**Task on S3**

**Amazon S3 Introduction**

Amazon **Simple Storage Service (S3)** is a **scalable, durable, and secure** cloud storage service offered by AWS. It is designed for storing and retrieving any amount of data from anywhere on the web.

**Key Features of Amazon S3**

1. **Scalability** – Automatically scales to store unlimited amounts of data.
2. **Durability** – Provides **99.999999999% (11 9s) durability** for stored objects.
3. **Availability** – Designed for **high availability** with multiple storage classes.
4. **Security** – Supports **IAM policies, encryption, and access control lists**.
5. **Cost-Effective** – Offers **pay-as-you-go pricing** with different storage classes.

**How Amazon S3 Works**

* Data is stored as **objects** in **buckets**.
* Objects can be retrieved via **unique URLs** or the AWS CLI/API.
* Supports **Lifecycle Policies** for automatic transition and deletion.
* Enables **Cross-Region Replication (CRR)** for redundancy.

**Use Cases**

* **Backup & Archiving** – Store large datasets securely.
* **Static Website Hosting** – Serve website content directly from S3.
* **Big Data & Analytics** – Store and process huge datasets efficiently.
* **Media Storage** – Host images, videos, and application assets.

**1) Create s3 bucket and upload some objects to s3**

Create an S3 Bucket

You can create an S3 bucket via the AWS Management Console or CLI.

Using AWS Console

- Sign in to the AWS Management Console.

- Navigate to S3.

- Click Create Bucket.

- Enter a unique Bucket Name (e.g., task-s3bucket29).

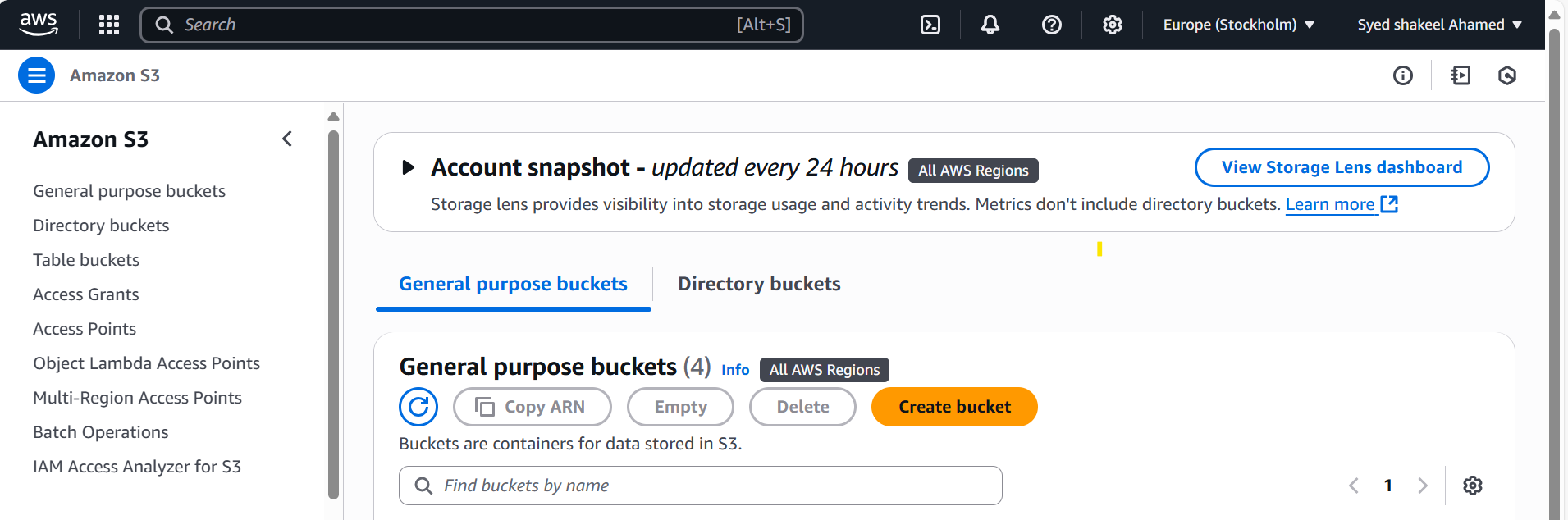
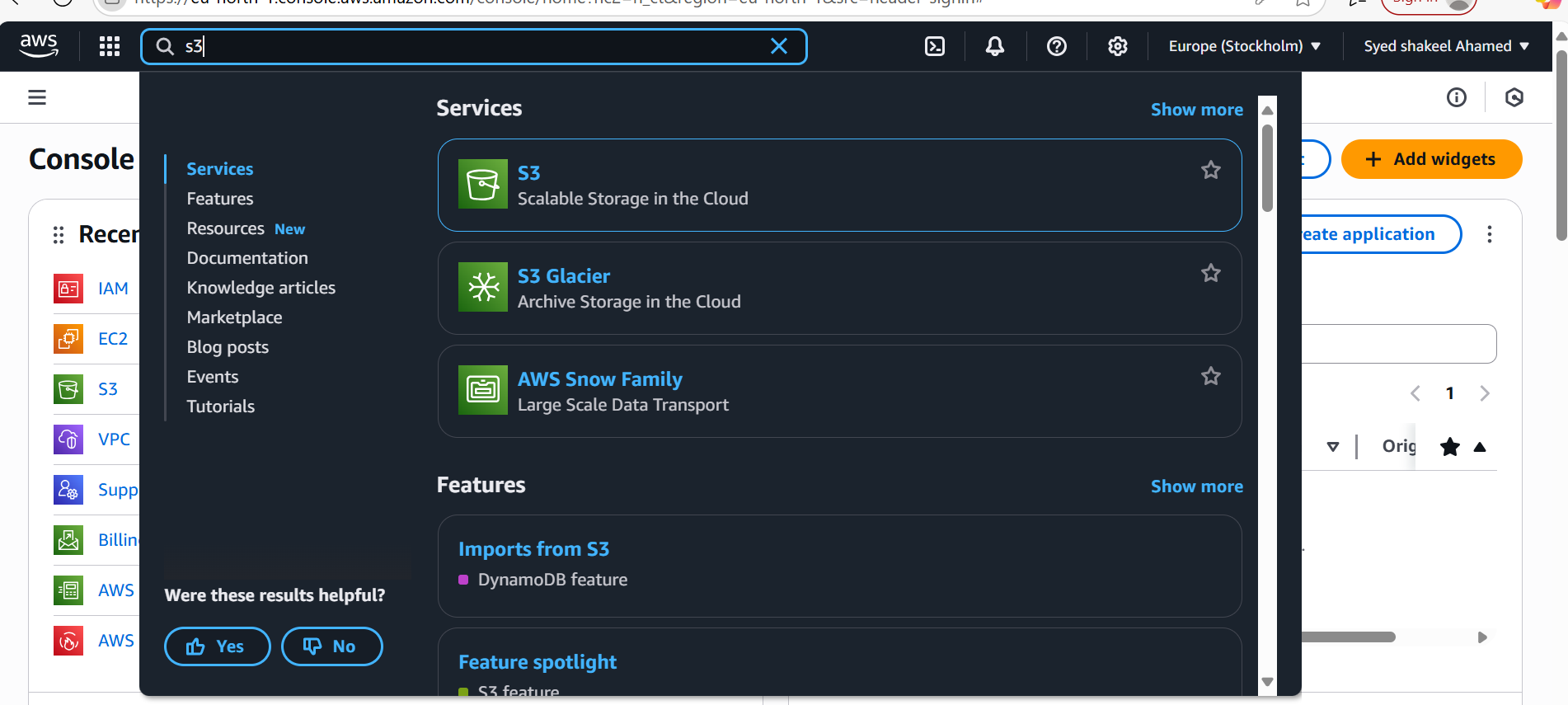
- Choose a Region where the bucket should reside.

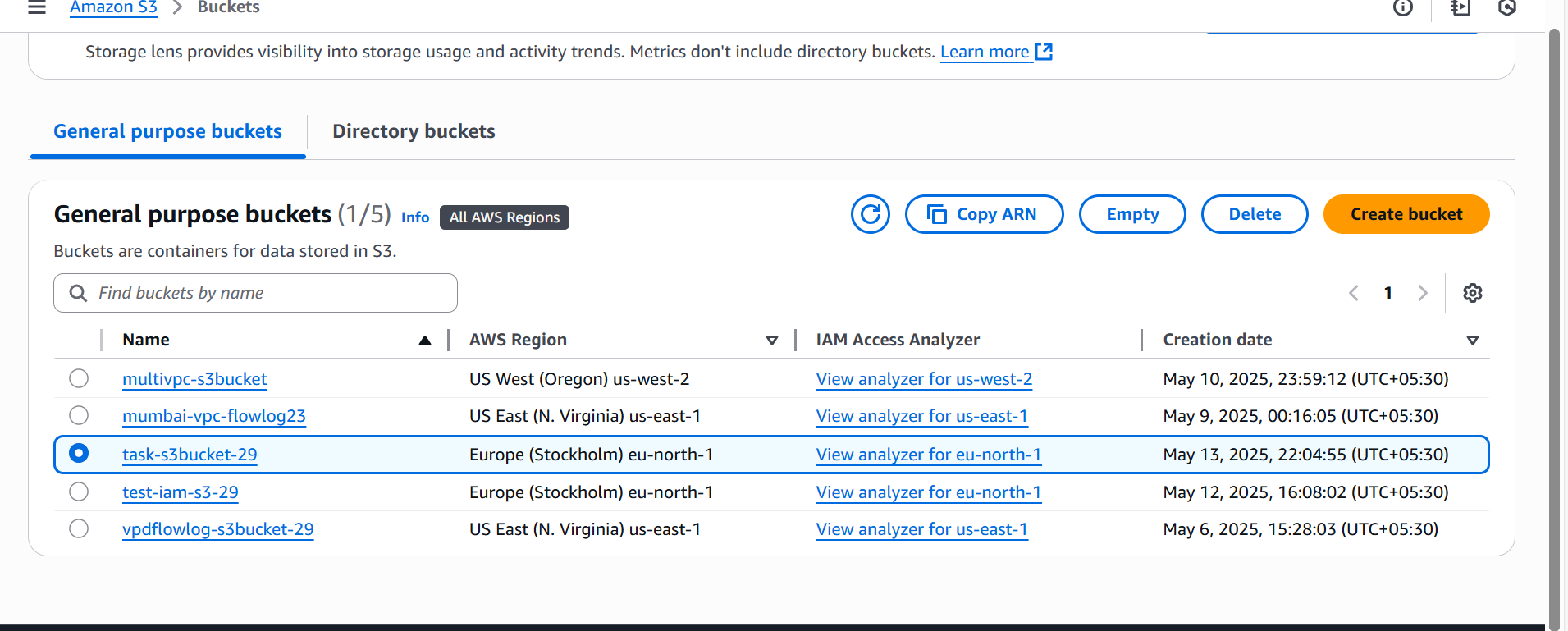
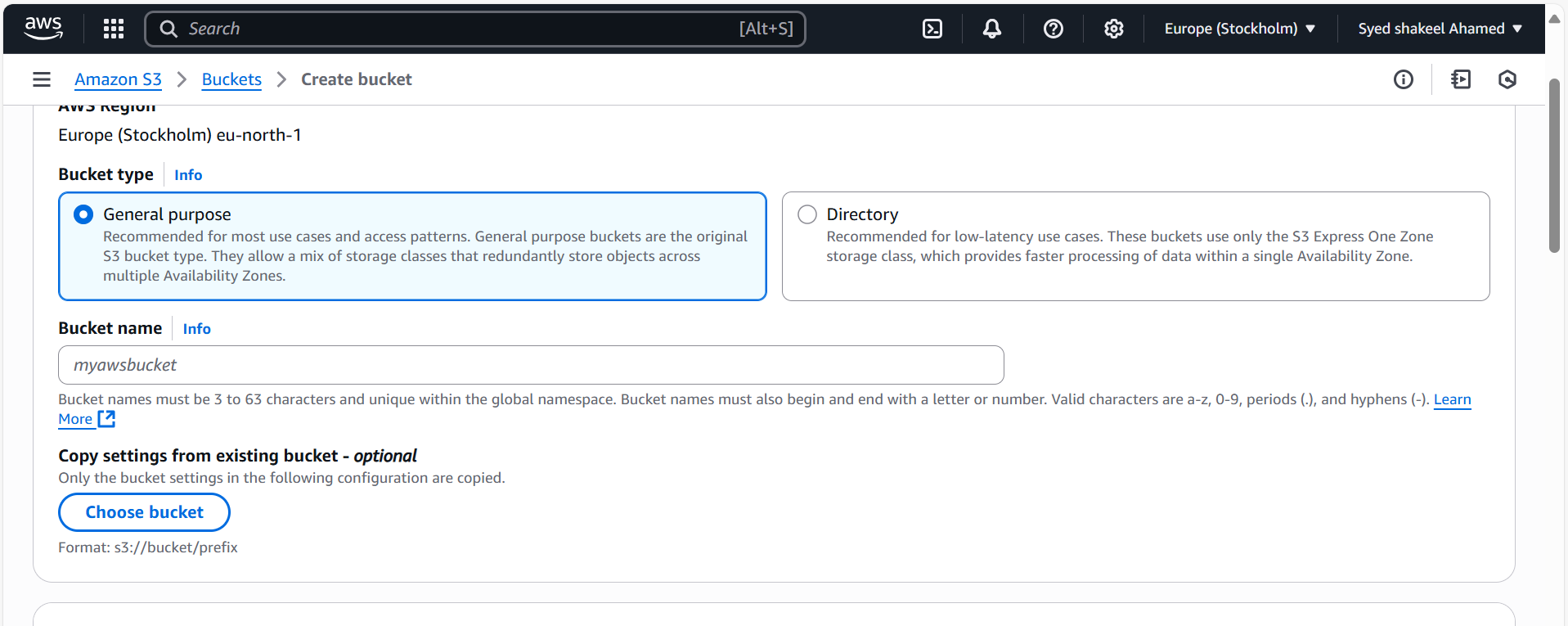
- Configure settings:

