

**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

**LABORATORY MANUAL**

**III Semester**

**Batch:2024-26**

**Name: Rohit Kulkarni**

**USN: 1MS24MC087**

**Course: Cloud Computing**

**Course code: 24MCASS3**

**Course Credits: 0 : 1: 2**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**Accredited by National Board of Accreditation & NAAC with 'A+' Grade, MSR Nagar, MSRIT Post, Bangalore-560054**

[**www.msrit.edu**](http://www.msrit.edu/)

**Table of Lab Programs**

|  |  |
| --- | --- |
| **Sl. No.** | **Programs/Exercise/Topic.** |
| 1**.** | AWS Account Setup and Configuration. AWS Console Overview. Enable MFA. Create AWS budget alert. |
| 2. | AWS Identity Access Management (IAM) User and Group creation. Enable AWS IAM MFA. Create an AWS Account Alias (for Alternate Sign-in URL) |
| 3. | Amazon S3 – Introduction, Bucket Creation and upload objects (files). |
| 4. | Amazon S3 – Static Website Hosting (Multi-Page website), Versioning, Cross-Region Replication rule. |
| 5. |  |
| 6. |  |
| 7. |  |
| 8. |  |
| 9. |  |
| 10. |  |
| 11. |  |
| 12. |  |
| 13. |  |
| 14. |  |
| 15. |  |
| 16. |  |
| 17. |  |
| 18. |  |
| 19. |  |
| 20. |  |
| 21. |  |
| 22. |  |
| 23. |  |
| 24. |  |
| 25. |  |
| 26. |  |
| 27. |  |

**Date: 8-10-2025**

**Exercise:** AWS Account Setup and Configuration. AWS Console Overview. Enable MFA. Create AWS budget alert.

AWS Console overview / AWS Home page/ AWS Dash board

Widgets – default view/ Add or remove widgets - small panels on the dashboard showing metrics or shortcuts; users can add or remove them as needed.

Region – Specifies the geographical data center location where your AWS resources are deployed.

Services – A categorized list of all AWS offerings such as Compute, Storage, Database, etc.

Search bar – A quick-access bar to search and pin frequently used services for faster access.

pin the most used services to console by clicking on star next to the service name.

Enable MFA

**Notes**

* Make sure your phone is unlocked, Bluetooth is on, and it uses a screen lock (fingerprint/PIN).

**Option 1: Add a Passkey for Easier Login**

**Step1: Go to Security Credentials**

* **Sign in to AWS console.**
* Go to **your username → Security credentials**.
* Under **Multi-factor authentication (MFA)** click **“Assign MFA device.”**
* Choose **“Passkeys and security keys”** → **Next**.
* On the next screen choose **“Phone or tablet”**.
* AWS will show a **browser pop-up** asking to use a device.
  1. Select **your phone** (or “Use another device” if it prompts).
* Look at your phone — you should get a **“Use passkey”** or **biometric prompt**.
* Approve using **fingerprint or phone PIN**.
* Back in AWS, click **Finish**. The passkey is now your MFA method.

Next time you sign in, just choose **“Sign in with a passkey” → approve on phone**.

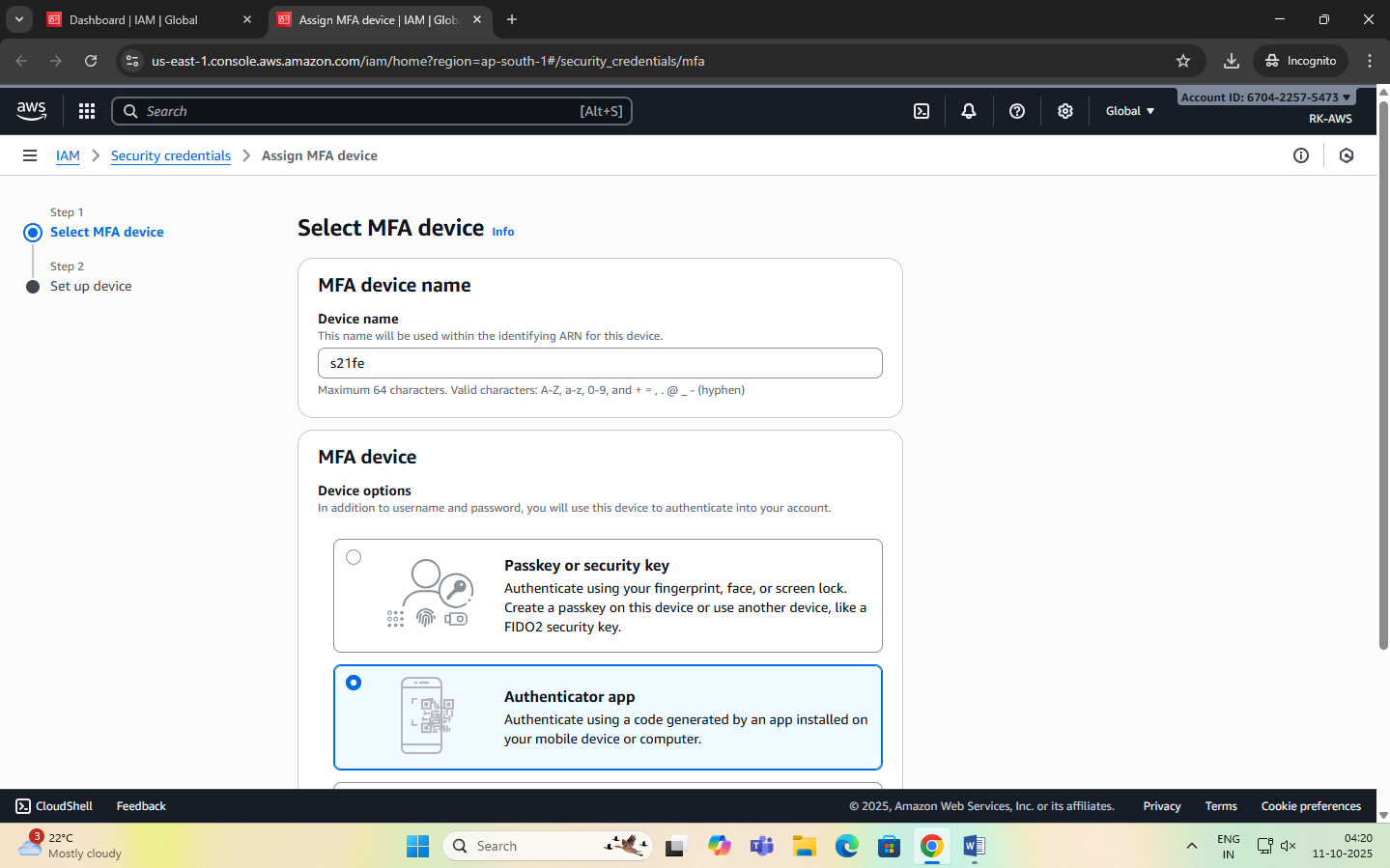
**Step 2: Test the Login**

* Sign out of AWS.
* Go to the login page.
* Choose “Sign in with a passkey” → Select your phone.
* Approve the prompt on your phone — you should be signed in without any MFA codes.

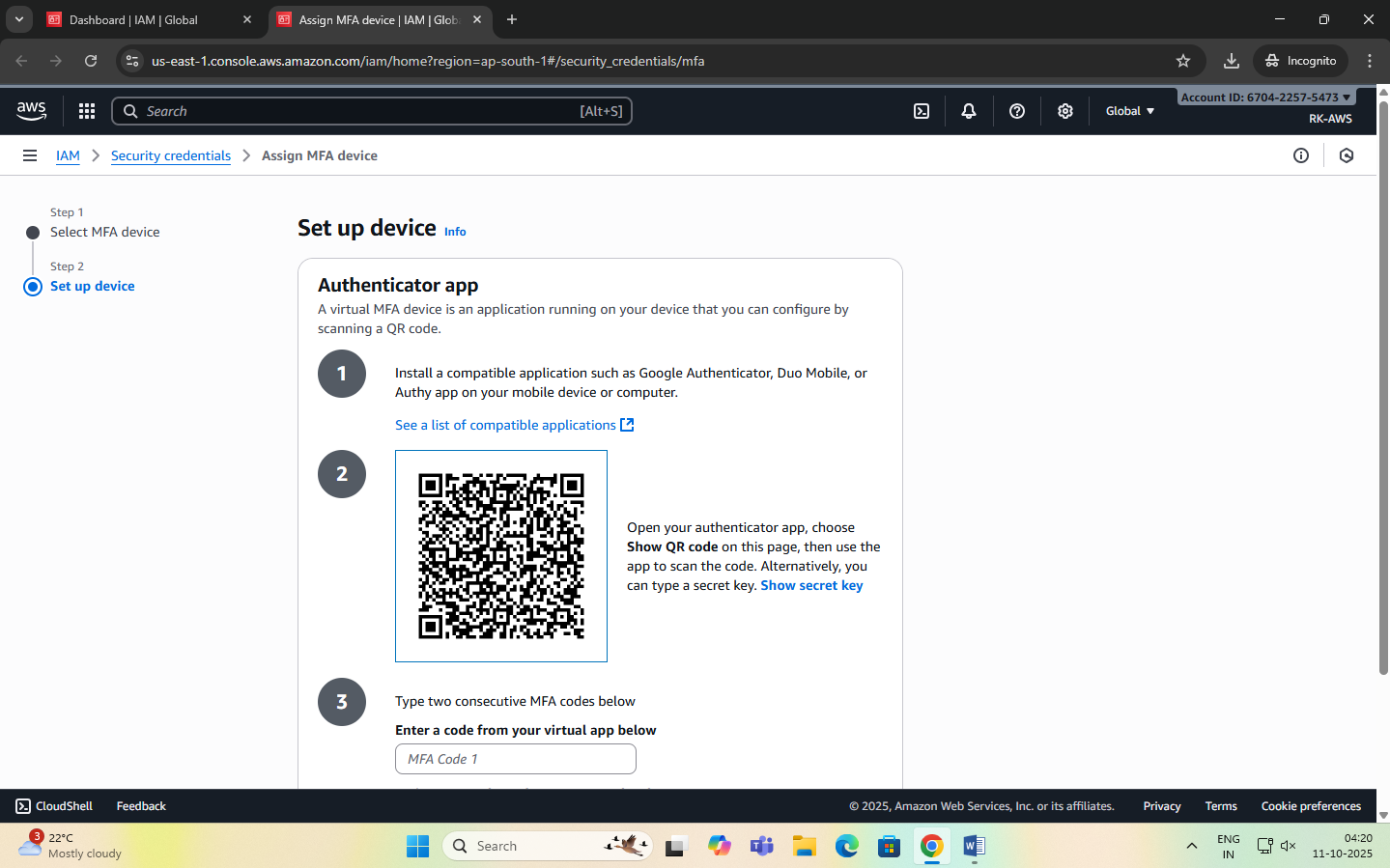
**Option 2: Authenticator App**

This uses a 6-digit code from Google Authenticator / Authy / Microsoft Authenticator.

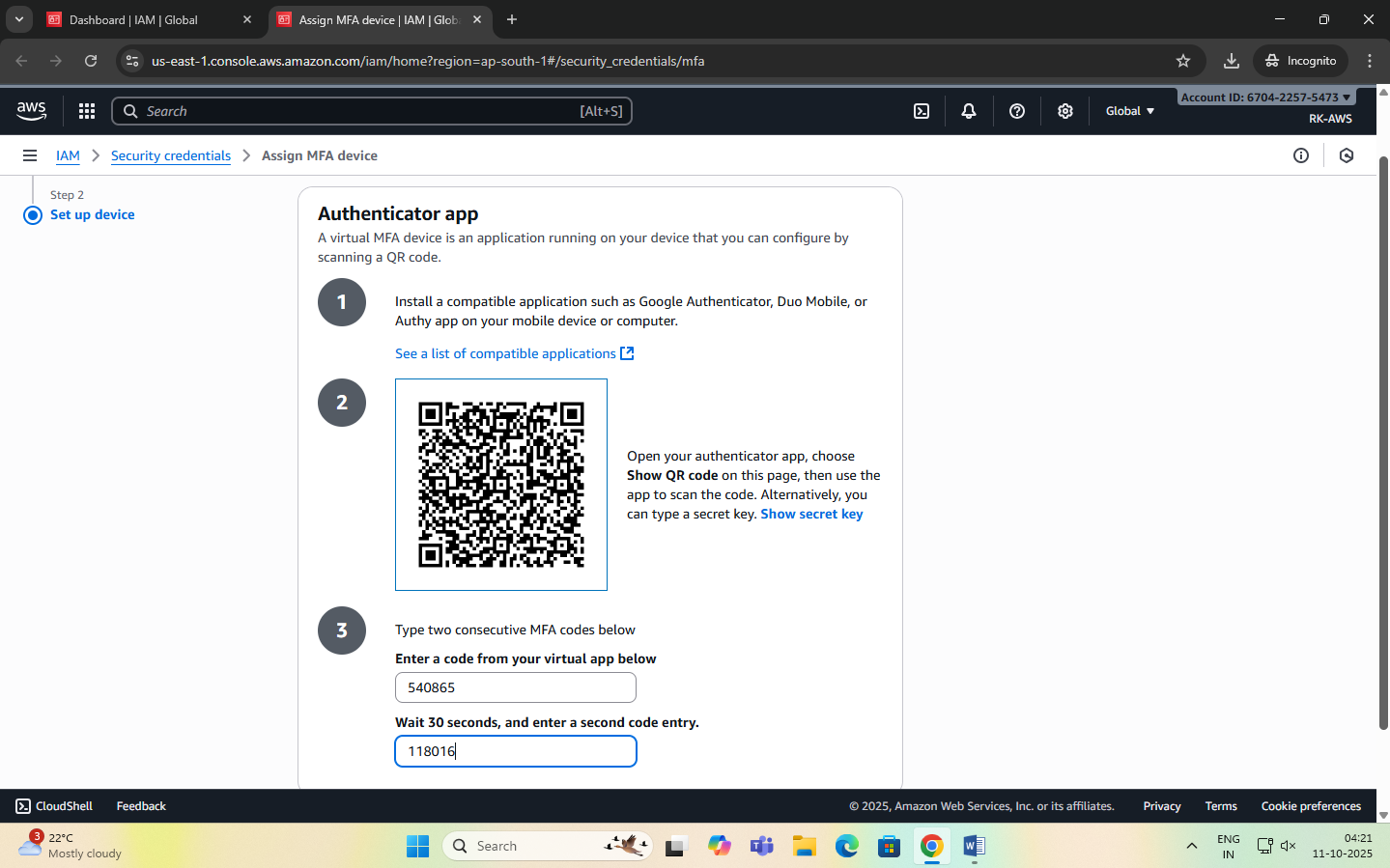
1. In **Security credentials**, click **“Assign MFA device.”**



1. Select **“Authenticator app”** → **Next**.
2. A QR code appears.



1. Open your authenticator app on your phone → **Add account → Scan QR code**.
2. The app shows a 6-digit code.
3. Enter that code back in AWS → **Assign MFA**. You’ll use the 6-digit code from the app each time you log in.



Create AWS budget alert

Allows to create a simple budget and to send alarms to registered email.

Example: if you are close to or exceeding your designated budget.

