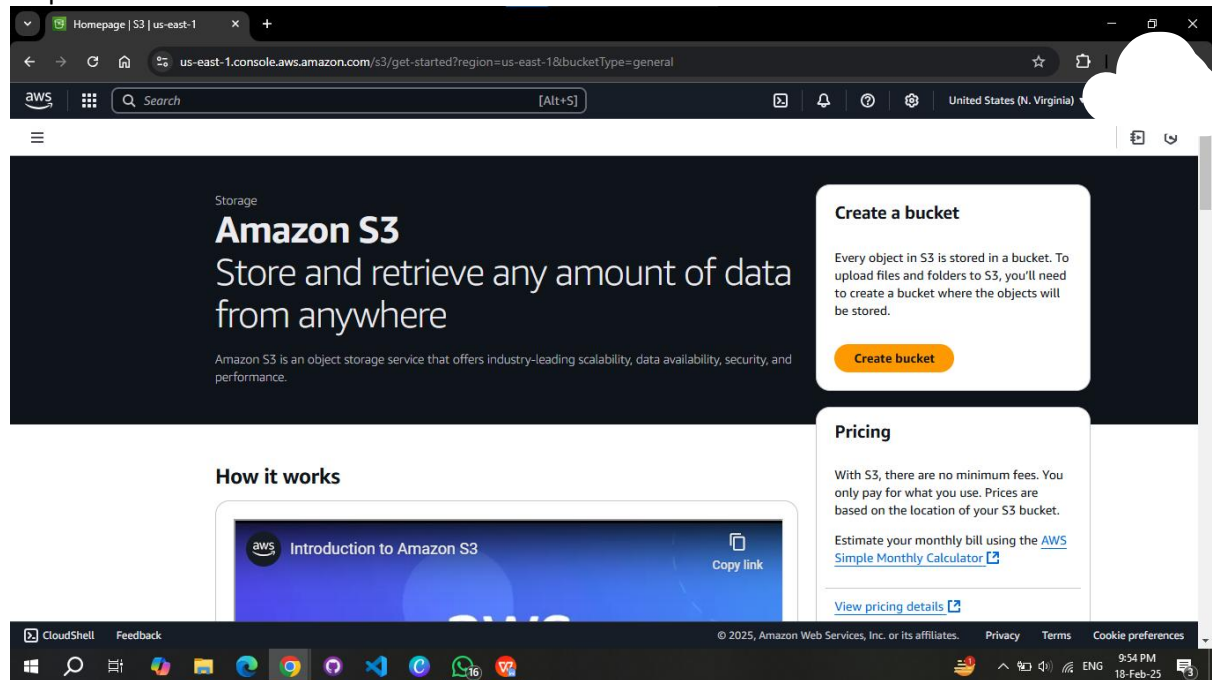


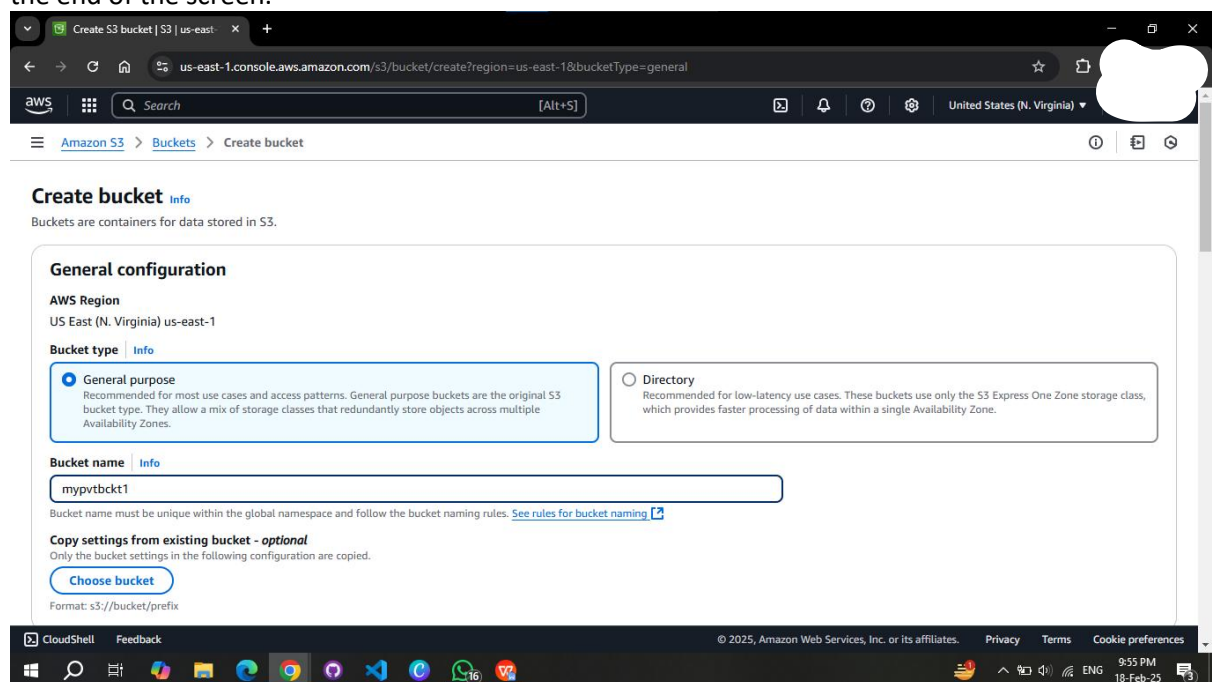
## Assignment : 4

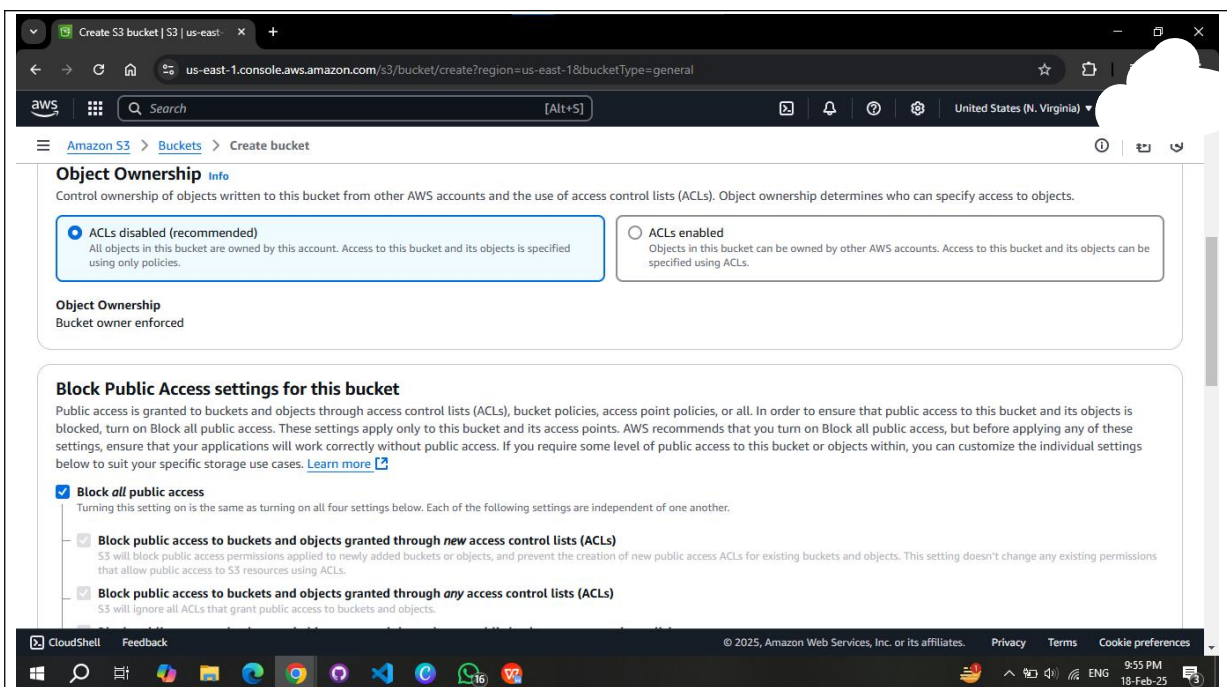
**Create a private bucket in AWS. Upload a file and check by presigned URL whether you can access the file or not.**