- Enable Versioning (if needed).

- Set Public Access Settings (recommended: block public access).

- Click Create Bucket.





Upload Objects to S3

You can upload files using the Console or CLI.

Using AWS Console

- Open the S3 Console.

- Click on your bucket (my-s3-bucket).

- Click Upload.

- Select files from your local system.

- Configure object settings:

- Set storage class (e.g., Standard, Glacier).

- Configure encryption (optional).

- Click Upload.

Set Permissions (Optional)

To Make an Object Public (If Allowed)

Enable Lifecycle Policies (Optional)

To automatically transition or delete objects, use lifecycle policies.

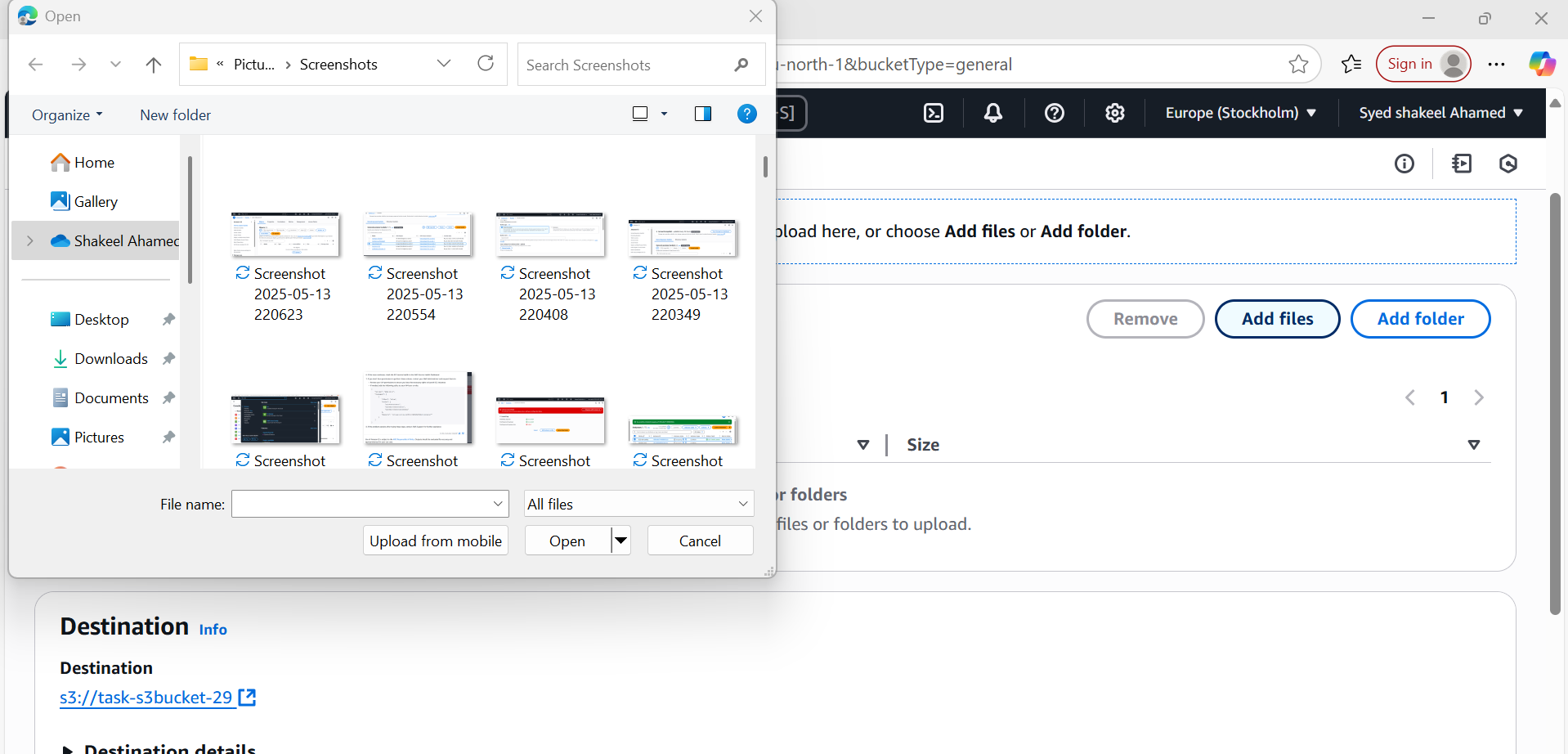
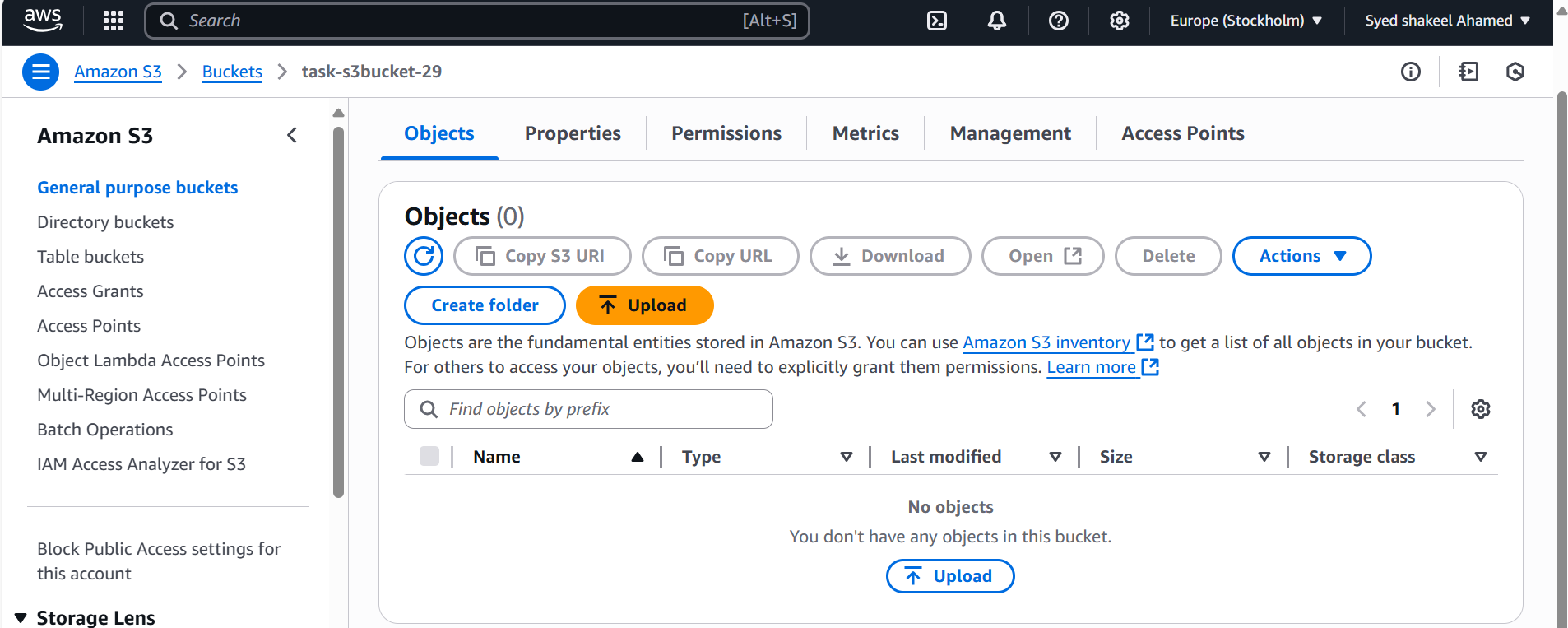
Example: Transition objects to Glacier after 30 days.