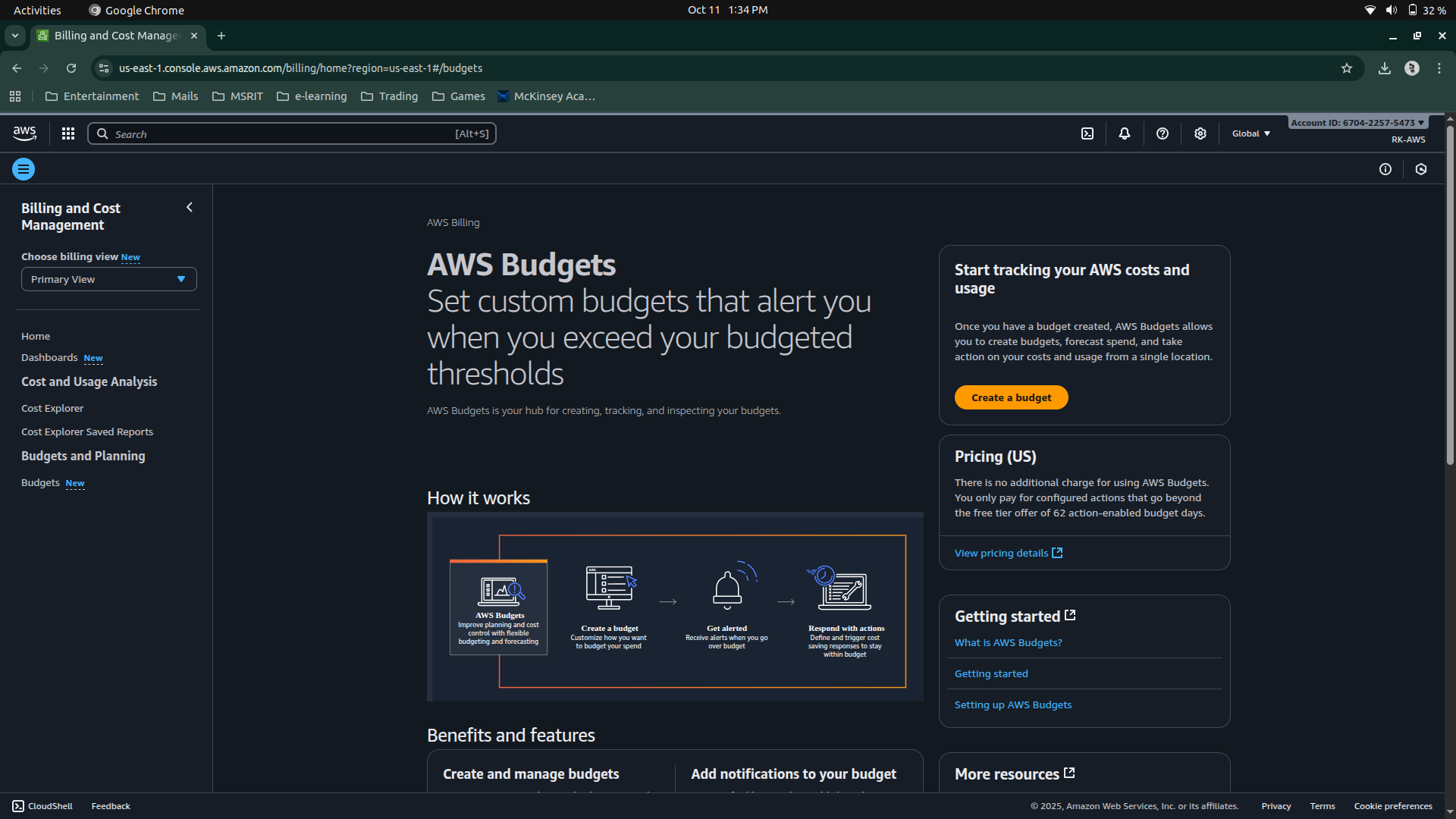
By setting a budget you can monitor budget threshold from the start.

Creating a budget

Step 1:

In the search bar type budgets and under the search results:

Select ‘Budgets’ from the Features group, which is essentially a feature of Billing and Cost Management service.



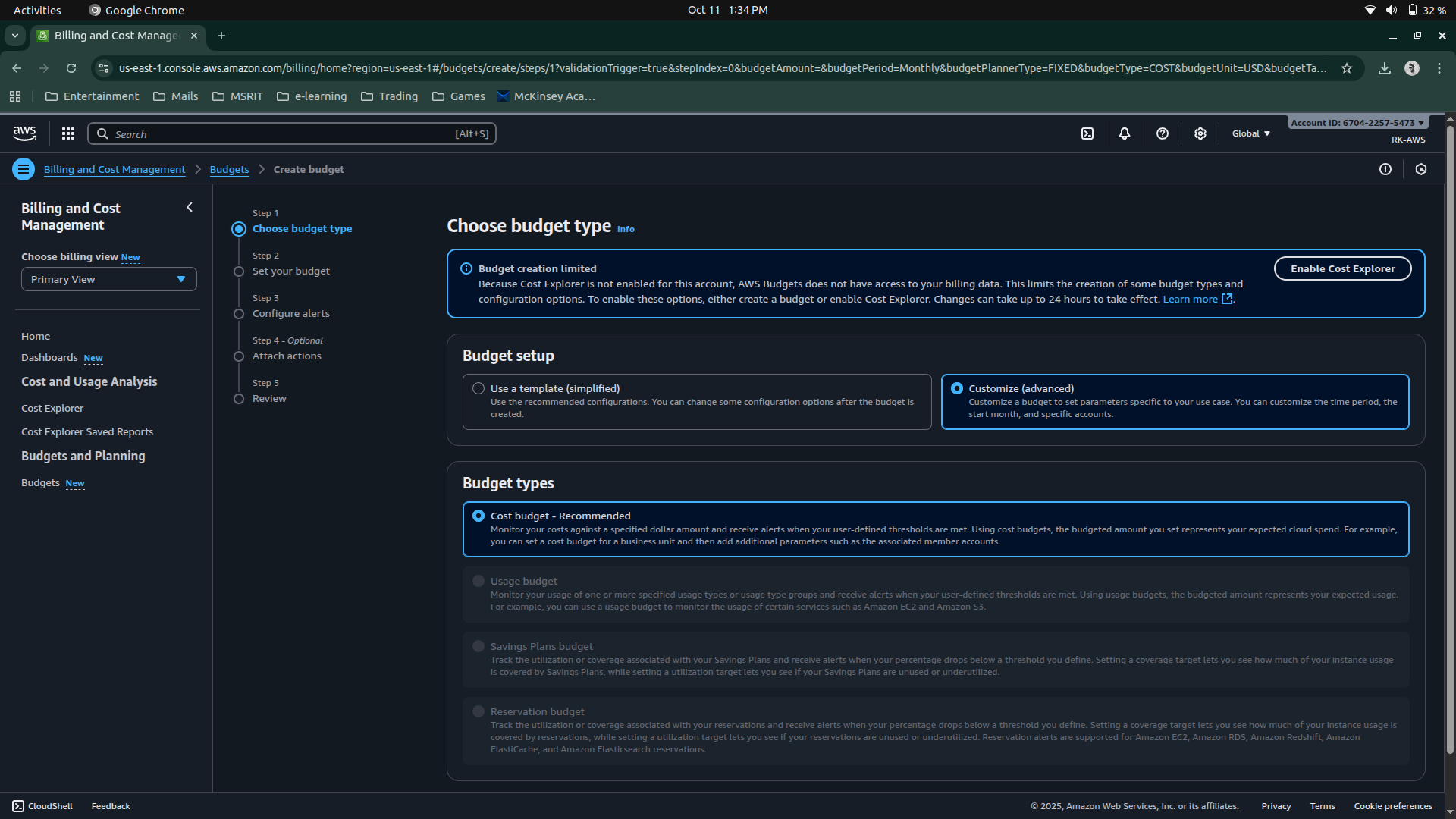
Step 2: After the ‘Budgets’ page loads

Click on Create Budget button

Under Budget setup select ‘Customize’ option

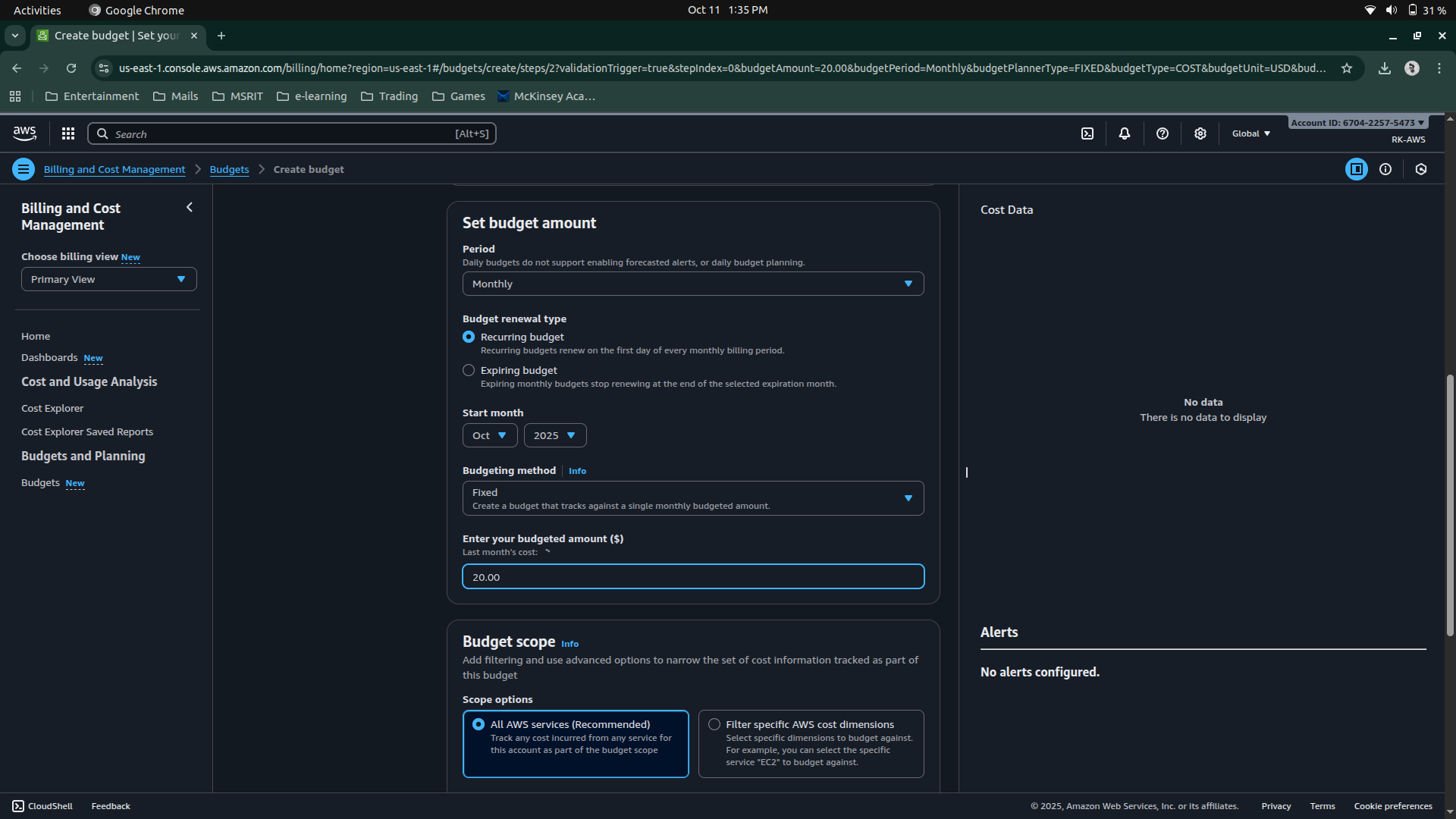
Under Budget types select ‘Cost budget’ option

Click on Next button.



Step 3: Set your budget page loads and fill in the details: MyBudget, Monthly, Recurring budget, set Month and Year, select Fixed – Budgeting method - Enter your budgeted amount – 20.00, select ‘All AWS services’ Scope options. Advanced options – leave it on default.

Click on Next button.



Step 4: Configure alerts

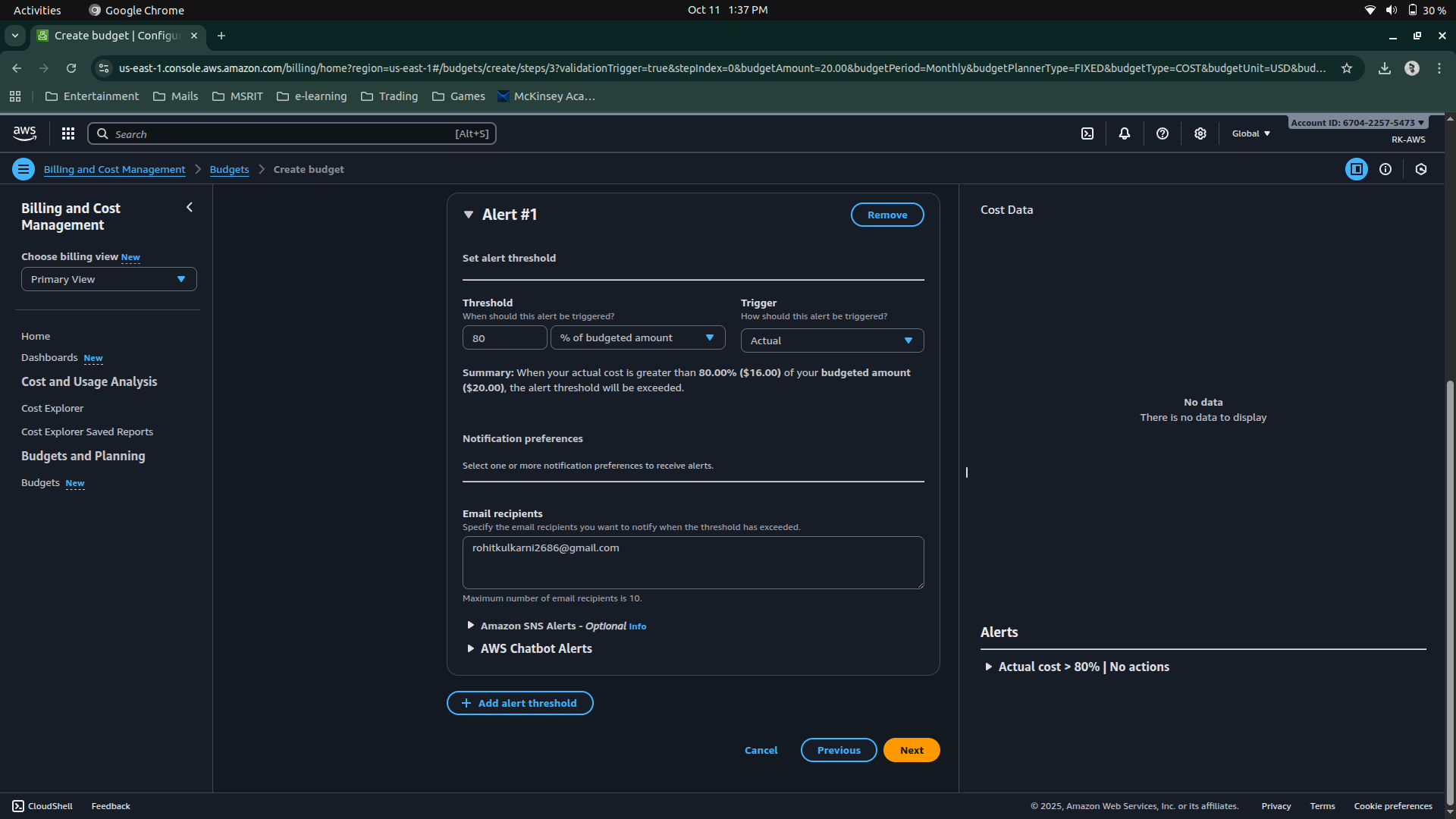
Click on Add an alert threshold

Under Set alert threshold

Set Threshold: 80 and Trigger: Actual

Email recipients: enter your email id.

