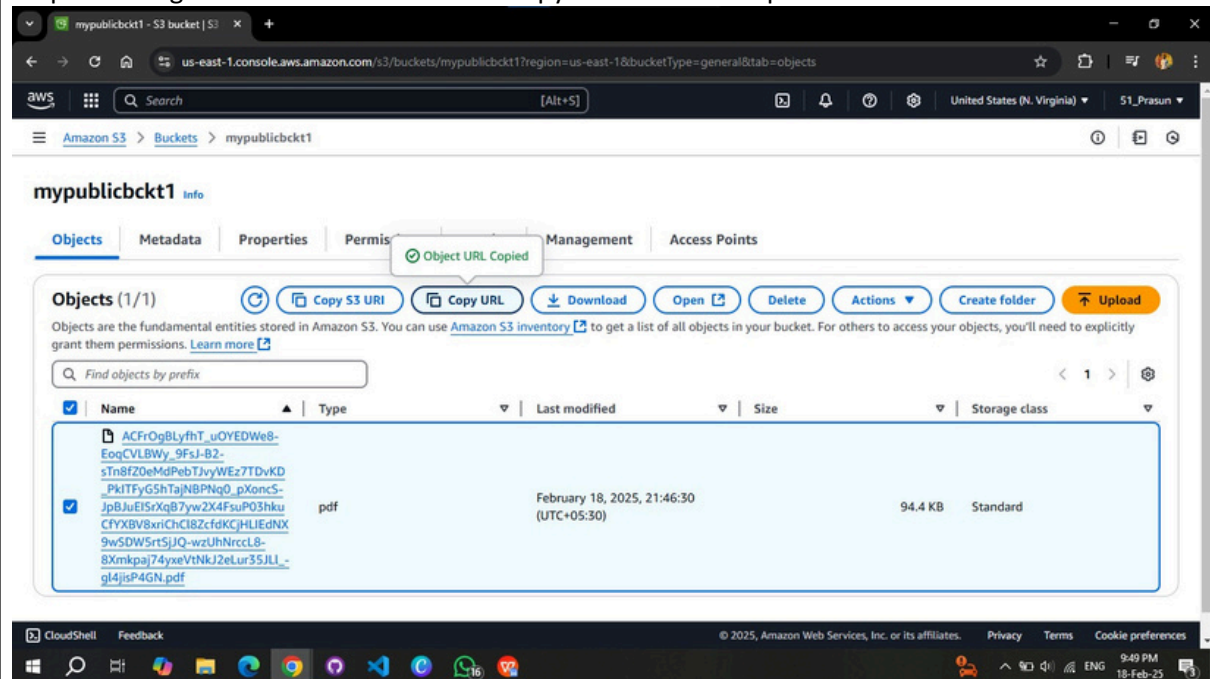
Step 9: In permissions option, click on edit.



Step 10: Edit the Access control list and give read-write permissions to everyone and save changes.



Step 11: Navigate back to the bucket and copy the URL of the uploaded file.



mypublicbckt1 info

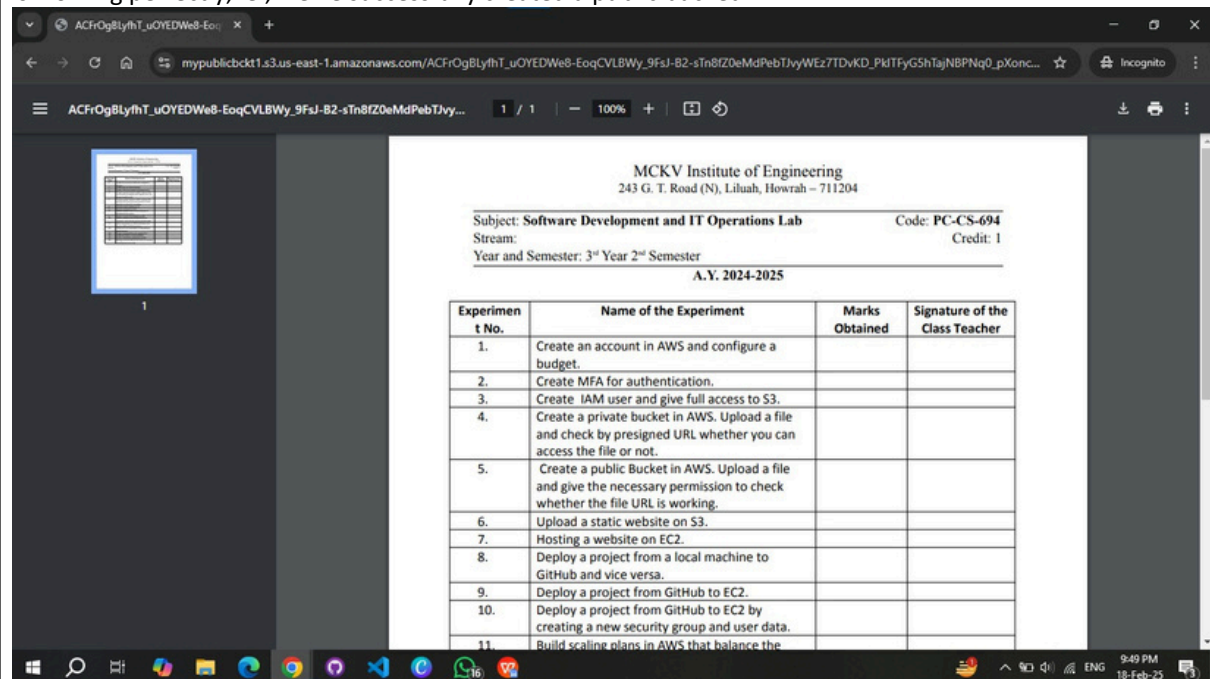
Objects (1/1)

Object URL Copied

Find objects by prefix

Name	Type	Last modified	Size	Storage class
ACFrOgBlythT_uOYEDWe8-EoqCvLBWY_9FsJ-B2-sTn8fZ0eMdPebTjvyWEz7TDvKD_PkITfyGShTajNBPnq0_pXoncS-JpBjuEISrKqB7yw2X4FsuP03hkuCFYXBV8xriChC18ZcfdKCJHLIEdNX9wSDW5rt5jJQ-wzUhnrccl8-8Xmkpaj74yxeVtNk12eLur35JLI-gl4jisp4GN.pdf	pdf	February 18, 2025, 21:46:30 (UTC+05:30)	94.4 KB	Standard

Step 12: Paste the copied URL in a new window or browser to check if the URL is working or not. Since the URL is working perfectly, ie., we've successfully created a public bucket.



MCKV Institute of Engineering
243 G. T. Road (N), Lilaah, Howrah - 711204

Subject: Software Development and IT Operations Lab Code: PC-CS-694
Stream: Credit: 1
Year and Semester: 3rd Year 2nd Semester
A.Y. 2024-2025

Experiment No.	Name of the Experiment	Marks Obtained	Signature of the Class Teacher
1.	Create an account in AWS and configure a budget.		
2.	Create MFA for authentication.		
3.	Create IAM user and give full access to S3.		
4.	Create a private bucket in AWS. Upload a file and check by presigned URL whether you can access the file or not.		
5.	Create a public Bucket in AWS. Upload a file and give the necessary permission to check whether the file URL is working.		
6.	Upload a static website on S3.		
7.	Hosting a website on EC2.		
8.	Deploy a project from a local machine to GitHub and vice versa.		
9.	Deploy a project from GitHub to EC2.		
10.	Deploy a project from GitHub to EC2 by creating a new security group and user data.		
11.	Build scaling plans in AWS that balance the		