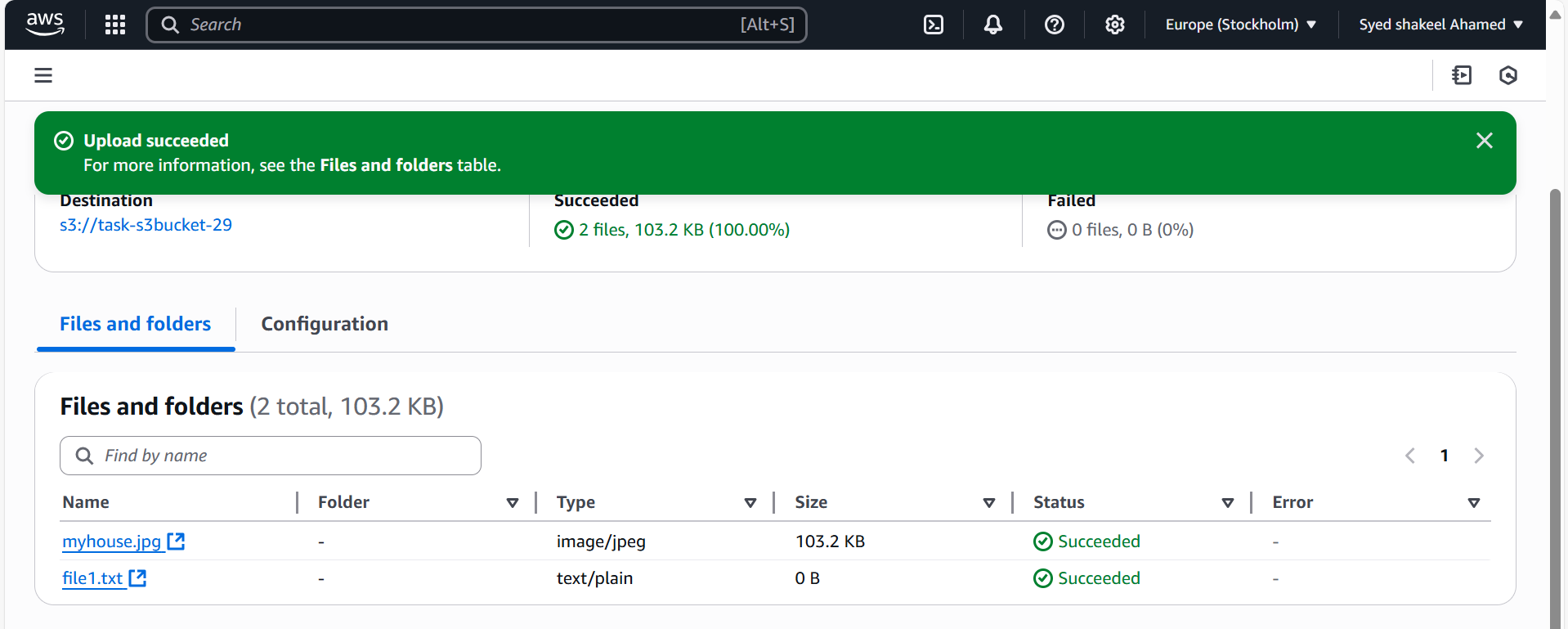
- Go to S3 Bucket settings.

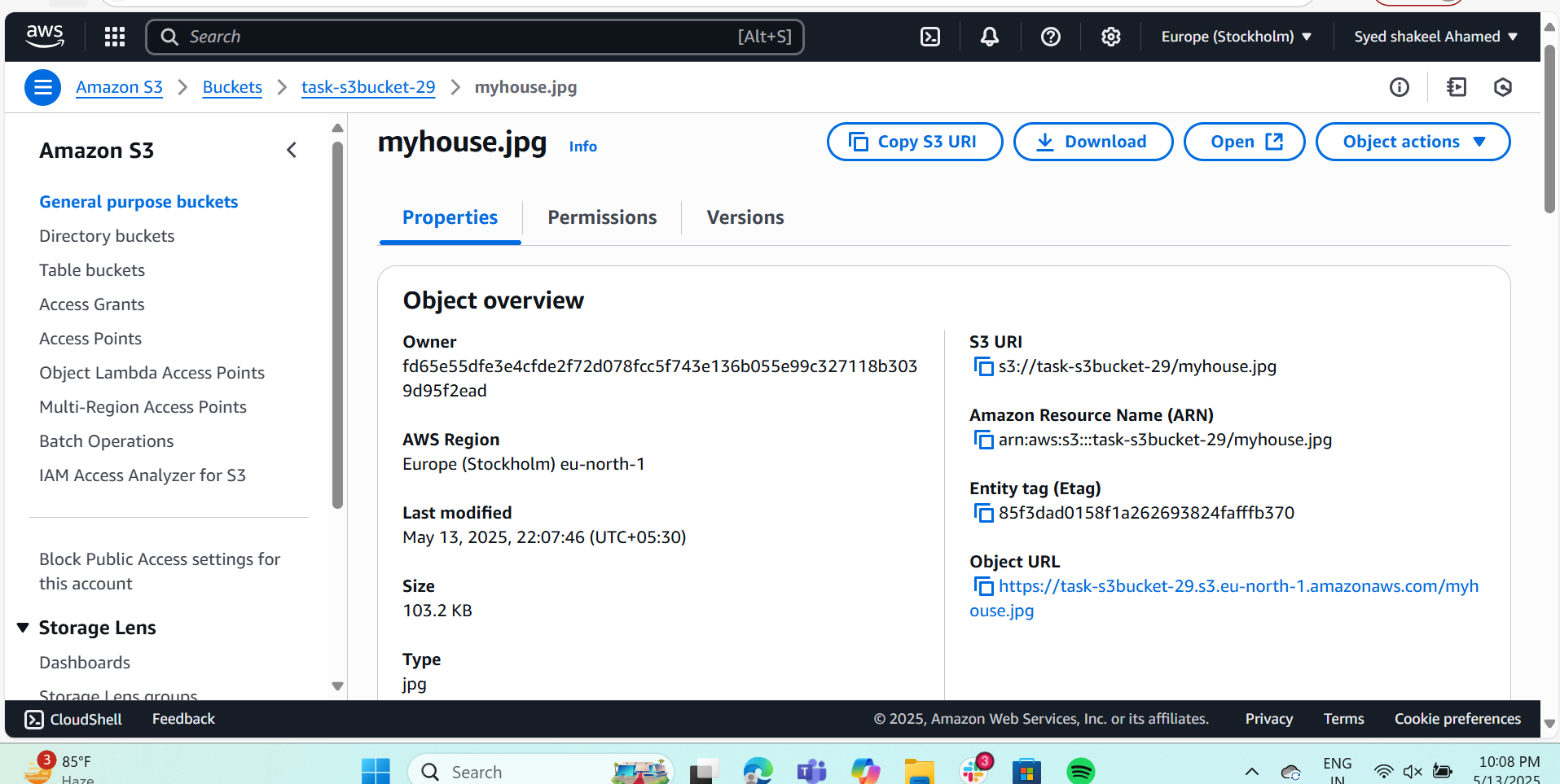
- Navigate to Management → Lifecycle Rules.

- Set transition rules.

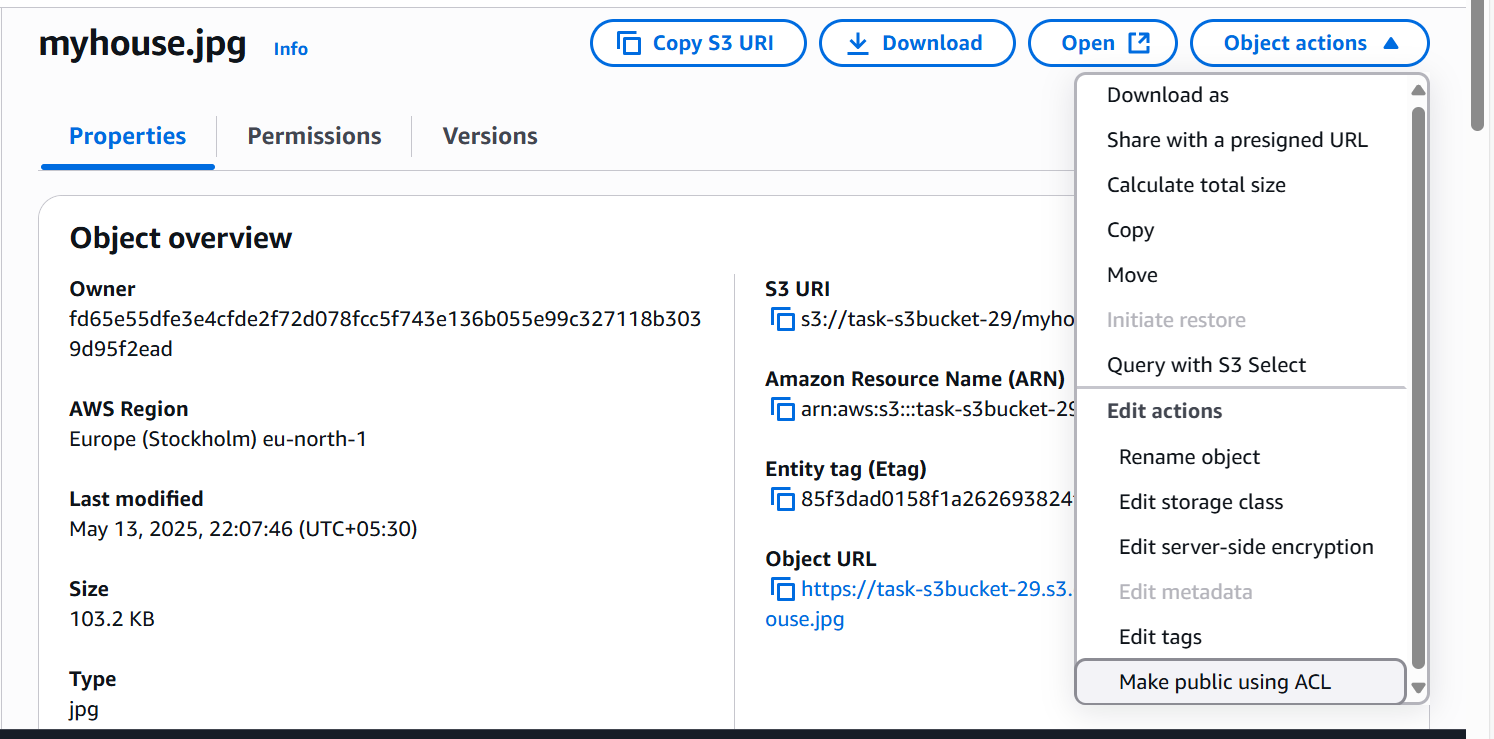
- Save changes.