Click on Next button.



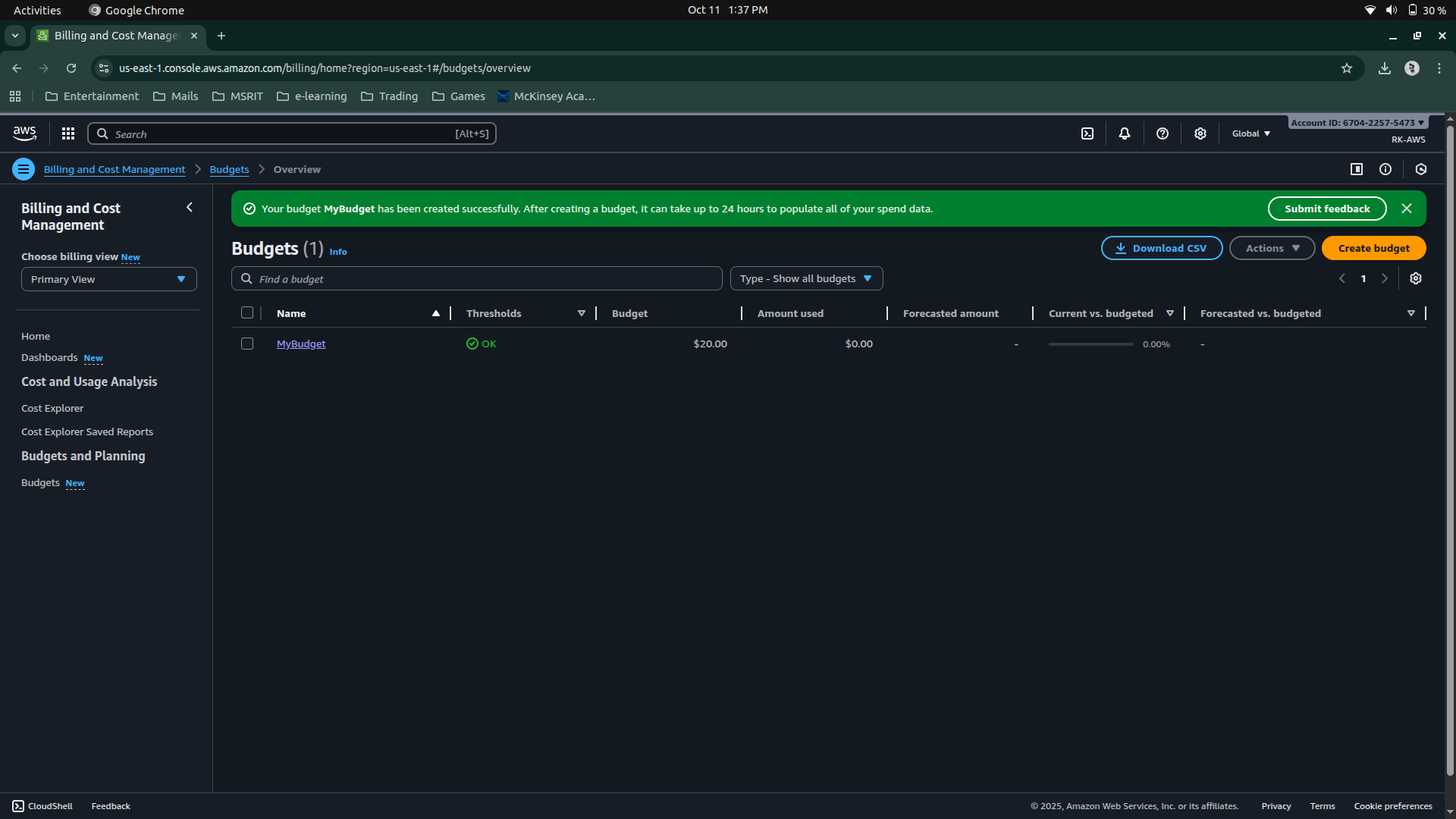
Step 5: Under Attach actions – leave it on default

Click on Next button.

Step 6: In review page – check and review all the options are set to what is desired.

Click on Create budget button.

Now, the budget has been created.



**Date: 9-10-2025**

**Exercise 2:** AWS Identity Access Management (IAM) User and Group creation. Enable AWS IAM MFA. Create an AWS Account Alias (for Alternate Sign-in URL)

AWS Identity and Access Management (IAM) is a security service that helps you control who can access your AWS resources and what actions they can perform. It is a global AWS service.  
It allows you to securely manage users, groups, roles, and permissions in your AWS account.

|  |  |
| --- | --- |
| Concept | Description |
| Root User | The account owner who created the AWS account. Has full access and should be used only for account setup. |
| Group | A collection of IAM users that share the same permissions. For example, a “Developers” group or “Students” group. |
| User | A person or application that interacts with AWS (e.g., student1, admin, developer). Each user has its own username, password, and access keys. |
| Policy | A JSON document that defines what actions are allowed or denied (e.g., “allow S3 read access”). |
| Role | A set of permissions that can be temporarily assumed by a user, service, or application — often used by EC2 instances or Lambda functions. |

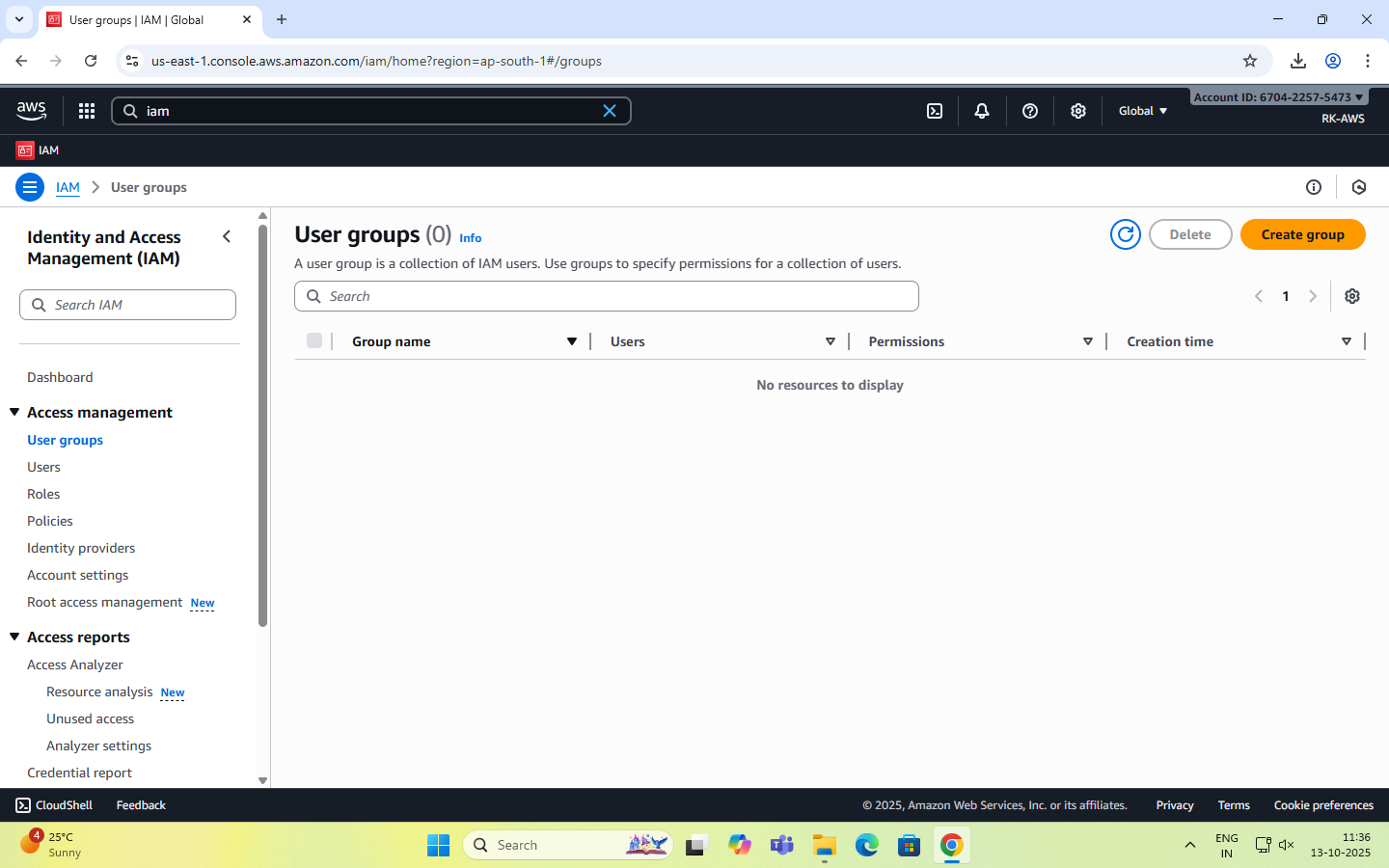
**STEPS for AWS IAM User Group Creation**

**Step 1: Sign in to AWS Console**

* Log in to your AWS Management Console using an administrator account.
* From the Services menu, search for IAM and open it.

**Step 2: Open Groups Section**

* In the IAM dashboard, look at the left sidebar.
* Click on “User groups” → then click “Create group”.

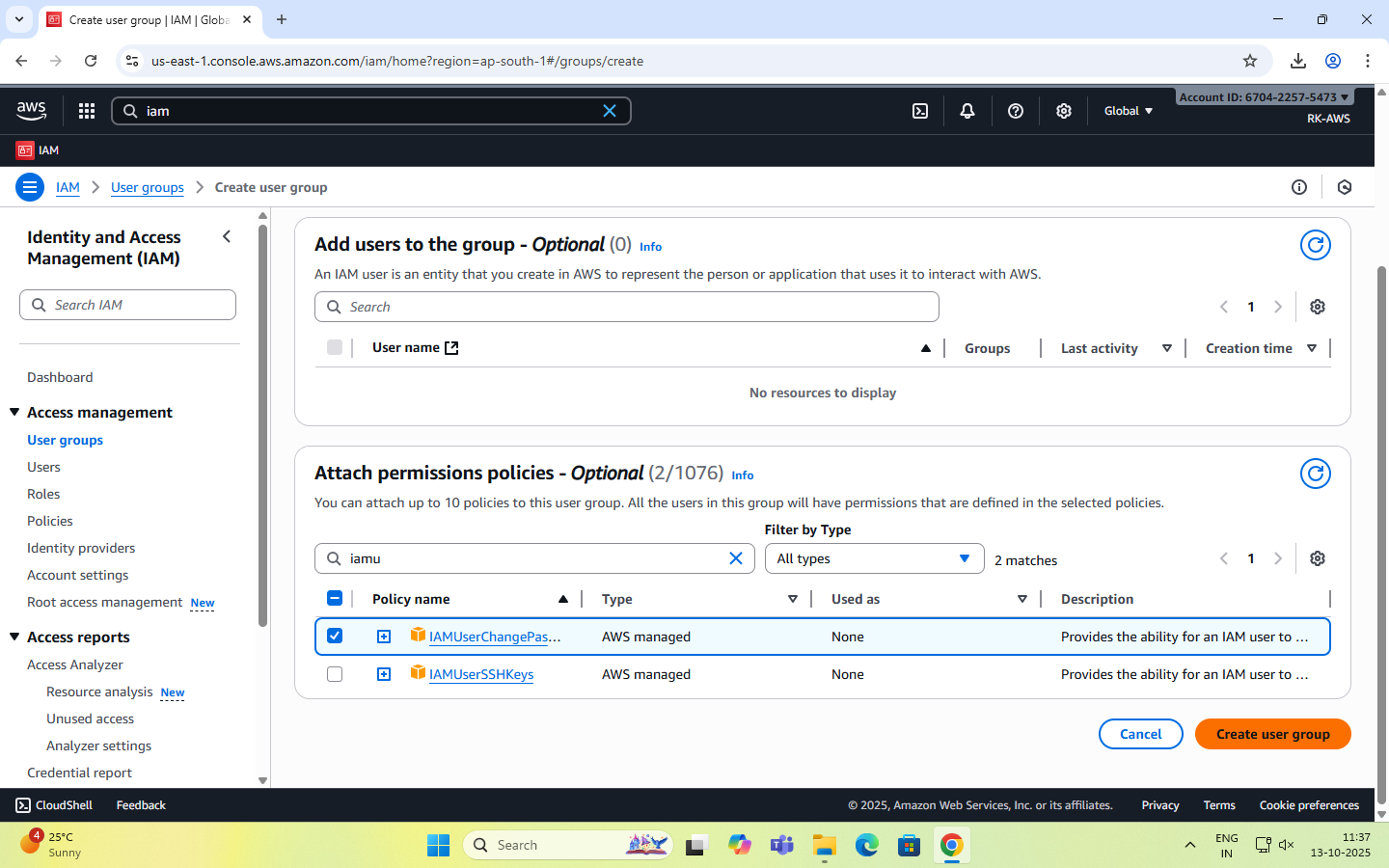


**Step 3: Name the Group**

* Enter a Group name (example: Developers, Admins, Students, etc.).
* Group names must be unique within your account.

**Step 4: Attach Permissions Policies**

* You can attach IAM policies to define what members of the group can do.  
  Select the following:
  + AdministratorAccess → Full access to all services.
  + IAMUserChangePassword

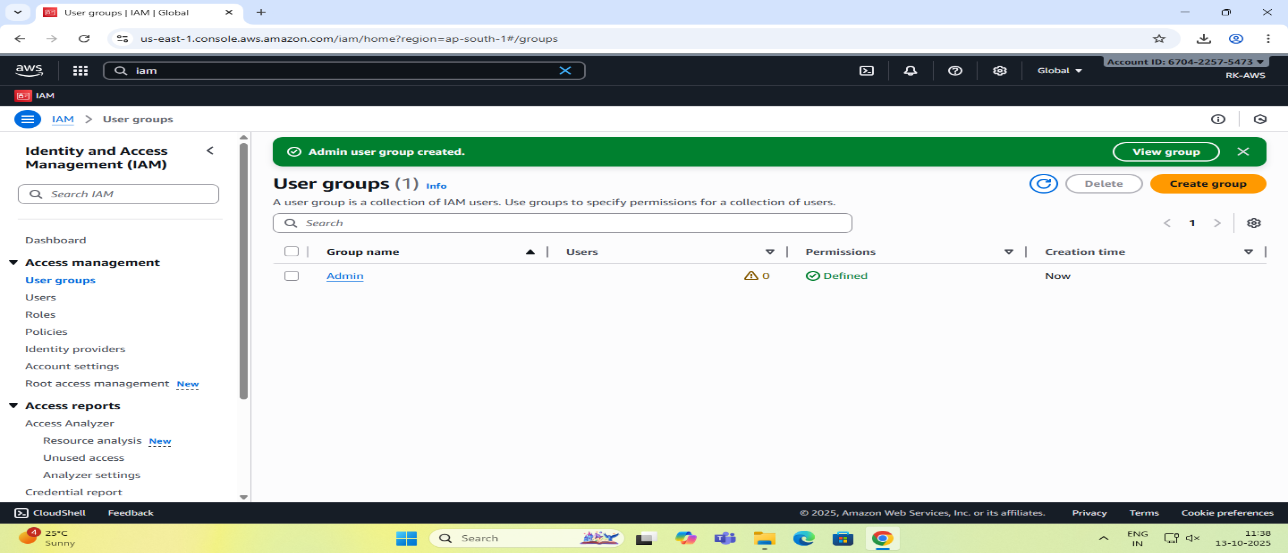


**Step 5: Review and Create**

* Review all details (group name + permissions).
* Click “Create group” to finalize.