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

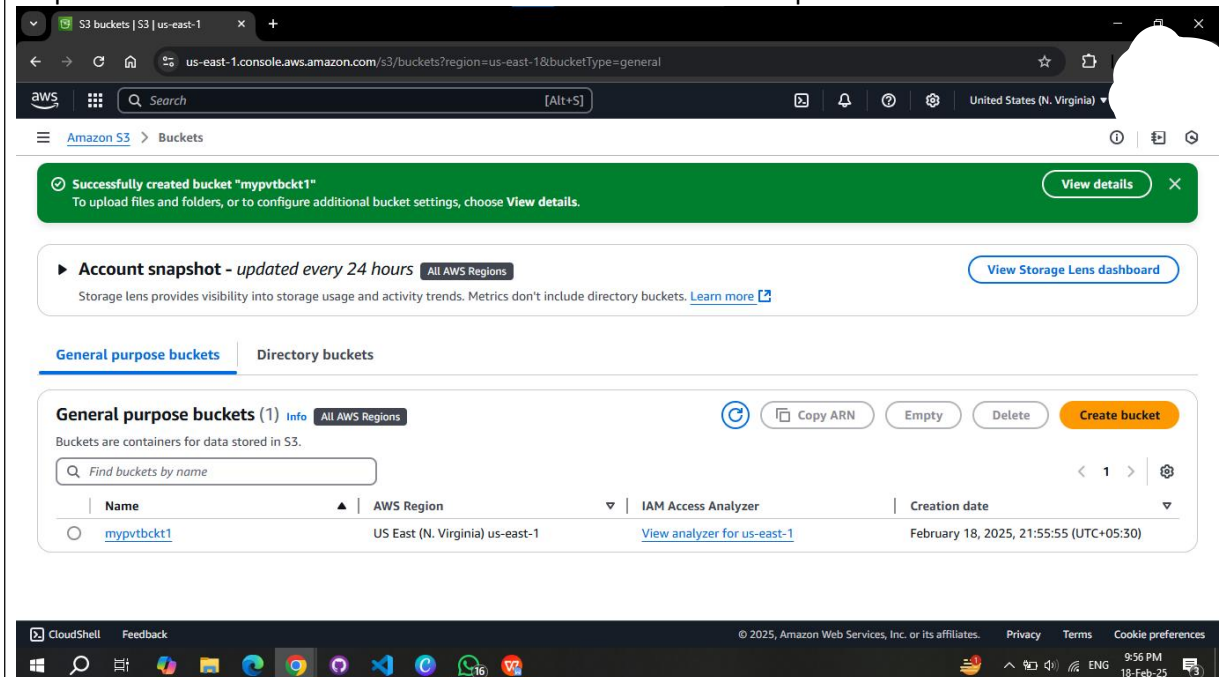


Step 2: In the Create bucket section, firstly name the bucket and keep 'Block all public access' box checked to keep the bucket private. Leave all other options as default and click on Create Bucket at the end of the screen.





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



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

The screenshot shows the AWS S3 console's 'Upload' page for a bucket named 'mypvtbckt1'. The page has a breadcrumb trail: 'Amazon S3 > Buckets > mypvtbckt1 > Upload'. Below the breadcrumb, there's an 'Upload' section with an 'Info' icon. It contains a message: '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](#)'. Below this is a dashed box with the text: 'Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.' Underneath is a 'Files and folders (0)' section. It has a 'Remove' button, 'Add files' and 'Add folder' buttons, and a search bar labeled 'Find by name'. Below the search bar is a table with columns: 'Name', 'Folder', 'Type', and 'Size'. The table is empty, with a message: 'No files or folders. You have not chosen any files or folders to upload.' Below the table is a 'Destination' section with an 'Info' icon. It shows the destination as 's3://mypvtbckt1'. The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray showing the time as 9:59 PM on 18-Feb-25.

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

This screenshot shows the same AWS S3 console 'Upload' page, but now a file has been added. The 'Files and folders' section now shows '(1 total, 94.4 KB)'. The search bar still says 'Find by name'. The table now has one row with the following data: 'ACFrOgBlyfhT\_uOYEDWe8-EoqCVLBWY...' in the 'Name' column, '-' in the 'Folder' column, 'application/pdf' in the 'Type' column, and '94.4 KB' in the 'Size' column. The 'Destination' section remains the same, showing 's3://mypvtbckt1'. Below the destination, there is a 'Destination details' section which is currently collapsed. The bottom of the screenshot shows the same Windows taskbar, but the system tray now shows the time as 10:00 PM on 18-Feb-25.

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

The screenshot shows the AWS console interface for an S3 bucket named 'mypvtbckt1'. A green banner at the top indicates 'Upload succeeded' with a close button. Below this, a 'Summary' section shows the destination as 's3://mypvtbckt1', with 1 file (94.4 KB) succeeded and 0 files failed. The 'Files and folders' tab is selected, showing a table with one file: 'ACFrOgBlyfT\_uOYEDWe8-...' of type 'application/pdf' and size '94.4 KB', with a status of 'Succeeded'. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 10:00 PM on 18-Feb-25.

Step 7: Select the uploaded file and in the Actions bar, select 'Share with a presigned URL'

The screenshot shows the AWS console interface for the 'mypvtbckt1' bucket. The 'Objects' tab is selected, displaying a list of objects. One object, 'ACFrOgBlyfT\_uOYEDWe8-...', is selected. The 'Actions' button in the top bar is clicked, opening a dropdown menu. The menu options include 'Download as', 'Share with a presigned URL', 'Calculate total size', 'Copy', 'Move', 'Initiate restore', 'Query with S3 Select', 'Edit actions', 'Rename object', 'Edit storage class', and 'Edit server-side encryption'. The 'Share with a presigned URL' option is highlighted. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 10:00 PM on 18-Feb-25.

Step 8: Set the Time limit upto which we want our file to be accessible and click on 'Create presigned URL'. And the copy it.

Step 9: Paste the copied URL in a new window or browser to check if the file is accessible or not within the time limit.

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		



Step 10: After the set time limit, the file is inaccessible, so we've successfully created a private bucket.