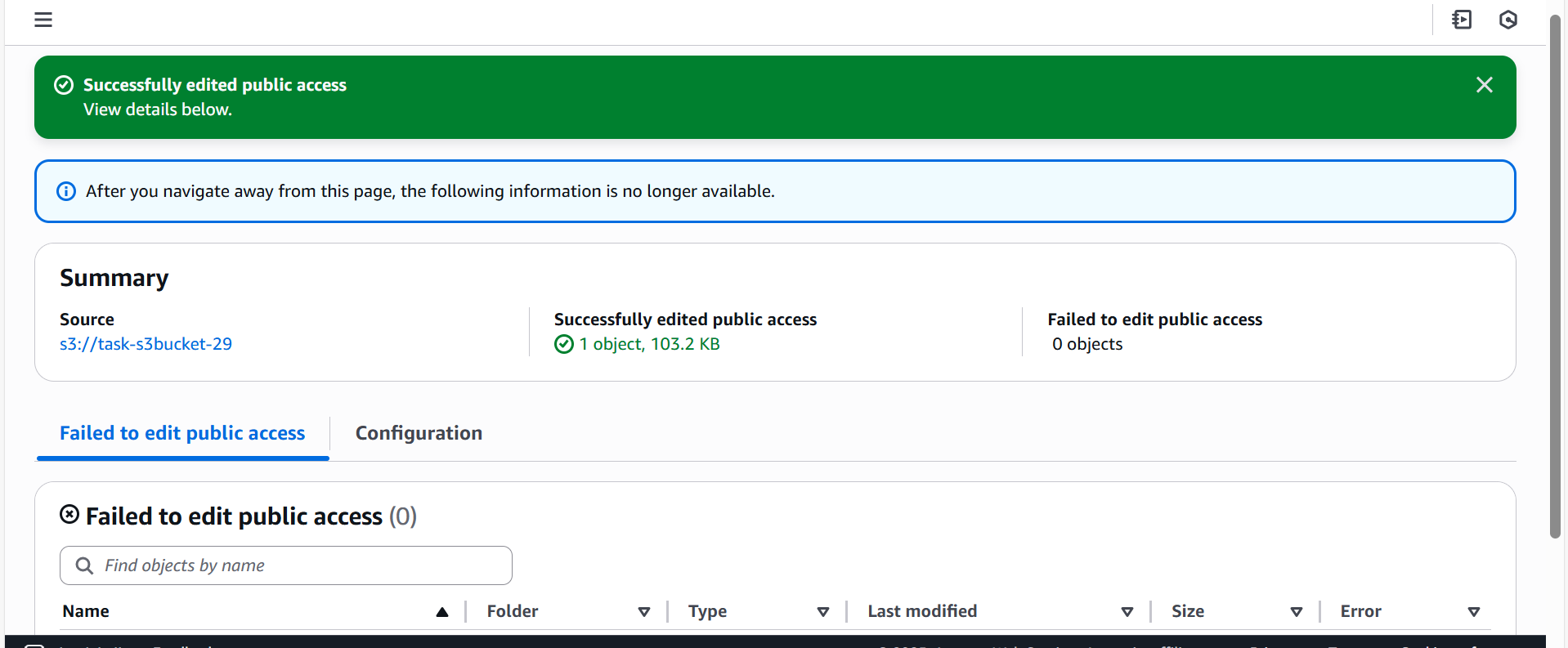


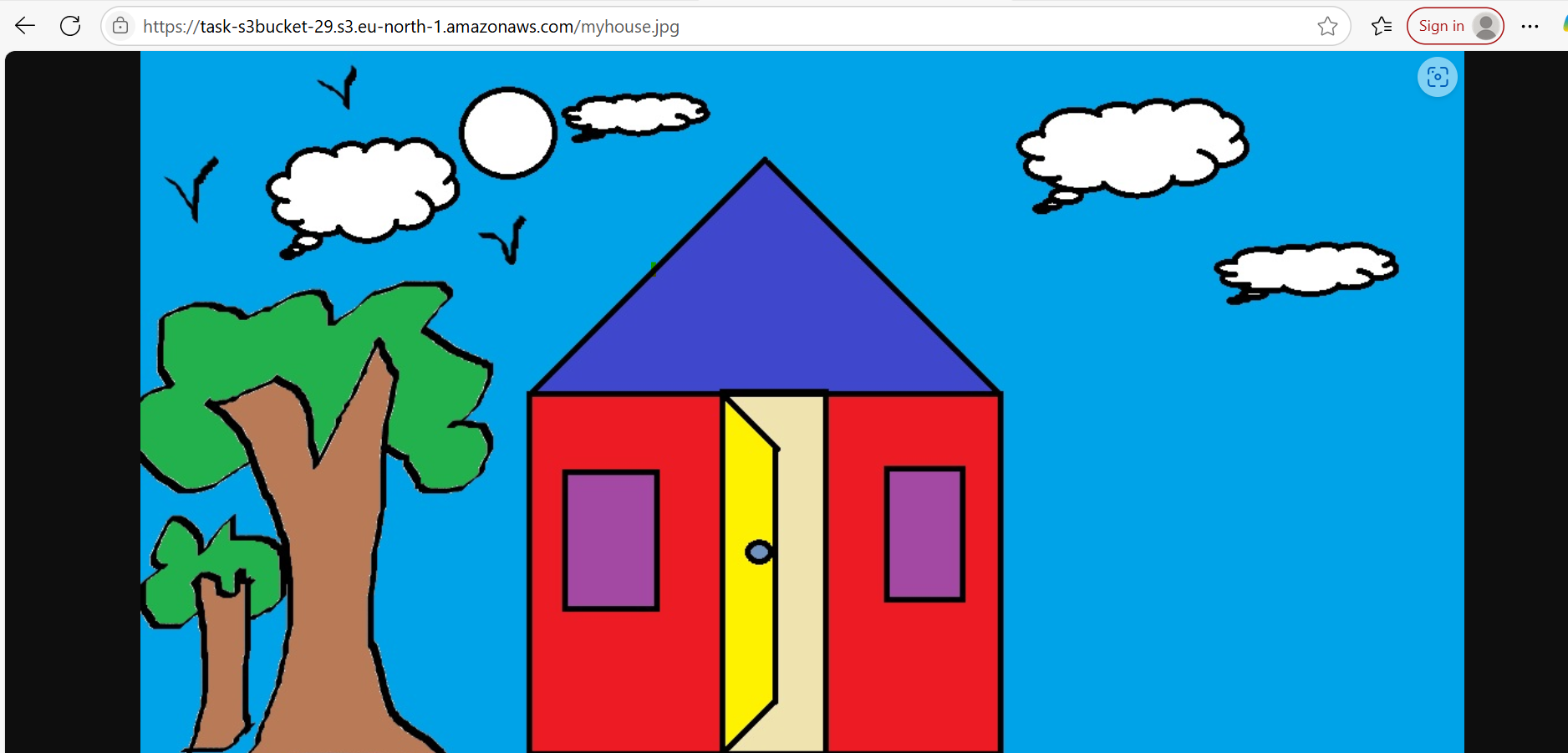












**2) Deploy static website in s3 bucket**

Enable Static Website Hosting

- Open the S3 Console.

- Select your bucket (tsk3-S3bucket29).

- Go to Properties → Static Website Hosting.

- Choose Enable.

- Select "Host a static website".

- Enter index document: index.html.

- Enter error document (optional): error.html.

- Click Save changes.

Upload Website Files

Using AWS Console

- Open the S3 Console.

- Select your bucket (my-static-site).

- Click Upload → Select files (index.html, styles.css, etc.).

- Click Upload.

Configure Bucket Policy for Public Access

Since S3 blocks public access by default, you need a bucket policy.

- Open the S3 Console.

- Navigate to Permissions → Bucket Policy.

- Add the following policy:

- Click Save Changes.

Access the Website

- Open the S3 Console.

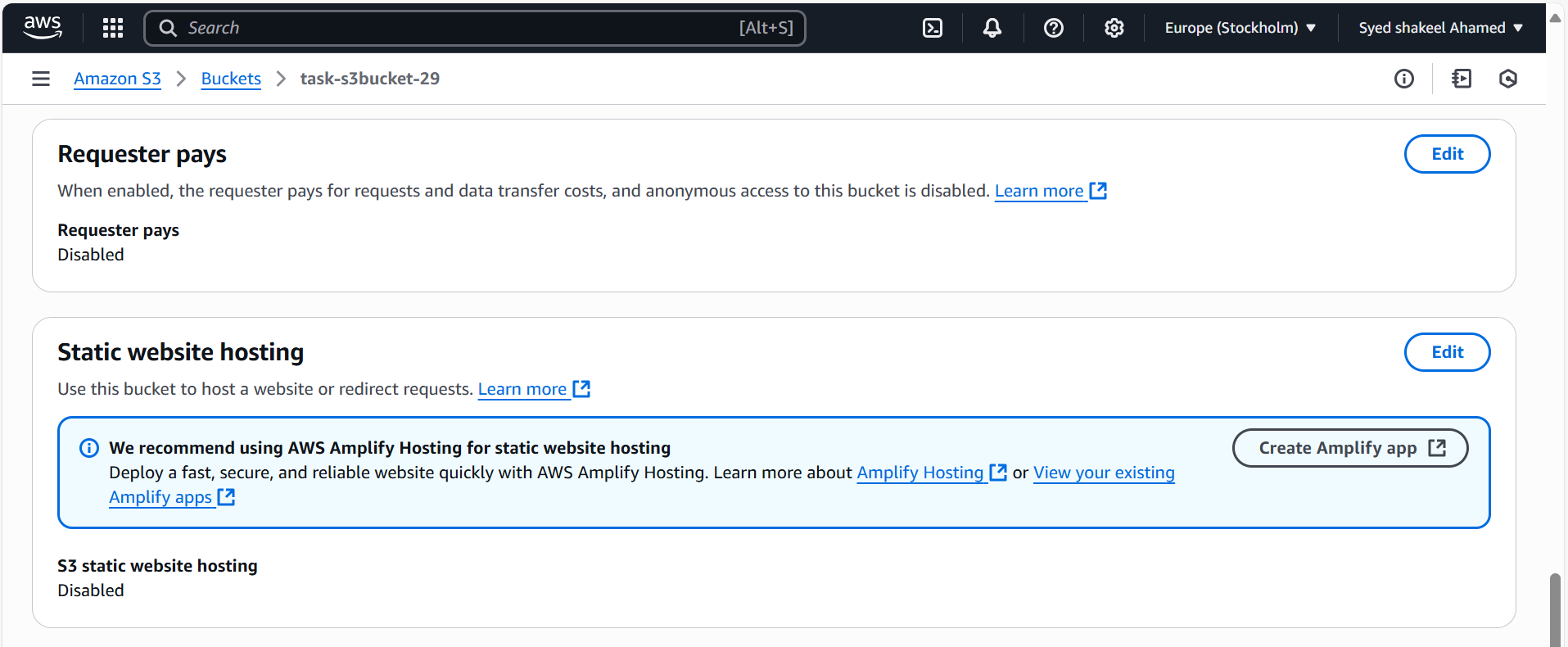
- Navigate to Properties → Static Website Hosting.