**Step 6: Group Successfully Created**

* The new group now appears in the IAM dashboard.
* Any user added to this group automatically inherits all permissions attached to the group.



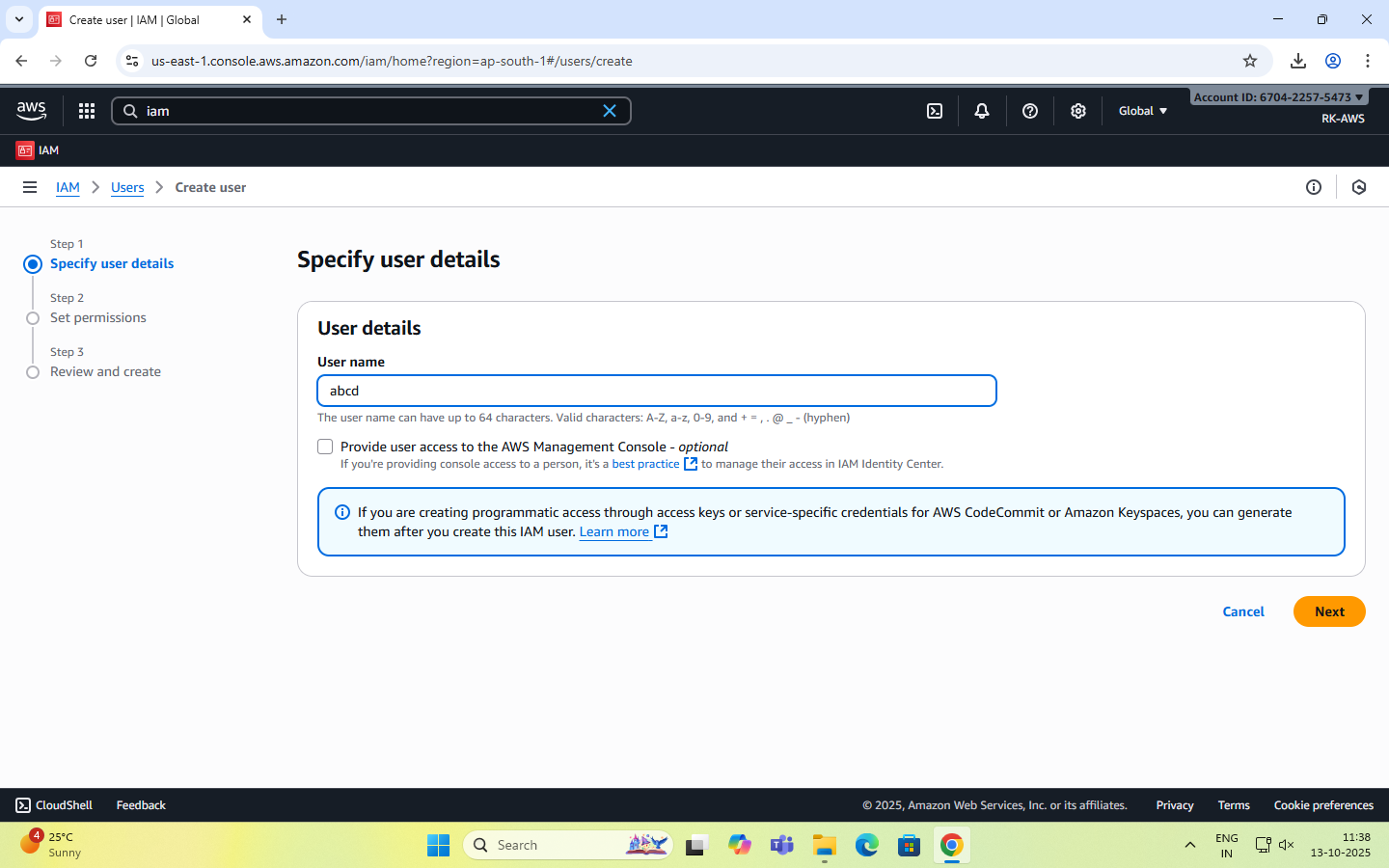
**STEPS for AWS IAM User Creation**

**Step 1: Sign in to AWS Management Console**

* Login to the AWS Management Console using your root user credentials.
* In the search bar, type IAM and open IAM service.

**Step 2: Navigate to Users Section**

* In the IAM dashboard’s left panel (info panel), click on Users.
* Then click on the “Add users” button.



**Step 3: Set User Details**

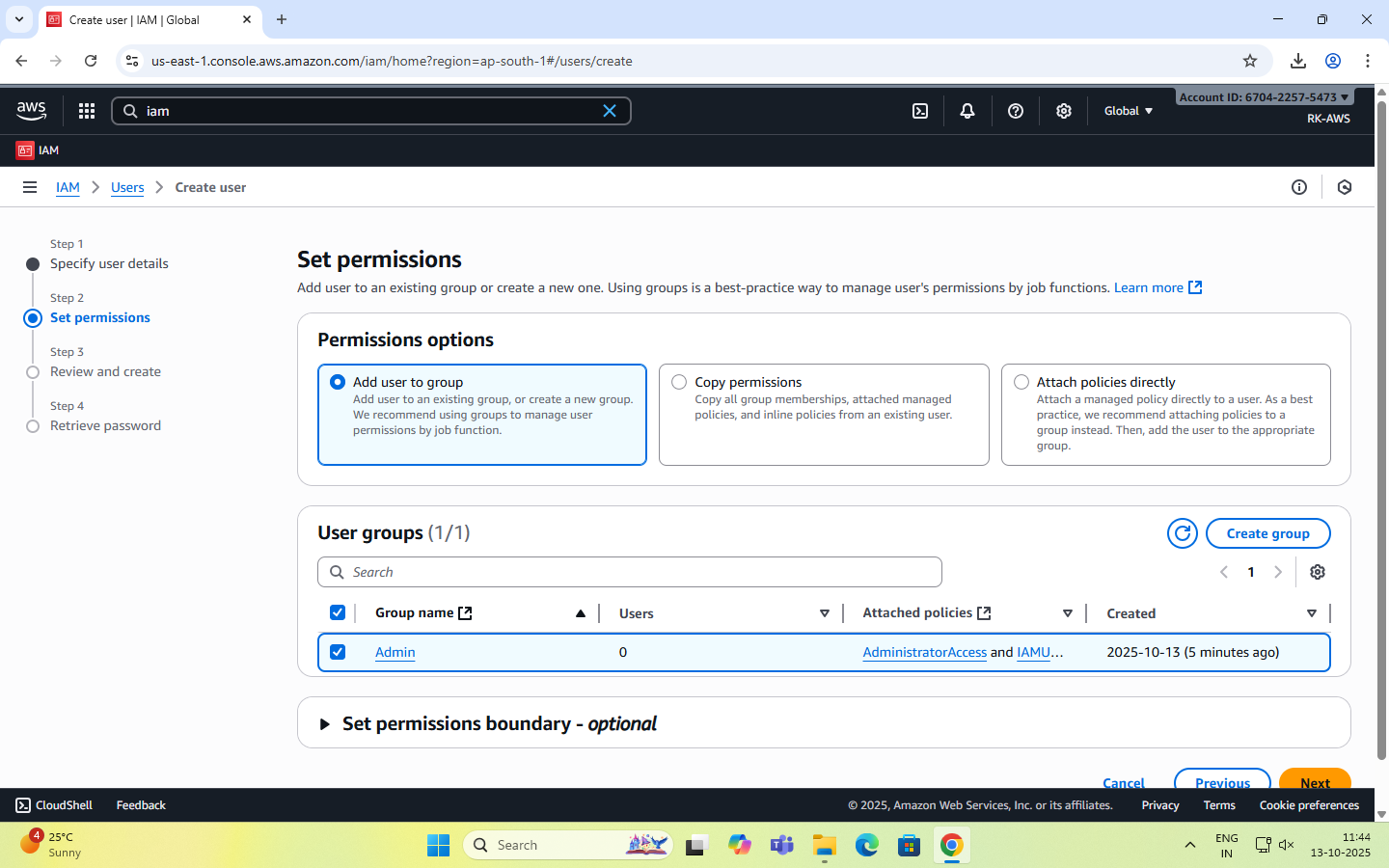
* Enter a user name.
* Checkbox – ‘Provide user access to the AWS Management Console’.
* Select ‘I want to create an IAM user’ option



**Step 4: Set Permissions**

You can grant permissions to the new user in three ways:

1. Add user to group – Assign predefined permission groups.
2. Copy permissions from an existing user.
3. Attach existing policies directly.



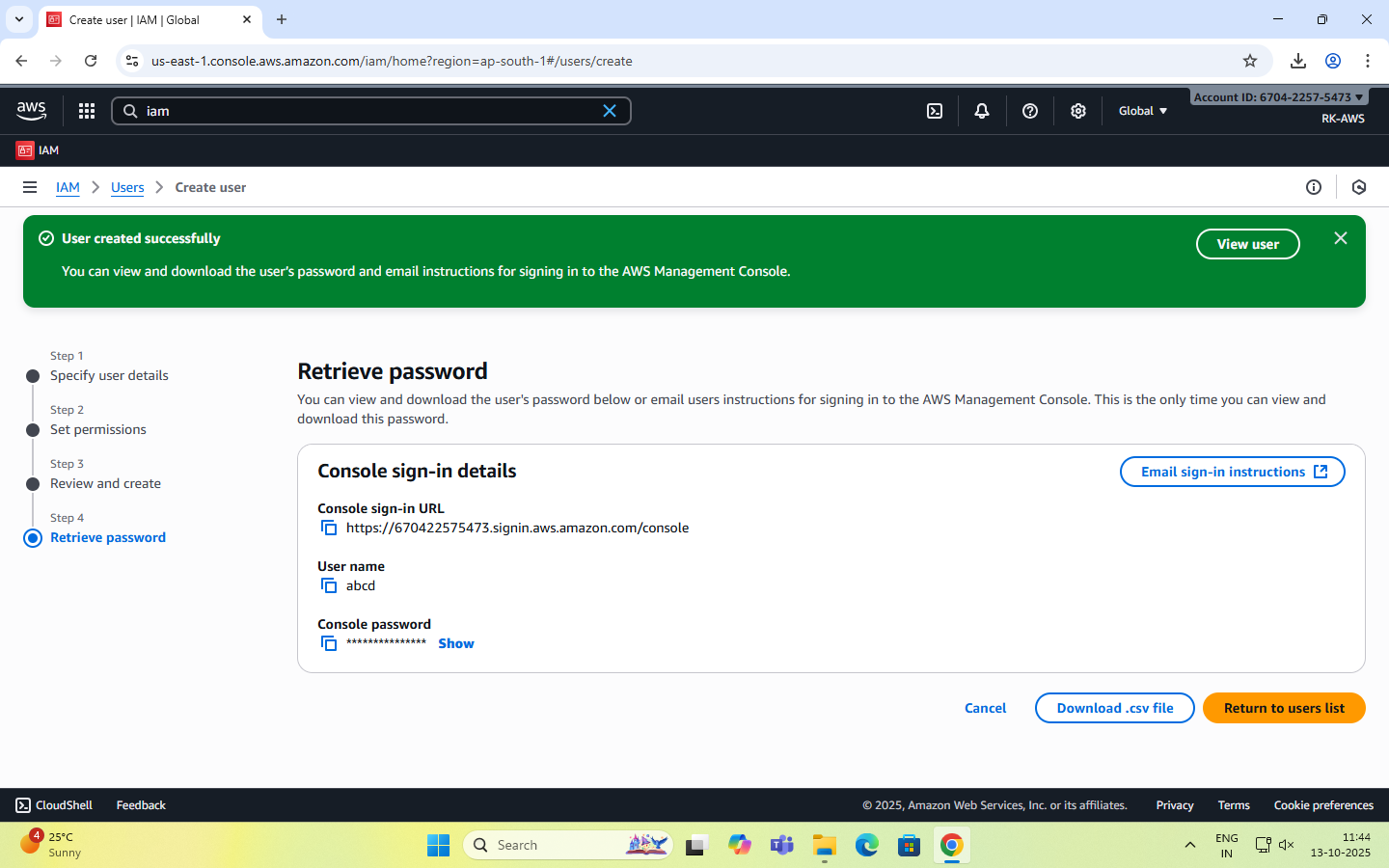
Recommended: Use IAM **groups** to manage permissions easily.

**Step 5: Set User Details and Tags (Optional)**

* Add tags like Department: MCA or Role: Faculty/Student for easy identification.
* Click Next to review.

**Step 6: Review and Create User**

* Review all settings (user name, permissions, tags).
* Click Create user.



**Step 7: Save Credentials**

* After creation, AWS displays:

Console sign-in URL - <https://rohit2001.signin.aws.amazon.com/console>

Username -

Password –

Account ID- 670422575473

**NOTE:** Download or copy these credentials immediately — they cannot be retrieved later.

**Step 8: Test the User Login**

* Visit the IAM login URL provided (unique for your AWS account).
* Log in with the newly created username and password.
* Verify access and permissions.

Enable AWS IAM MFA

**Step 1:** Sign in to AWS Console as root user

**Step 2: Open IAM Dashboard**

* In the search bar, type IAM and select IAM (Identity and Access Management).
* From the left navigation pane, select Users.

**Step 3: Select a User**

* Click the user name for whom you want to enable MFA.
* This opens the User Summary page.

**Step 4: Go to Security Credentials Tab**

* Click the Security credentials tab.
* Scroll down to the section “Multi-factor authentication (MFA)”.

**Step 5: Assign MFA Device**

* Click “Assign MFA device”.
* Choose the MFA type:
  1. Virtual MFA device (e.g., Google Authenticator, Authy — most common)
  2. Security key (hardware-based, e.g., YubiKey)
  3. Authenticator app on phone

**Step 6: Configure Virtual MFA**

* If you select Virtual MFA device:
  1. Open your Google Authenticator or Authy app on your phone.
  2. Scan the QR code shown on the AWS screen.
  3. The app starts generating 6-digit codes.

**Step 7: Verify MFA**

* Enter two consecutive codes from your app in the verification fields.
* Click “Assign MFA”.

**Step 8: Confirm Setup**

* You will see a green checkmark confirming MFA is successfully assigned.
* The user now requires MFA each time they sign in.

Create an AWS Account Alias (for Alternate Sign-in URL)

As an IAM user you can sign in using the default URL or create an account alias for it.An Account Alias gives your AWS account a name instead of using the long numeric Account ID in your sign-in URL.  
This makes it easier for IAM users to remember and log in.

**Step 1:**

* Sign in to the AWS Management Console using root user or an IAM user with administrative privileges.
* In the search bar, type IAM, and open the IAM service.

**Step 2:**

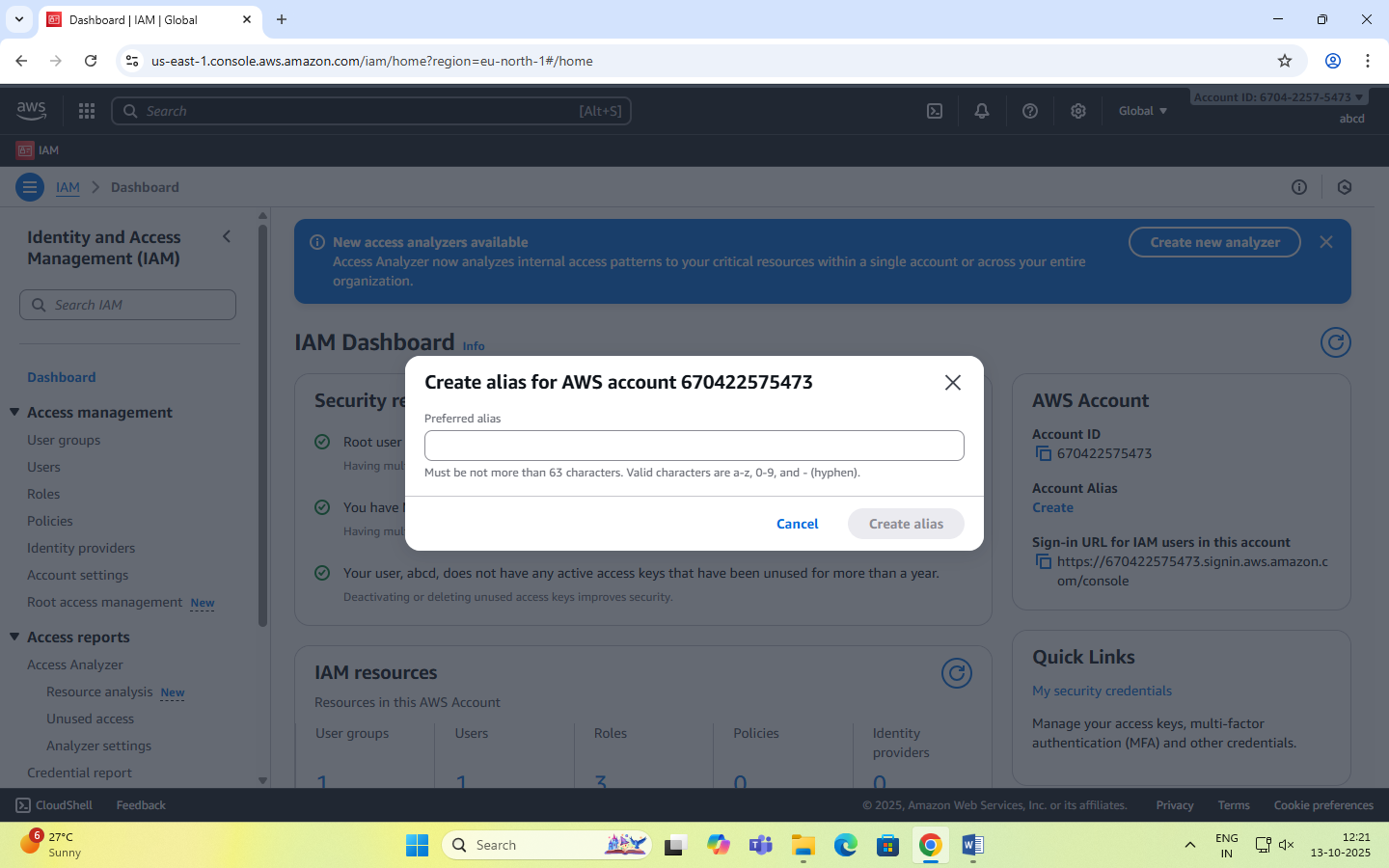
* In the left navigation pane, scroll down and select Dashboard.

**Step 3:**

* Under the “AWS Account” section, find “Account Alias”.
* Click on “Create” (or “Edit” if one already exists).

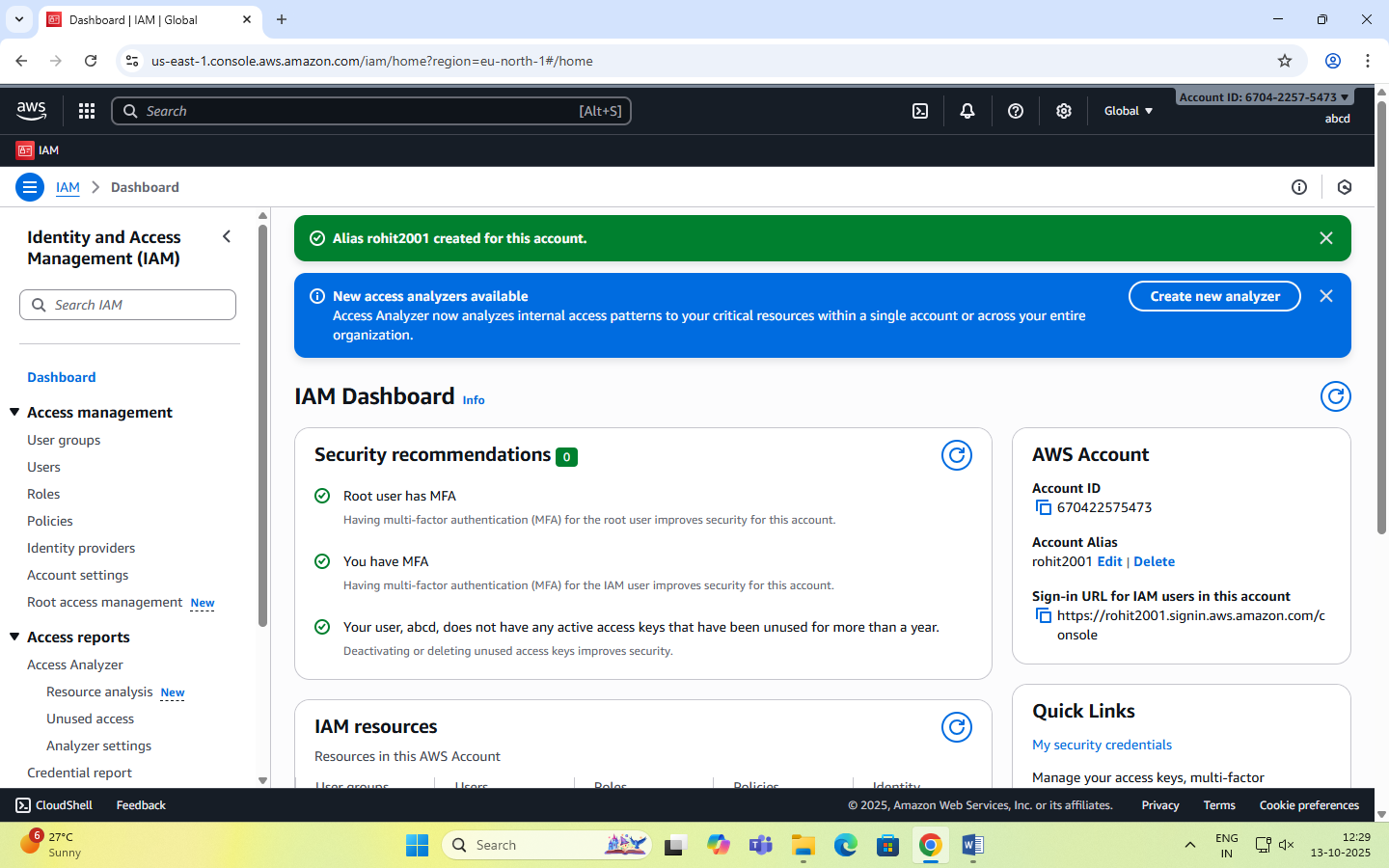
**Step 4:**

* In the pop-up box, enter your preferred alias name.
* Click “Create alias”.



**Step 5:**

* Once created, you’ll see a new Sign-in URL displayed.



**Date: 14-10-2025**

**Exercise-3:** Amazon S3 – Introduction, Bucket Creation and upload objects (files).

Amazon S3 (Simple Storage Service) is a fully managed, object-based storage service offered by AWS.  
It allows you to store and retrieve unlimited amounts of data from anywhere on the web.

Unlike traditional file systems that store data in folders, S3 stores data as objects inside buckets.  
Each object has:

* Data (the actual file),
* Metadata (information about the file),
* Unique key (its name within the bucket).

You can store unlimited data in Amazon S3.

However, each AWS account can create up to 100 buckets by default (can be increased by request).

Each object (file) can be up to 5 TB (terabytes) in size.

Single upload limit: 5 GB (may vary), but larger files can be uploaded using Multipart Upload.

Amazon S3 is designed for:

* Durability: 99.999999999% (11 nines) – this means your data is extremely safe.
* Availability: 99.99% uptime – your data is almost always accessible.
* S3 automatically stores multiple copies of your data across multiple Availability Zones in a region.

Storage classes: S3 offers different storage classes depending on cost and frequency of access.

|  |  |
| --- | --- |
| Storage Class | Description |
| S3 Standard | For frequently accessed data (default). |
| S3 Intelligent-Tiering | Automatically moves data between frequent - infrequent access tiers. |
| S3 Standard-IA | For infrequently accessed data but still quickly available. |
| S3 Glacier | For archival storage (low cost, slower retrieval). |
| S3 Glacier Deep Archive | For long-term archival (lowest cost). |

Each object can be accessed through a **unique URL**.  
S3 is commonly used for:

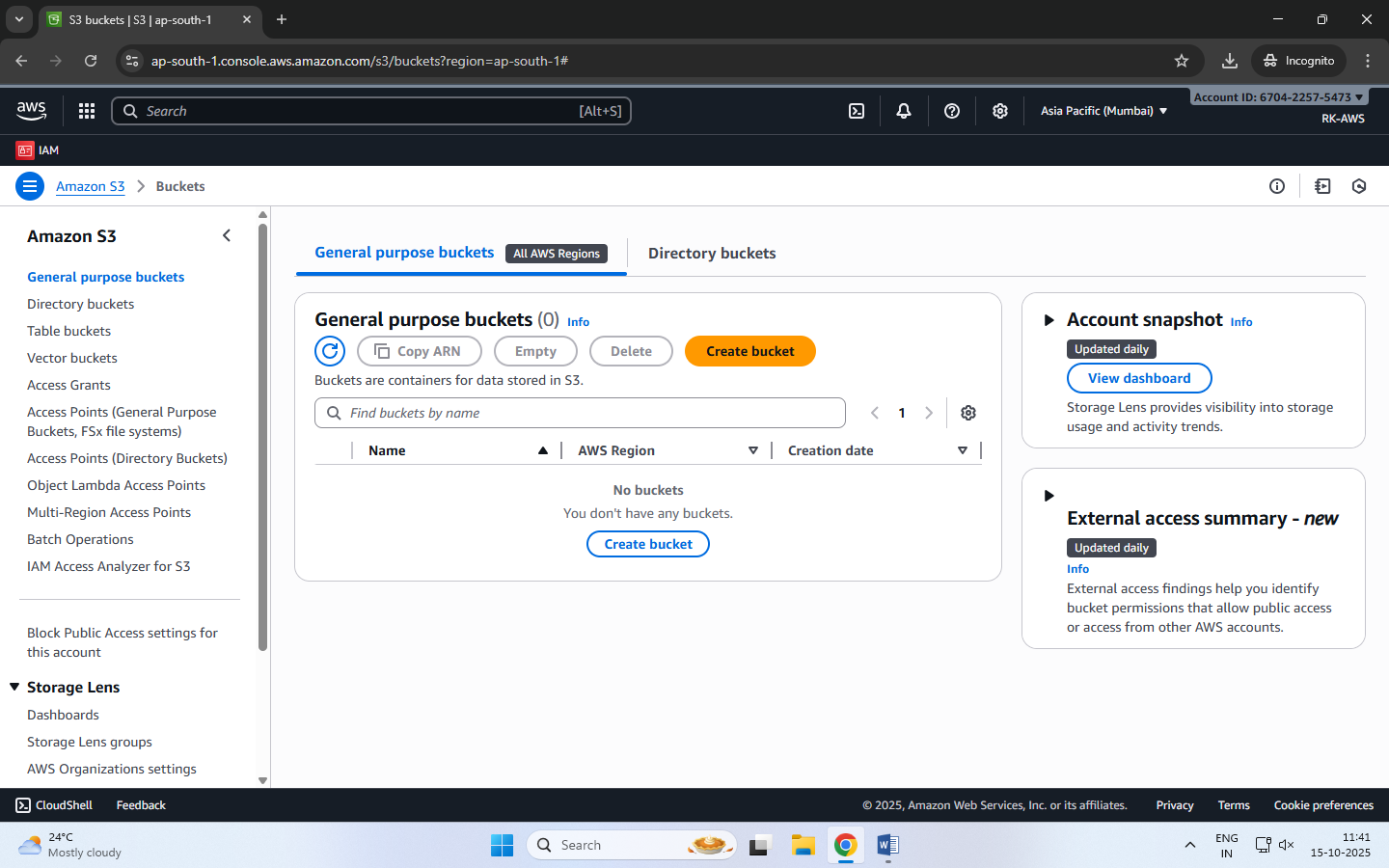
* Backup and archival
* Static website hosting
* Application data storage
* Media content delivery

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Buckets | Top-level containers for storing objects. |
| Objects | Actual files stored in S3 (e.g., images, HTML, PDFs). |
| Region | Each bucket is created in a specific AWS region. |
| Versioning | Maintains multiple versions of the same object for recovery. |
| Static Website Hosting | Allows you to host HTML pages directly from an S3 bucket. |

**Steps to Create an S3 Bucket and Upload an Image**

**Step 1: Open the S3 Service**

* Sign in to your AWS Management Console.
* In the search bar, type S3 and click Amazon S3.



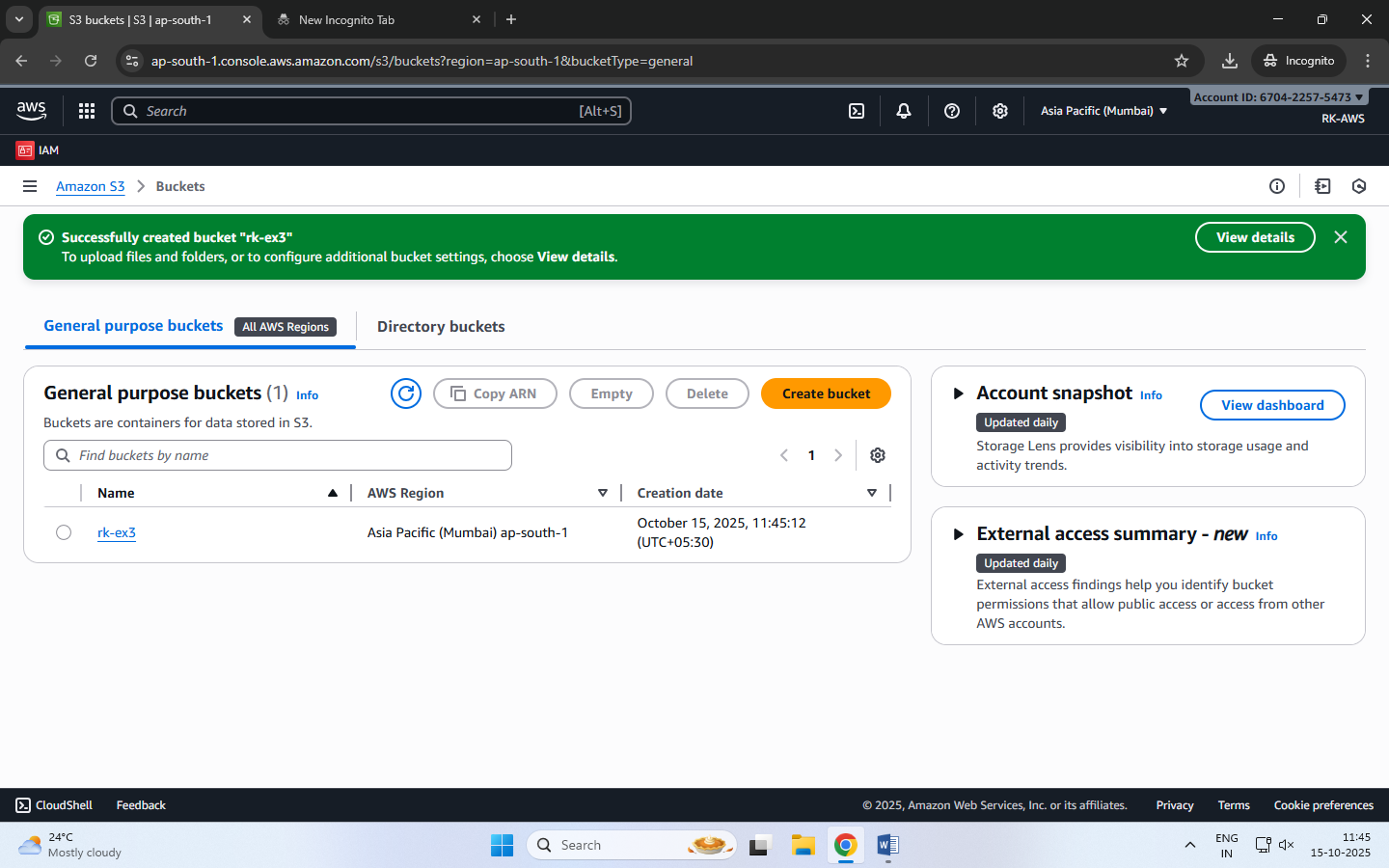
**Step 2: Create a New Bucket**

* Click “Create bucket”.
* Bucket type: General purpose
* Bucket name: Enter a name (Bucket names must be globally unique and lowercase.)
* AWS Region: Choose the region nearest to you.
* Scroll down and uncheck:

“Block all public access” → Uncheck this (for viewing file publicly).