- Copy the Website Endpoint http://task-s3bucket-29.s3-website.eu-north-1.amazonaws.com

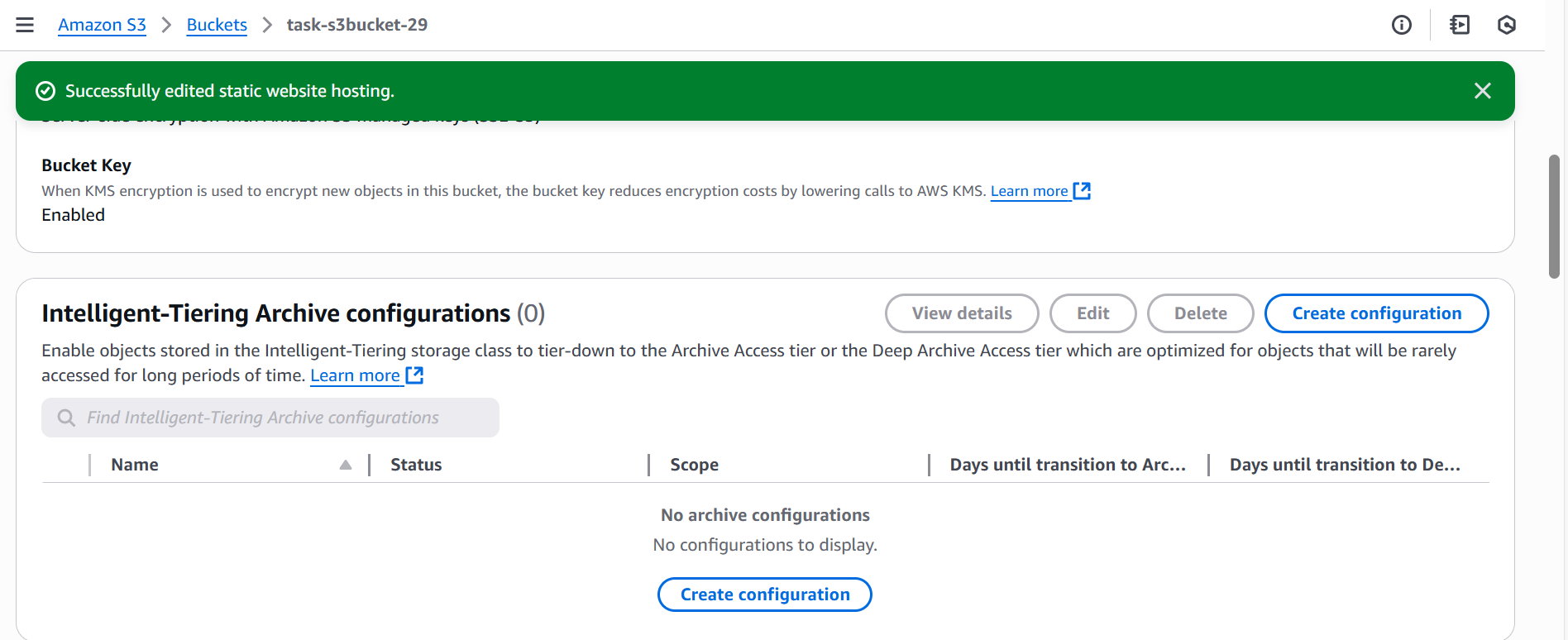
- Paste it in your browser to view your website.

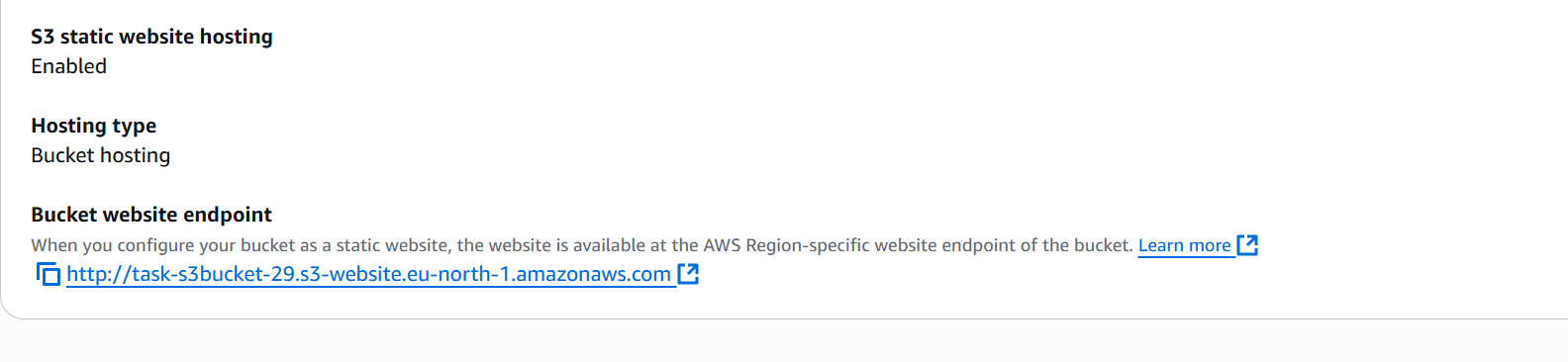
You will be successfully able to access the index.html

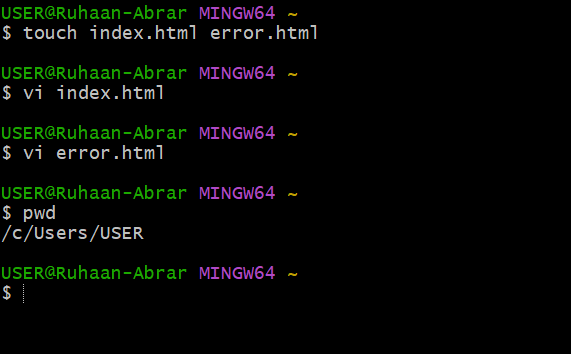
To access error.html page you need to delete the index.html object then only it will be accessible through browser