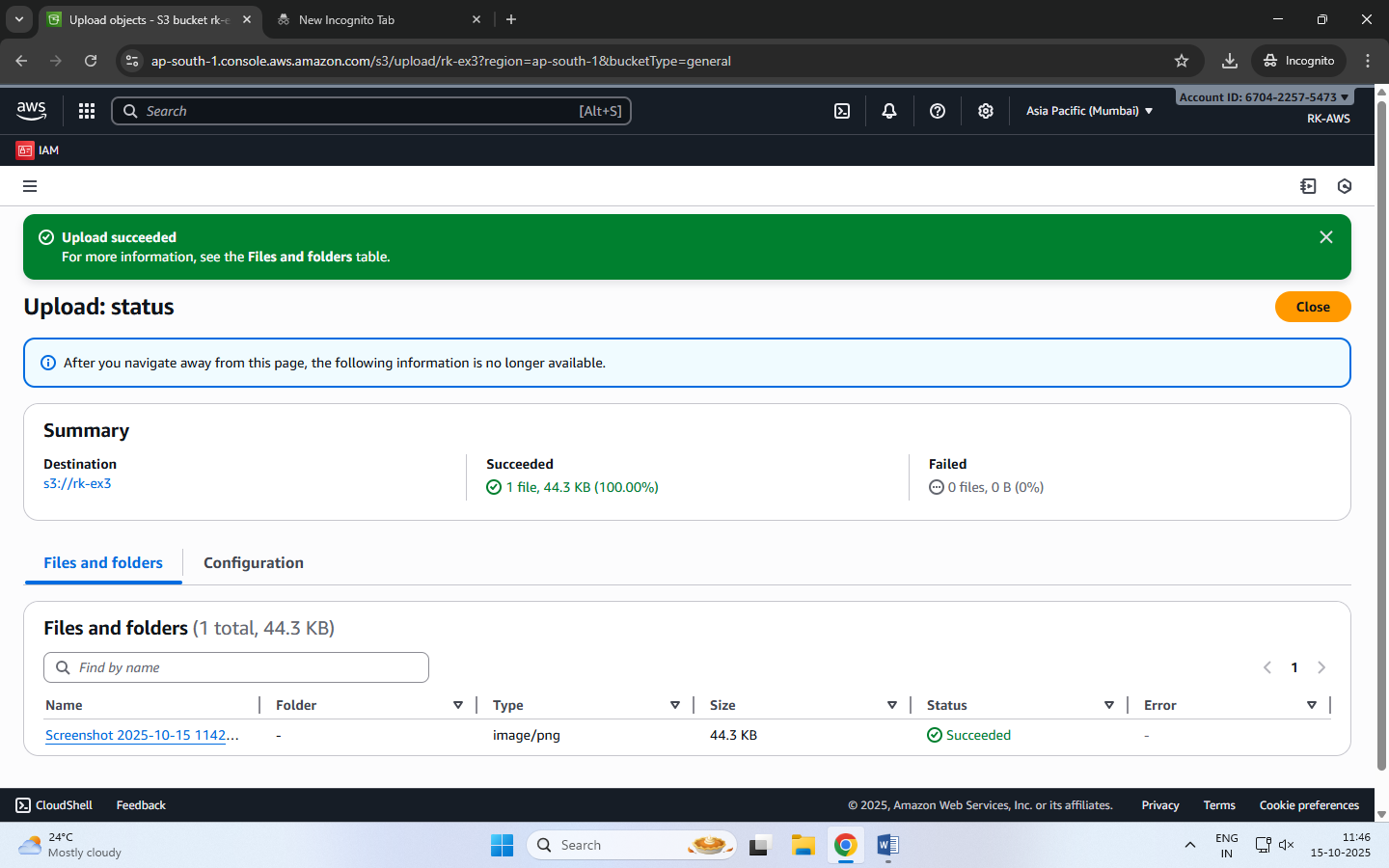
Confirm the warning checkbox.

* Keep all other settings as default.
* Click Create bucket.



**Step 3: Upload an Image File**

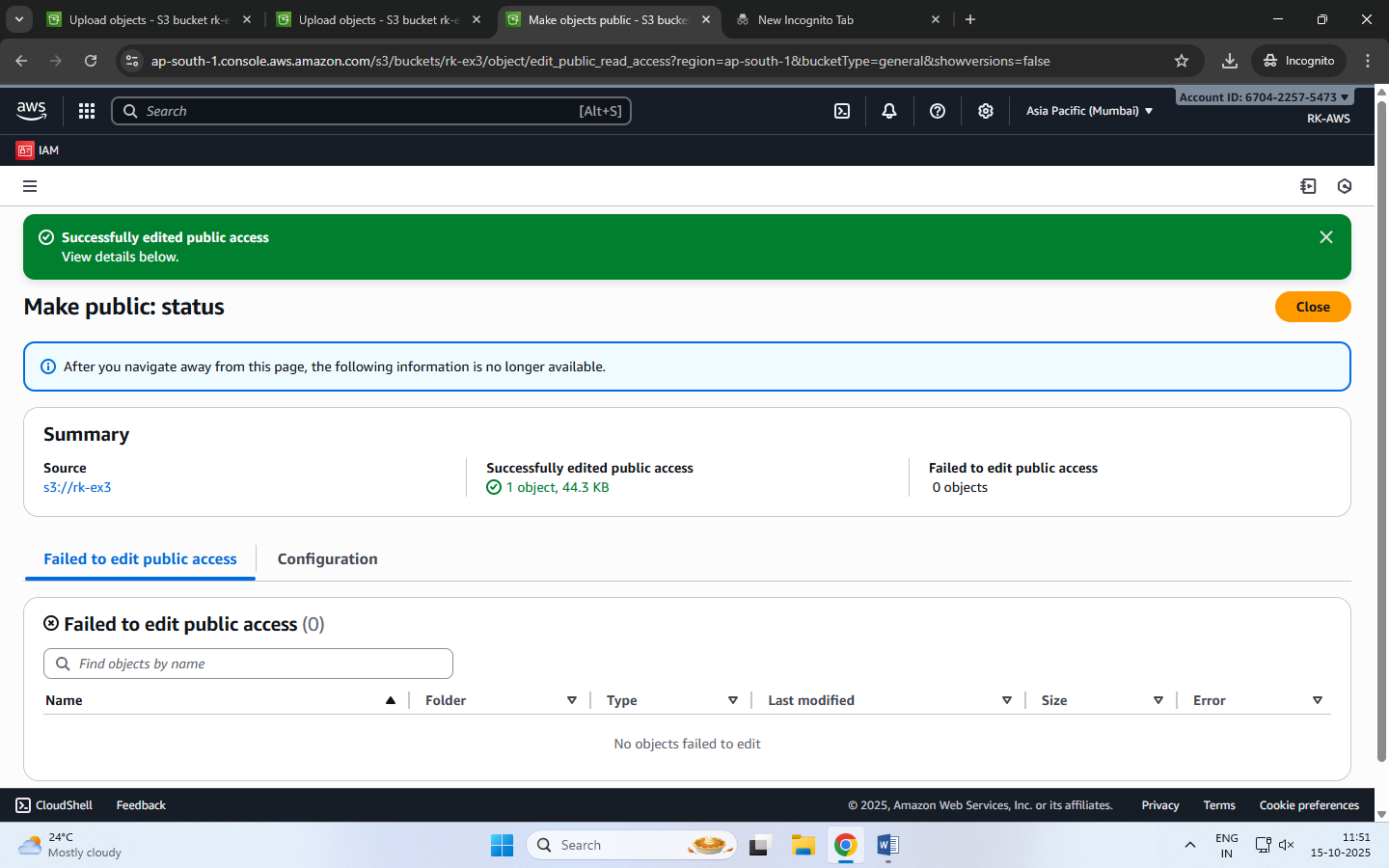
1. Click on the newly created bucket name.
2. Click Upload → Add files.
3. Choose any image file from your computer.
4. Scroll down and click Upload.



**Step 4: Make the Object Public**

By default, S3 objects are private. To view them in a browser, make the file public.

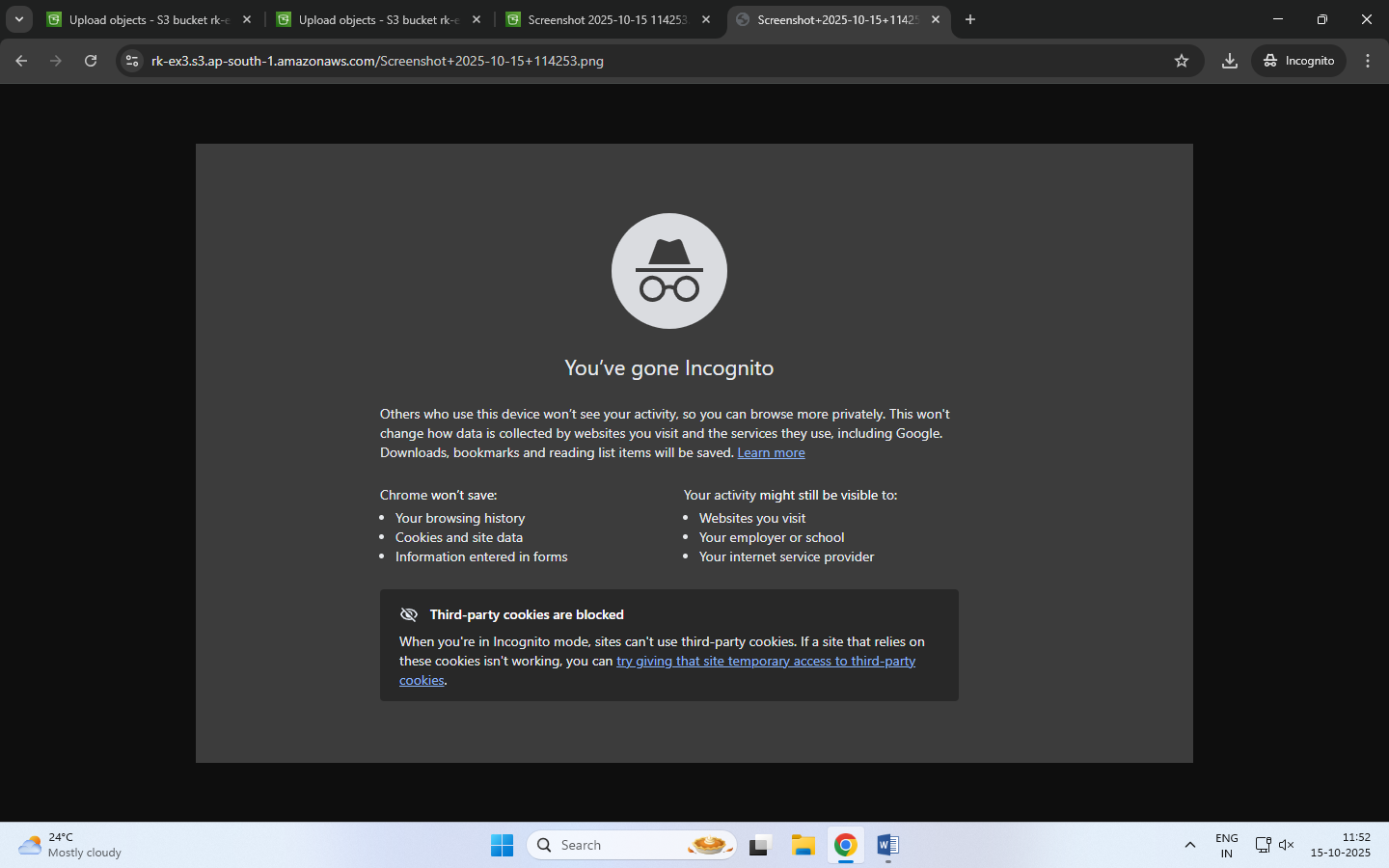
1. After upload, go to the Objects tab inside your bucket.
2. Select the uploaded file.
3. Click Actions → Make public using ACL (or Object actions → Make public, depending on console version).
4. Confirm.



**Step 5: Copy the Object URL**

1. Open the object again by clicking its name.
2. Scroll down to the **Object URL** section.
3. Copy this URL and open it in a new browser tab.

The image is displayed directly from the S3 bucket — meaning AWS S3 is serving that object over HTTP.



**Date: 15-10-2025**

**Exercise-4:** Amazon S3 – Static Website Hosting (Multi-Page website), Versioning, Cross-Region Replication rule.

**Amazon S3 Static Website Hosting**

Amazon S3 can host a static website – [a website consisting of only HTML, CSS, JavaScript, images, etc. – no server-side scripting like PHP or Python].

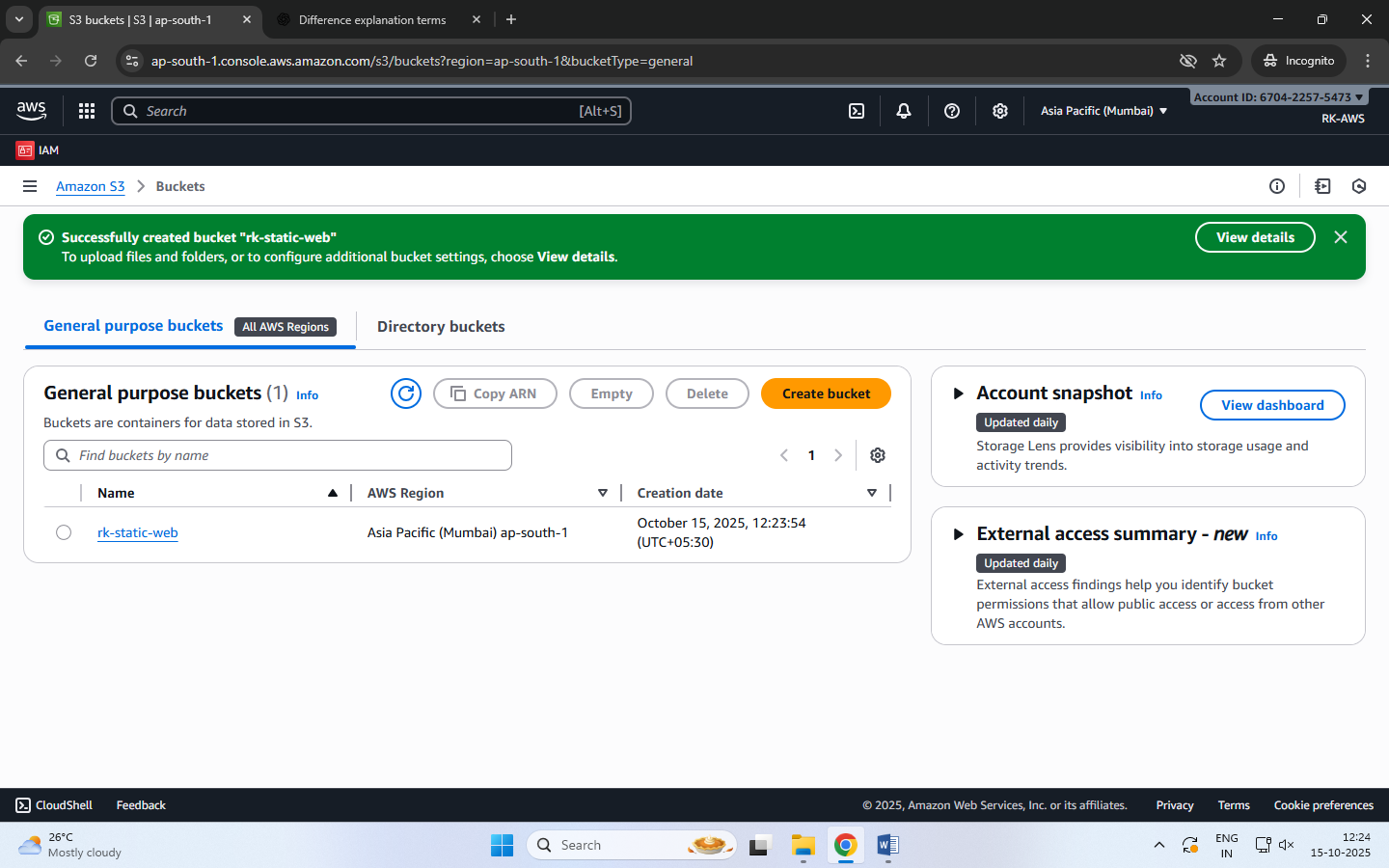
When you enable “Static Website Hosting,” your S3 bucket acts like a web server, and AWS provides a public website URL to access it.

You can create a multi-page static website (e.g., index.html, about.html, contact.html) and upload it to S3. Links within these pages allow users to navigate between them just like a normal website.

**Steps to Create a Multi-Page Static Website on S3**

**Step 1:** Create an S3 Bucket

* Open the AWS Management Console → Navigate to S3.
* Click Create bucket.
* Select Bucket type: General purpose
* Enter a unique bucket name (e.g., my-static-web-demo).
* Uncheck “Block all public access.”
* Click Create bucket.



**Step 2:** Prepare Website Files

Before uploading, organize your files in a folder structure as follows:

my-website/

│

├── index.html

├── about.html

├── contact.html

├── error.html

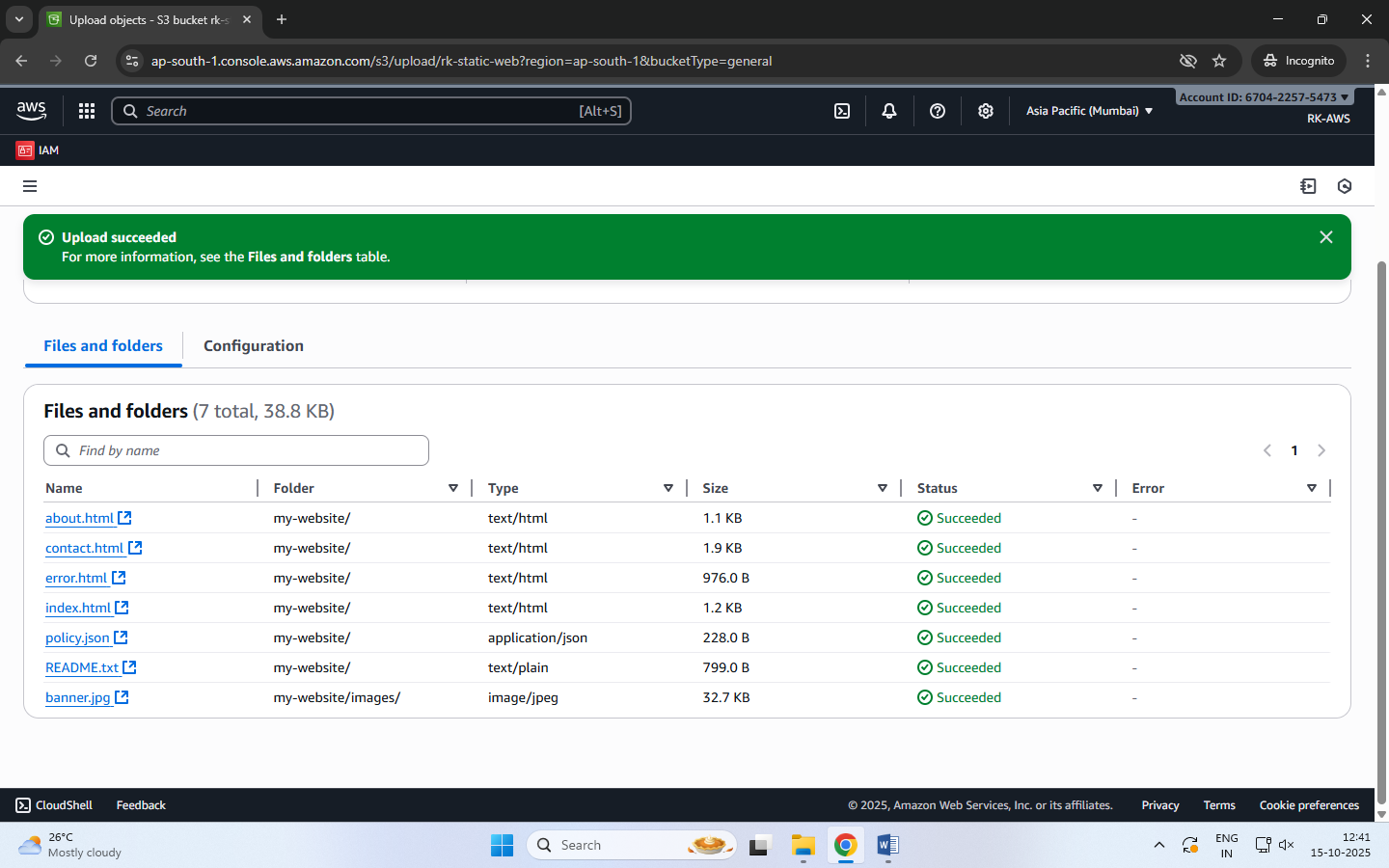
└── images/

└── banner.jpg

Each HTML file should include navigation links.

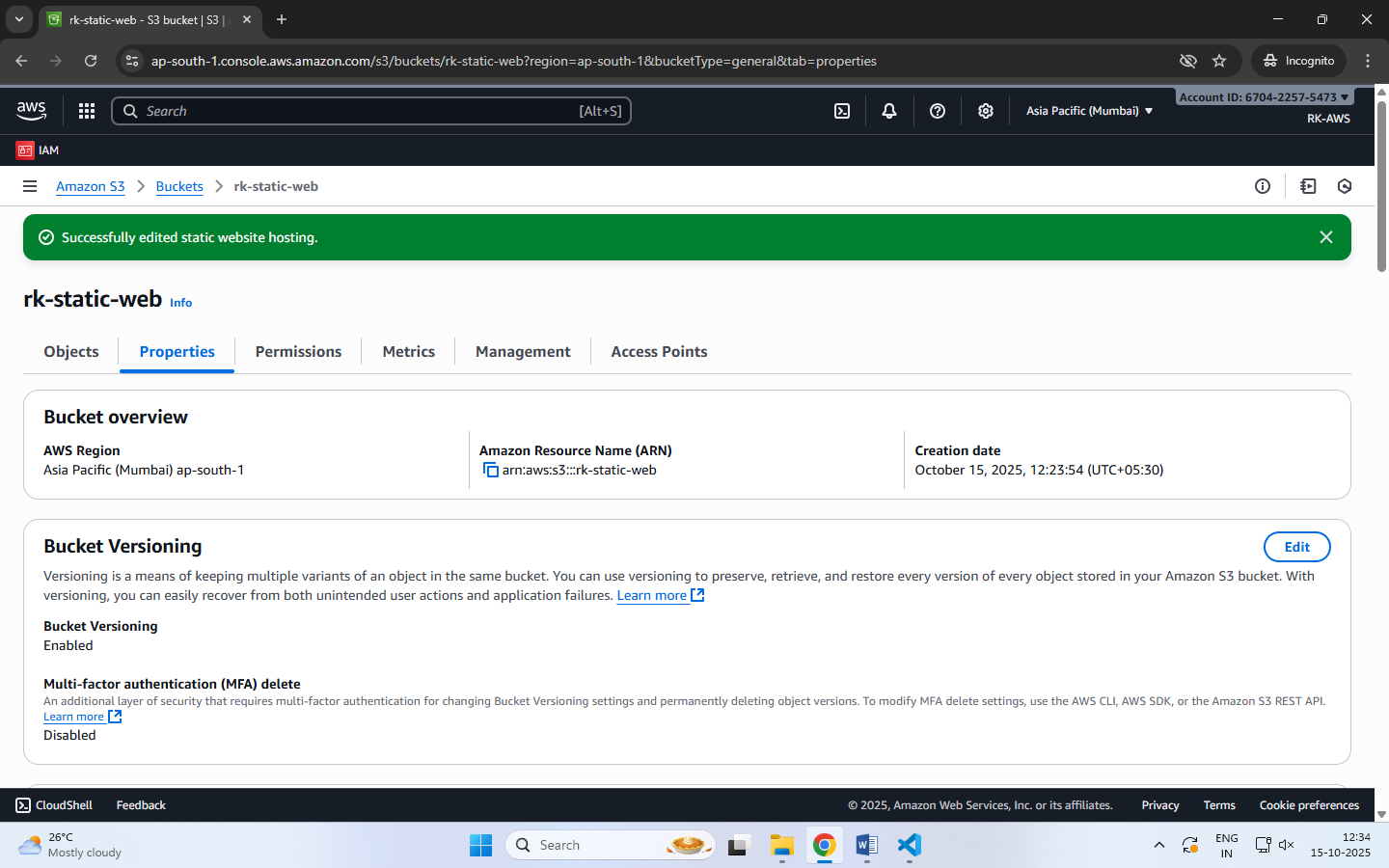
**Step 3:** Upload Website Files

* Open your S3 bucket → Click Upload.
* Add all files and folders (HTML, CSS, JS, images).
* Click Upload to store them in S3.



**Step 4:** Enable Static Website Hosting

* Go to the Properties tab of the bucket.
* Scroll down to Static website hosting → Click Edit.
* Choose Enable and select ‘Host a static website’.
* Set:
  + Index document: index.html
  + Error document: error.html
* Click Save changes.



**Step 5:** Make Files Public (Bucket Policy)

By default, your files are private. To make them public:

* Go to the Permissions tab → Bucket Policy → Edit.
* Paste the following policy (replace bucket name):

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

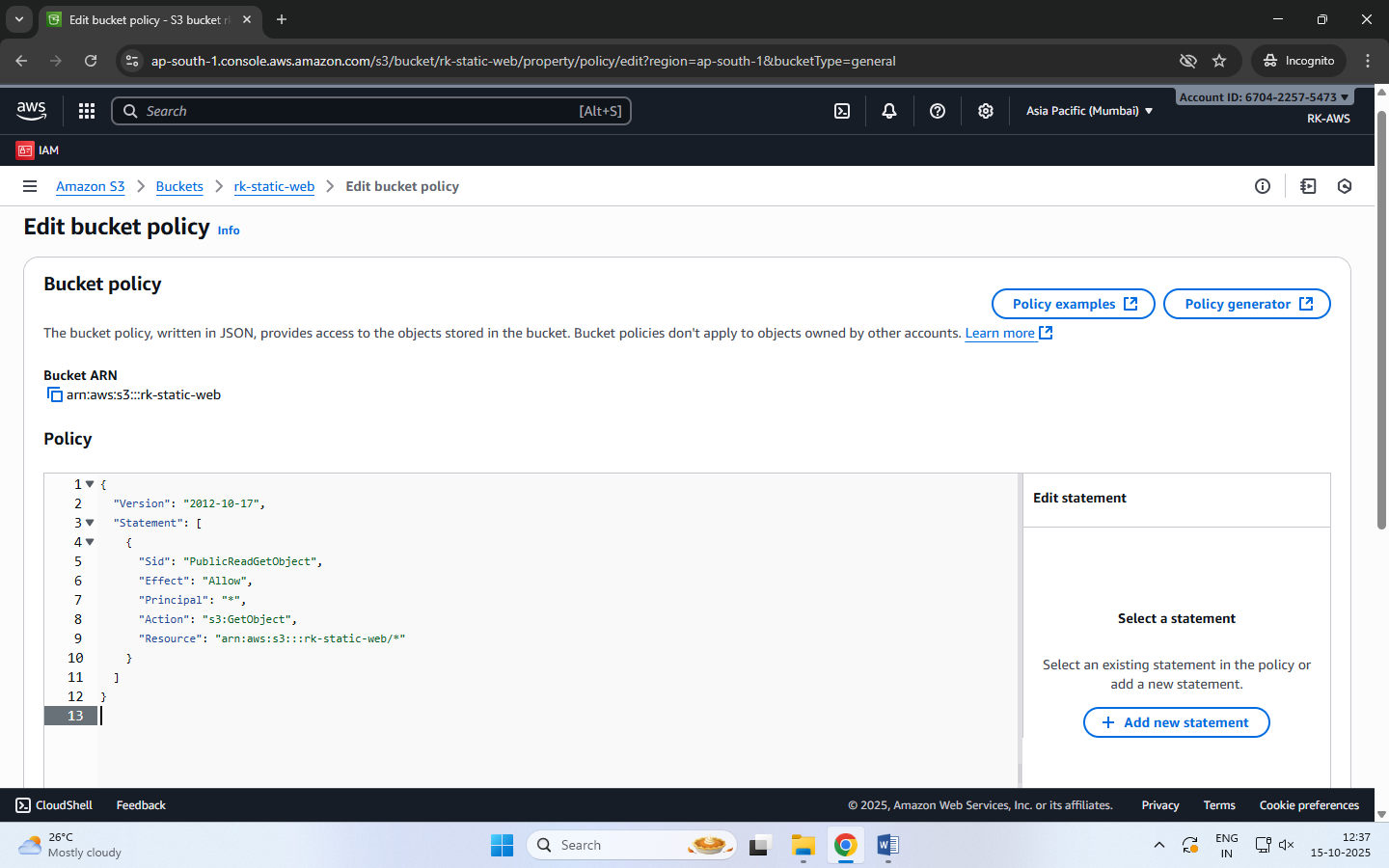
"Resource": "arn:aws:s3:::my-static-web-demo/\*"