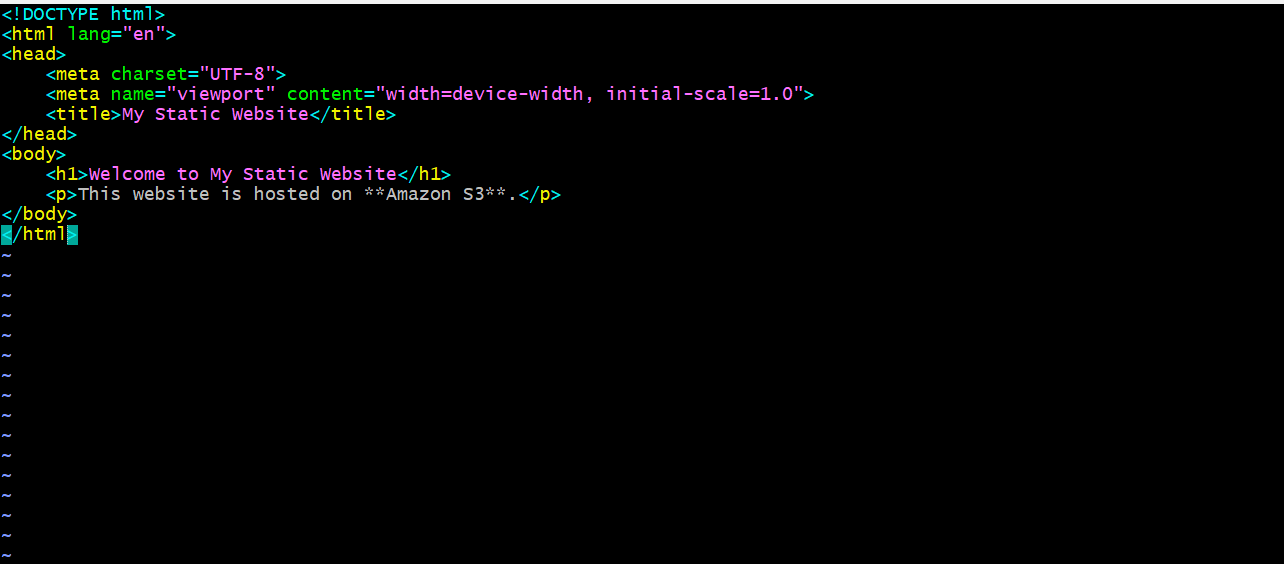


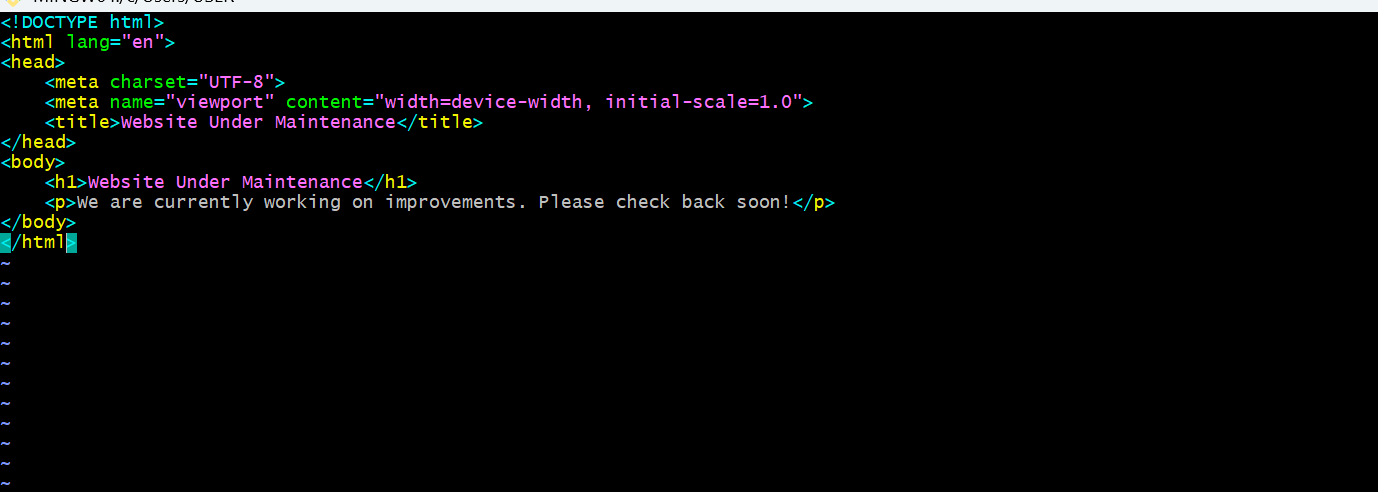


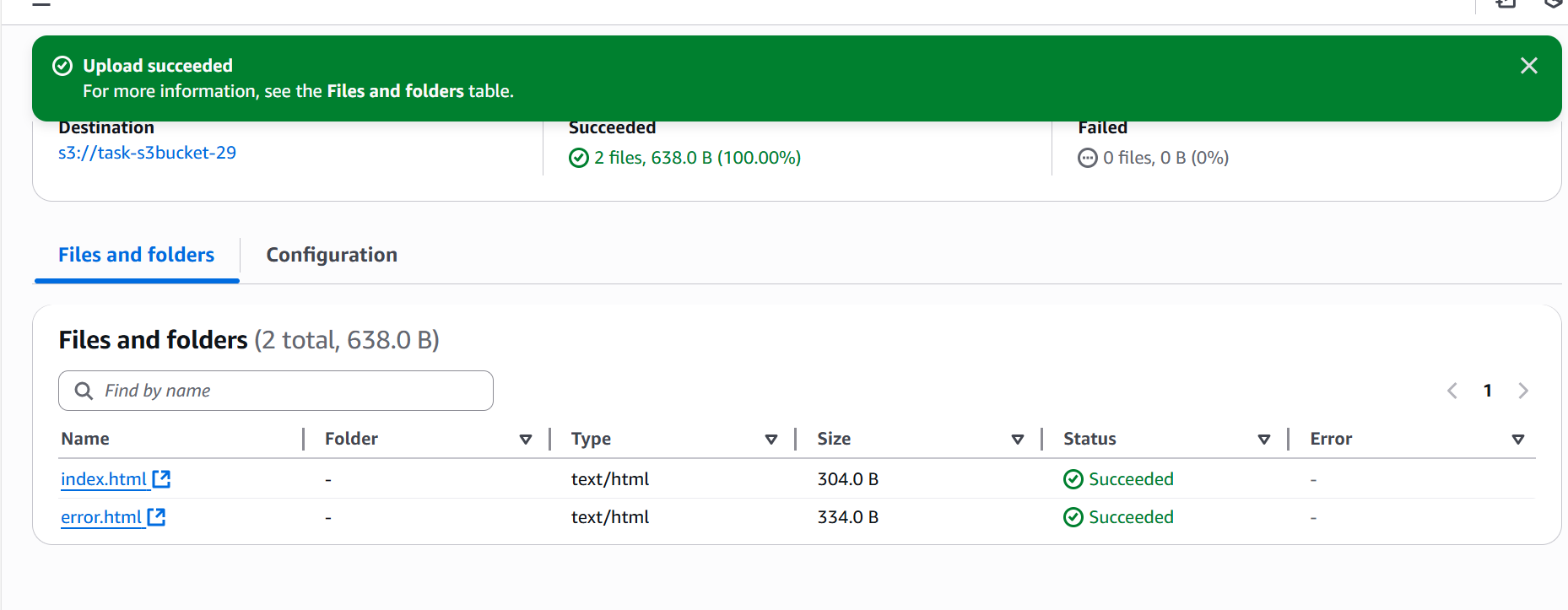
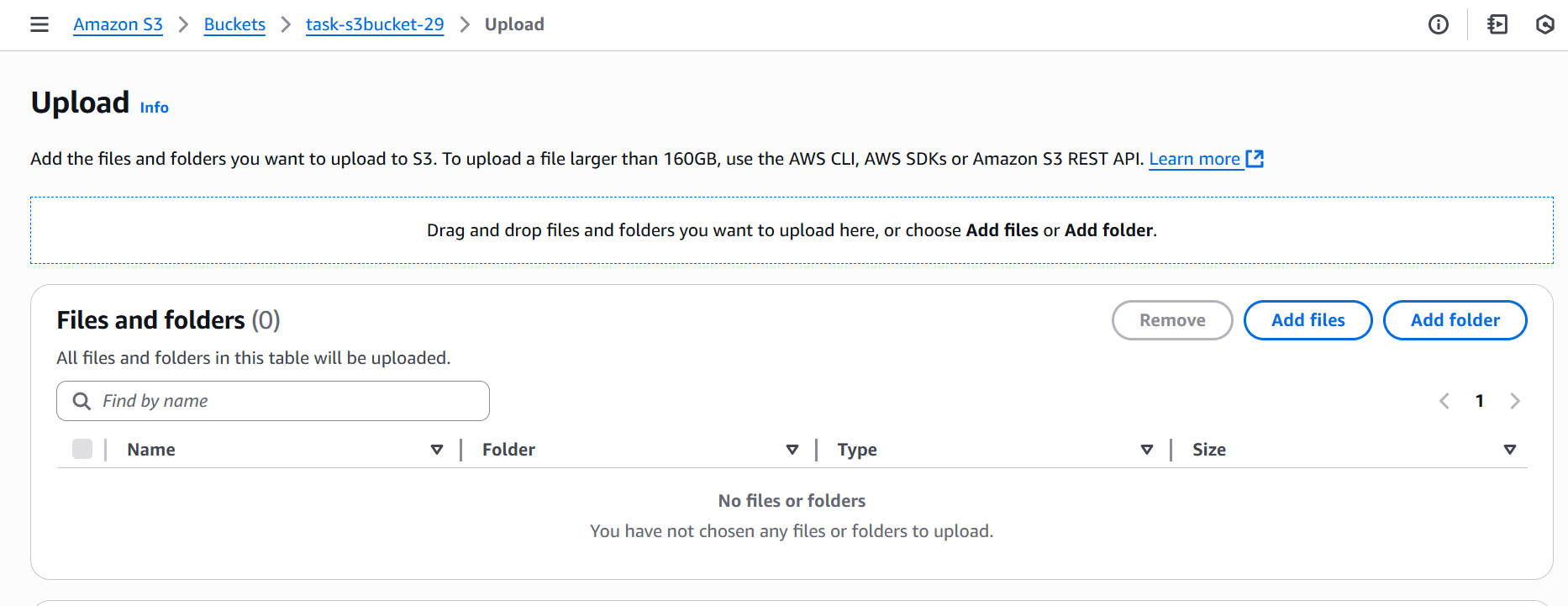




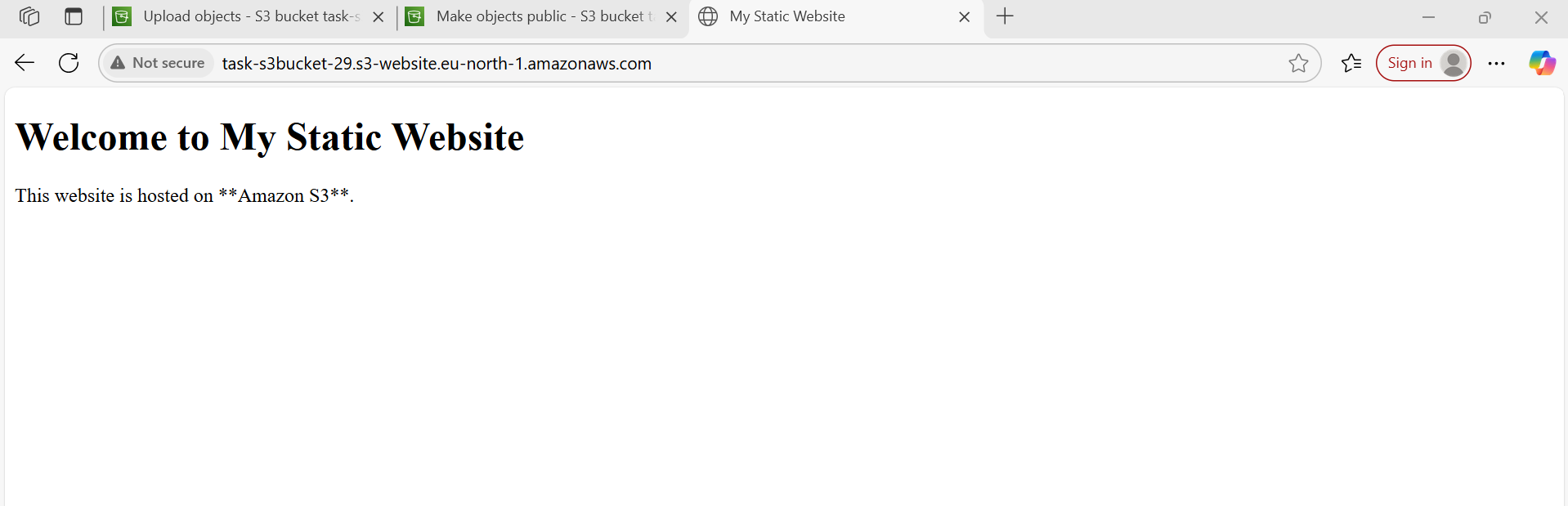




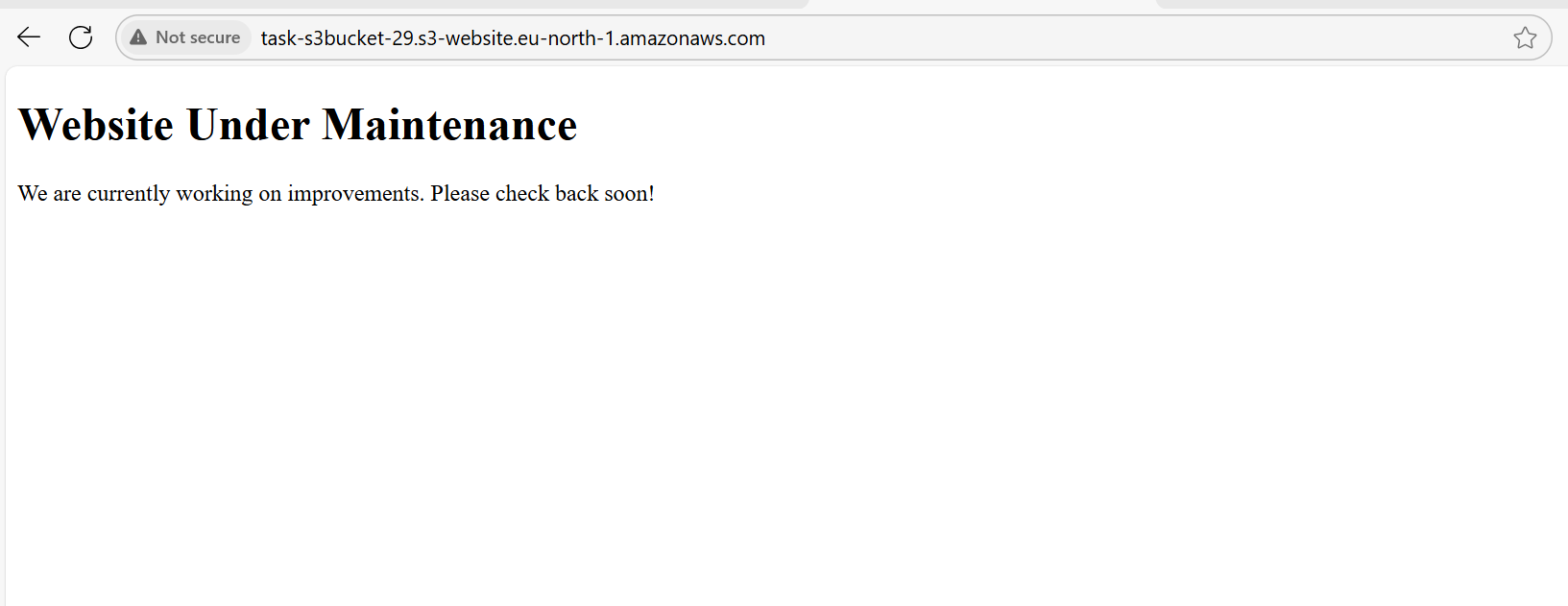
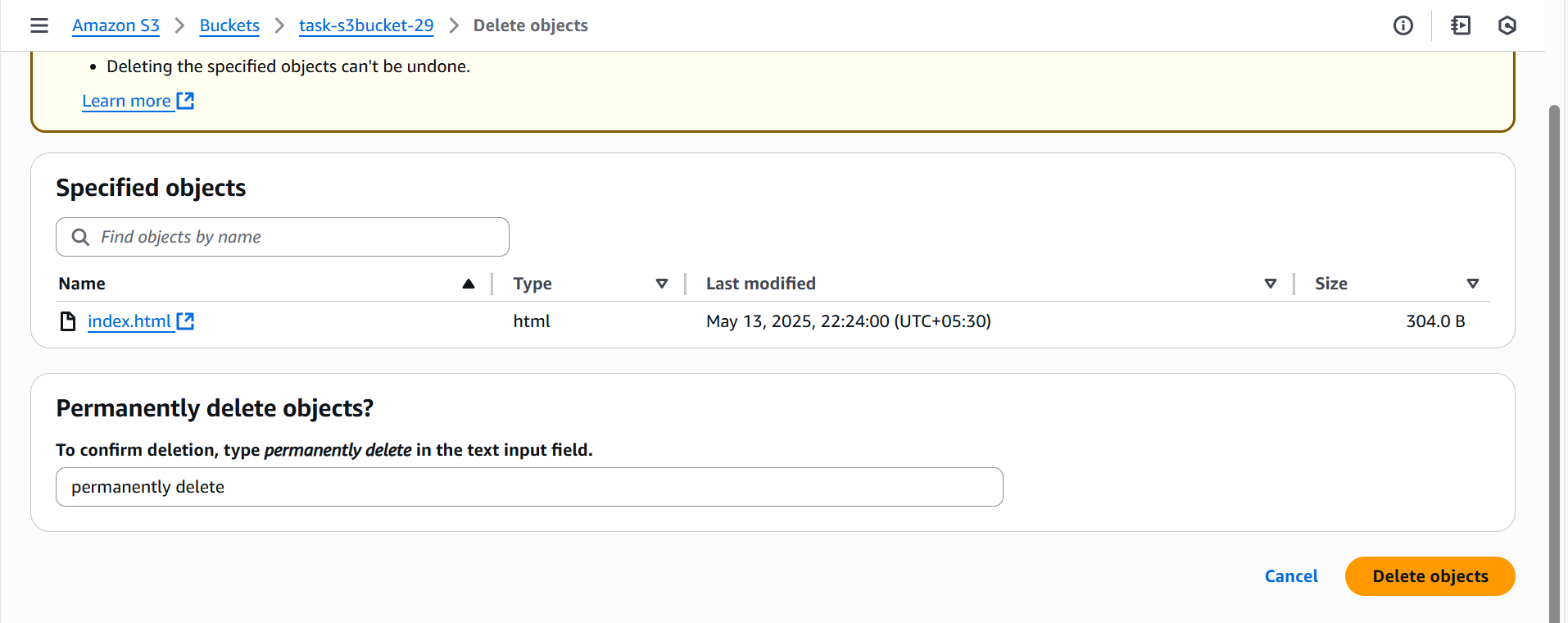




Index.html



Error.html



**3) Enable cross region replication on s3 buckets.**

Enabling Cross-Region Replication (CRR) on S3

Prerequisites

- Two S3 buckets:

- Source Bucket (e.g., source-bucket)

- Destination Bucket (e.g., destination-bucket)

- IAM role with necessary replication permissions.

- Versioning enabled on both buckets.

Enable Versioning on Both Buckets

Using AWS Console

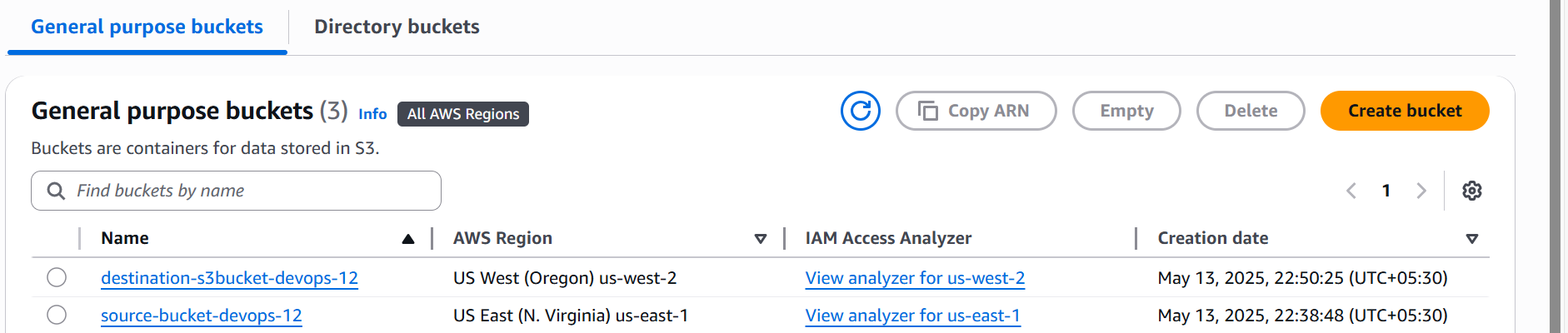
- Open the S3 Console.

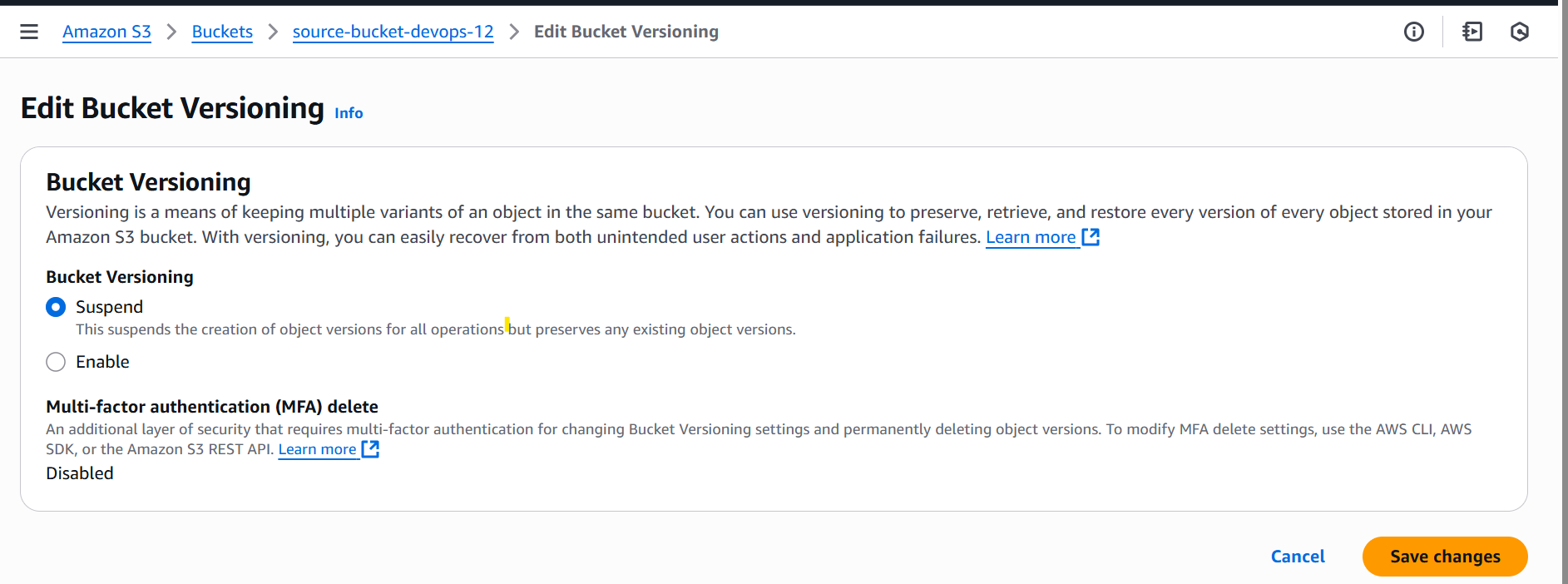
- Select Source Bucket (source-bucket).

- Navigate to Properties → Bucket Versioning.

- Click Enable Versioning → Save changes.

- Repeat steps 1-4 for Destination Bucket (destination-bucket).





Configure Replication Rule

Using AWS Console

- Open the S3 Console.

- Select Source Bucket → Navigate to Management.

- Click Replication Rules → Create Replication Rule.

- Enter Rule Name (e.g., CRR-Replication).

- Choose Destination Bucket (destination-bucket).

- Select IAM Role (created in Step 2).