}

]

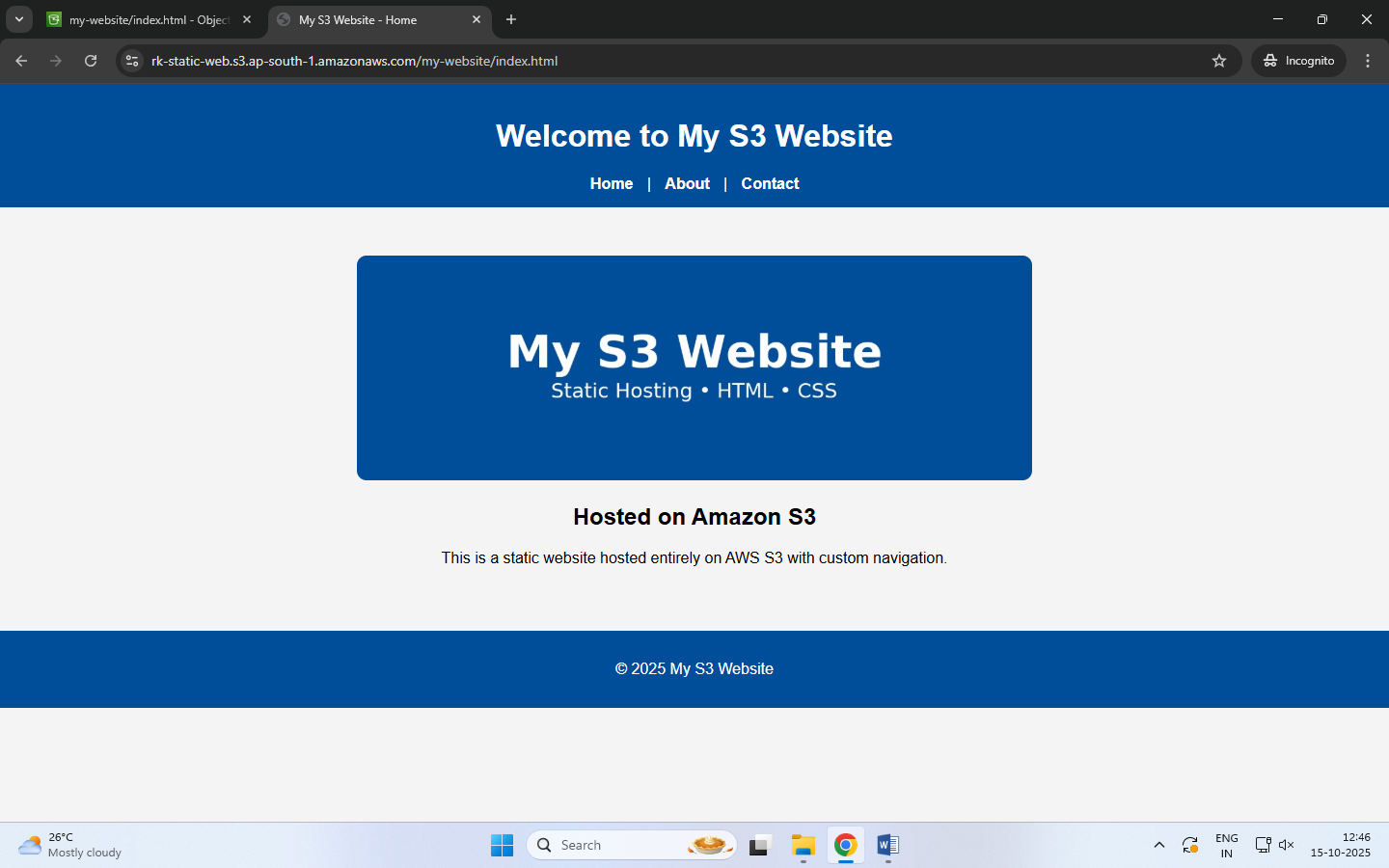
}

* Save the changes.



**Step 6:** Access Your Website

* Go to the Properties tab → Scroll to Static website hosting.
* Copy the Bucket Website Endpoint URL.
* Paste it into your browser — your homepage (index.html) should appear.
* Use the header links to navigate between pages (About, Contact, etc.).



**When Will the Error Page Be Shown?**

If a user enters a wrong URL or tries to access a file that doesn’t exist (e.g., /abc.html), Amazon S3 automatically displays the file you set as the **Error document** (error.html).

**Amazon S3 Versioning**

Versioning allows you to keep multiple versions of an object in a bucket.

If a file is accidentally deleted or overwritten, you can recover the previous version.

Each version gets a unique version ID.

**Steps to Enable Versioning**

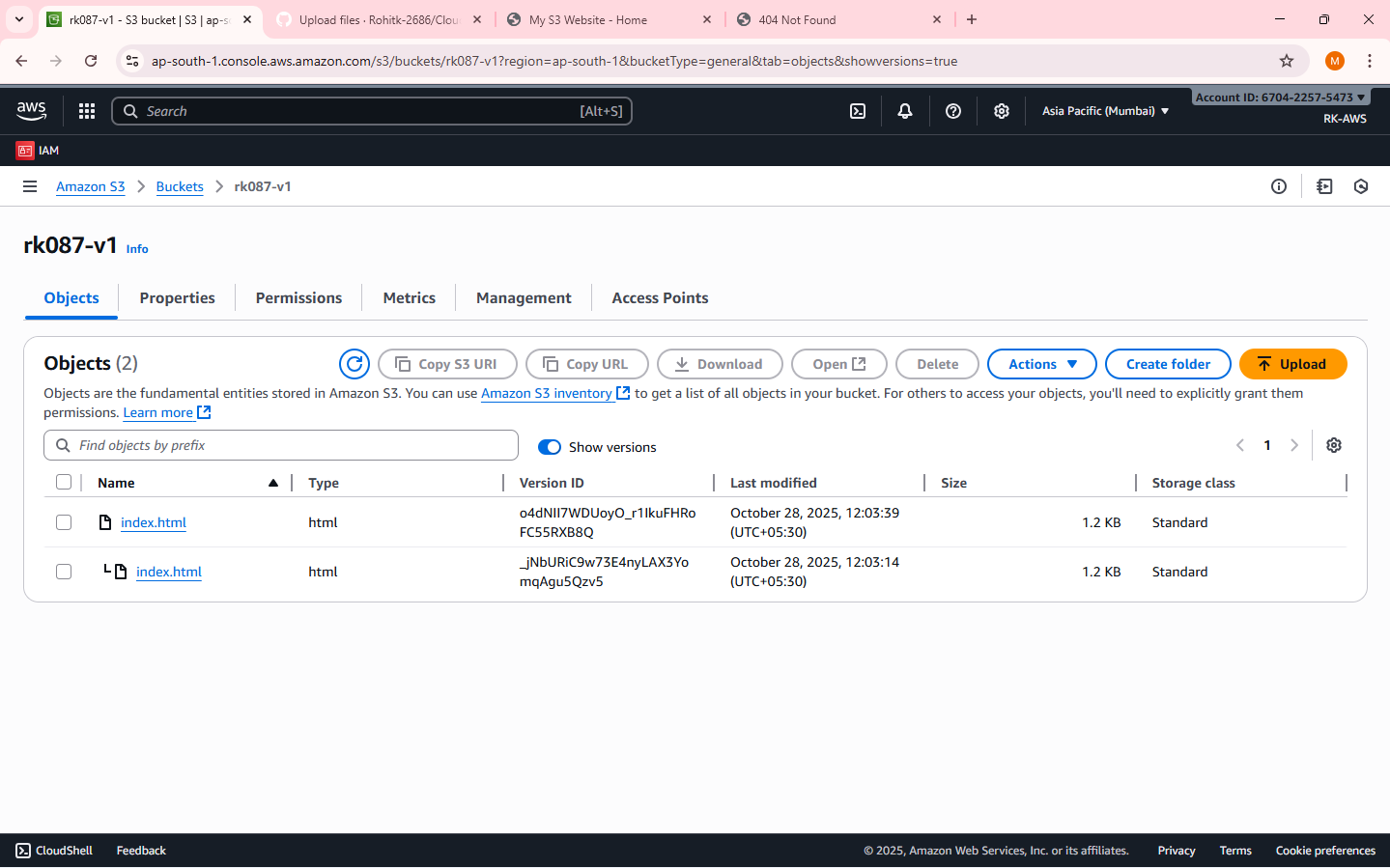
* Go to your S3 bucket
* Open the Properties tab
* Scroll to Bucket Versioning
* Click Edit → Enable
* Click Save changes

Now whenever you upload a file with the same name, S3 will keep both versions.

**Note:** You can view versions by clicking “List versions” in the bucket objects page.

**To Restore or Delete a Specific Version**

* Click the object name → Versions
* Select the desired version → Download / Delete  
  (Deleting only adds a delete marker — older versions are still stored.)



**Cross-Region Replication (CRR) – is a rule created on S3**

CRR automatically copies objects from one S3 bucket (source) to another (destination) in a different AWS Region.

Used for disaster recovery, compliance, or low-latency access in another region.

Requires Versioning to be enabled on both buckets.

**Steps to Set Up CRR**

**NOTE:** Enable Versioning on both:

Source bucket

Destination bucket

**Step 1:** Choose a different region before you create Destination bucket

Create Destination Bucket first.

**Step 2:** Create Source Bucket

Give replication permission:

Source bucket → Management tab→ Replication rules → Create rule



**Step 3:** Create Replication Rule page

Enter a Replication rule name

Status: Enabled

Source bucket section:

Choose a rule scope: select “Apply to all objects in the bucket”

Destination:

Select “Choose a bucket in this account”

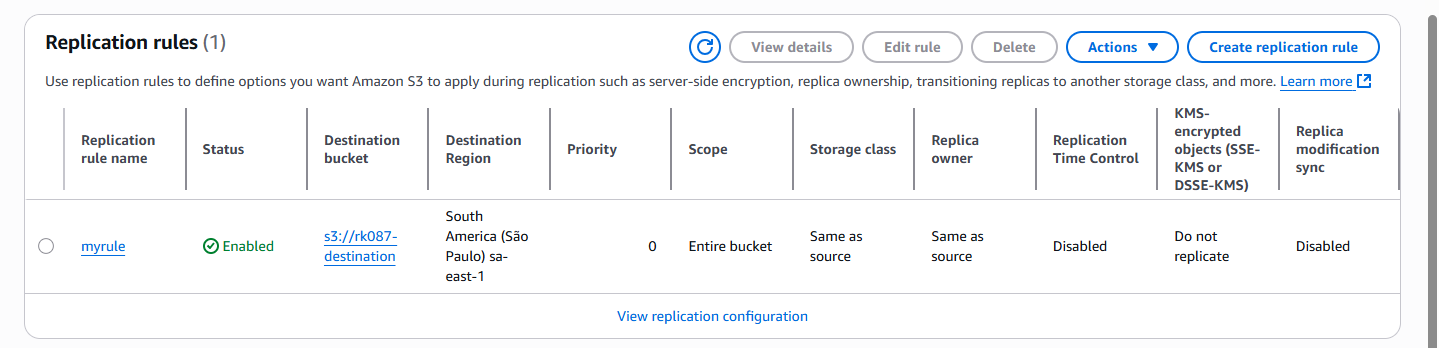
Bucket name: Select the destination bucket

IAM role:

Select “Create new role”

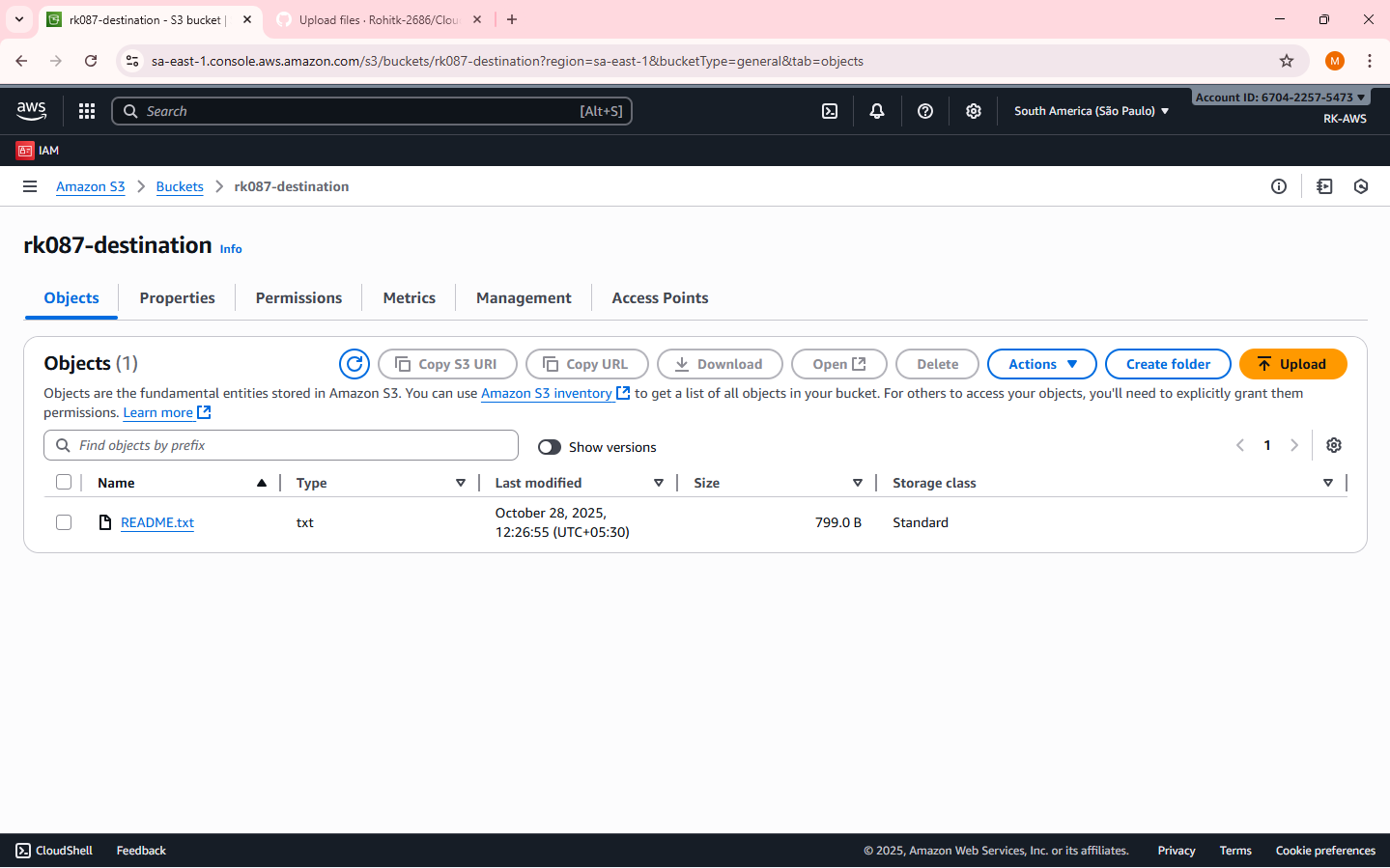
[An IAM role gives Amazon S3 permission to replicate objects from your source bucket to your destination bucket.  
S3 replication won’t work unless you assign (or create) a role with the right permissions.

IAM Role: Either create a new role automatically or choose an existing one]



**Step 4:** Save

Any new objects uploaded to the source bucket will automatically replicate to the destination region.



**Note:** Replication is not retroactive — only new uploads after enabling CRR are copied.

Reminder – Resource cleanup – to release/delete/terminate the resources created.