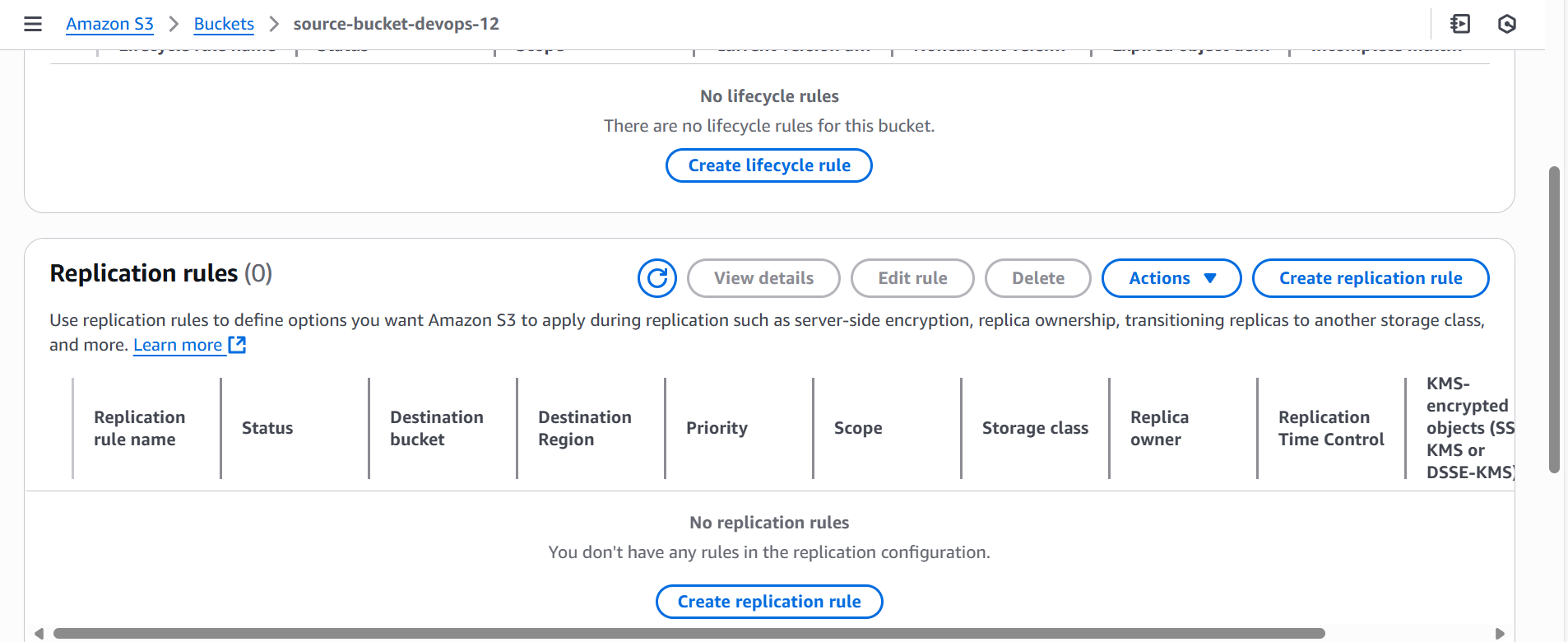
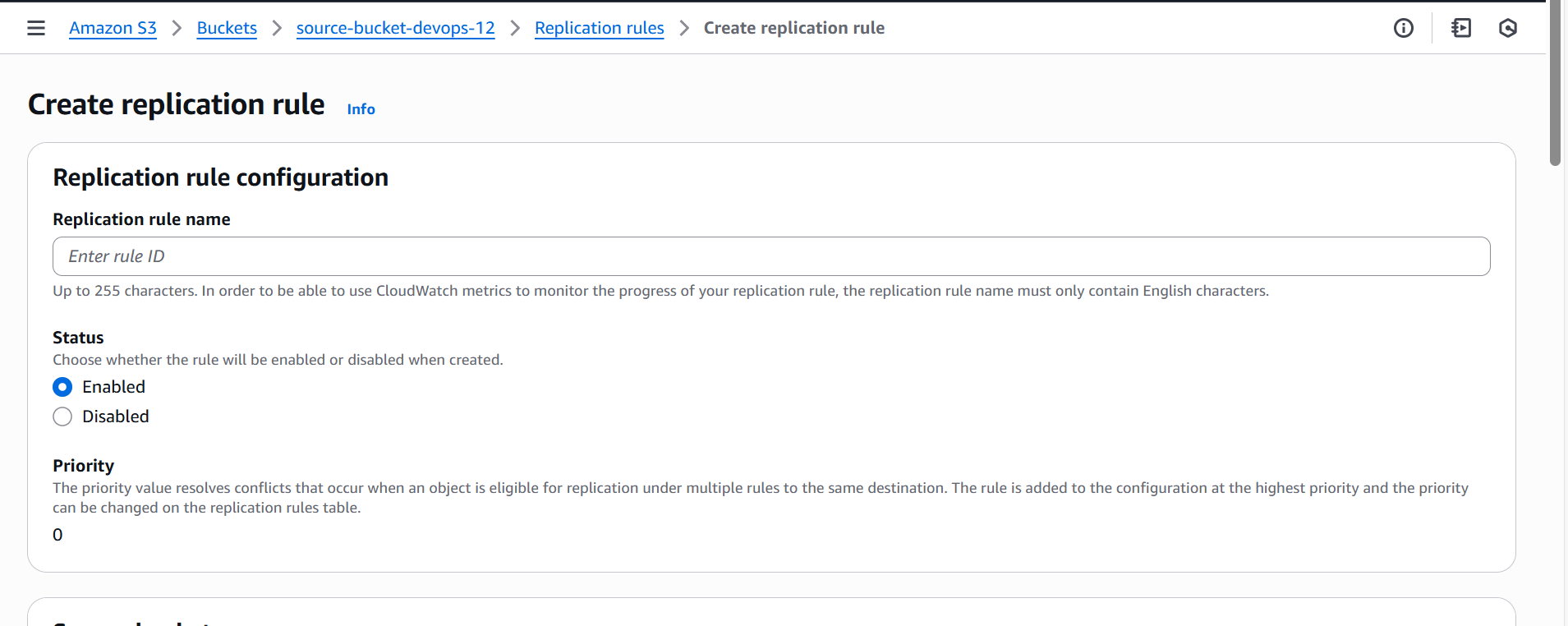
- Configure Additional Settings:

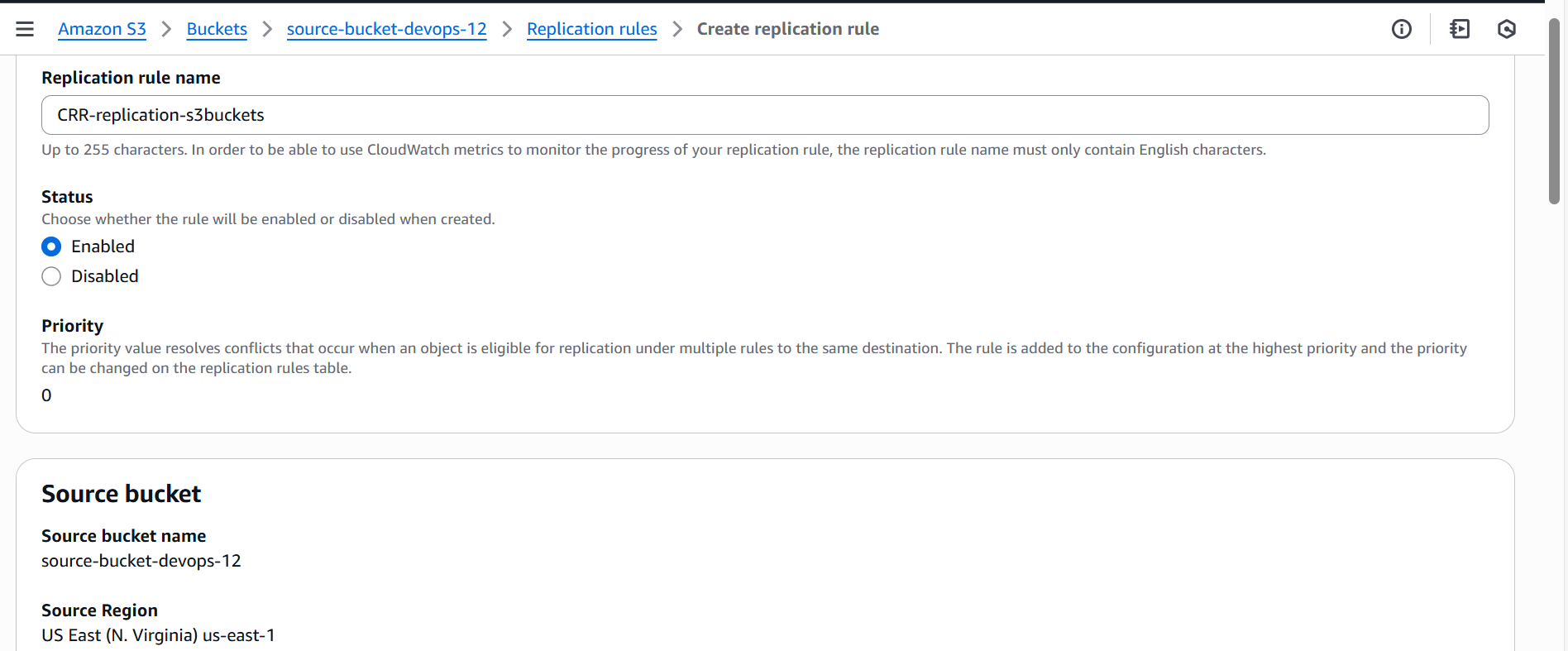
- Sync Existing Objects (optional)

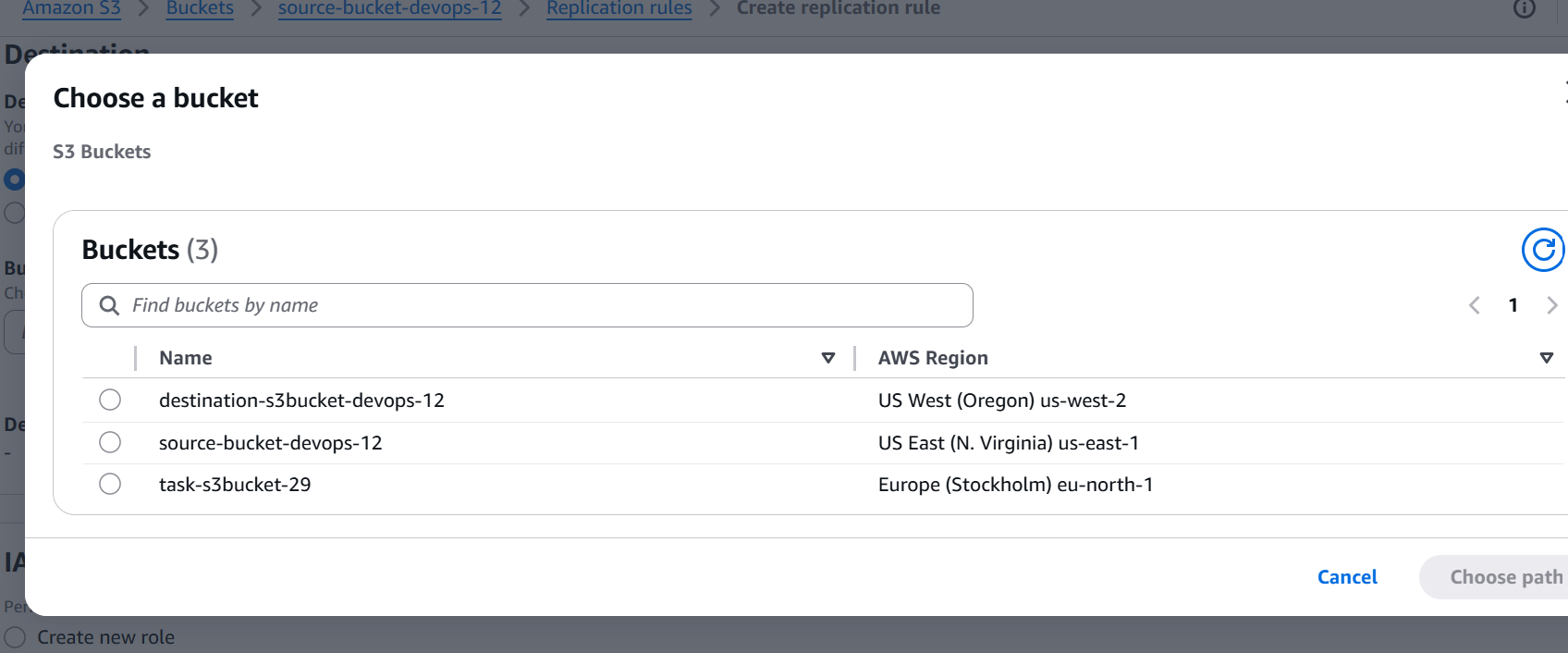
- Enable Encryption (optional)

- Replication Time Control (optional)

- Click Save Rule.





Create an IAM Role for Replication

- Open the IAM Console.

- Navigate to Roles → Create Role.

- Choose AWS Service → Select S3.

- Attach the following IAM policy:

{

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

"Statement": [

{

"Effect": "Allow",

"Action": ["s3:GetObjectVersion", "s3:GetObjectVersionAcl"],

"Resource": "arn:aws:s3:::source-bucket/\*"

},

{

"Effect": "Allow",

"Action": ["s3:ReplicateObject", "s3:ReplicateDelete"],

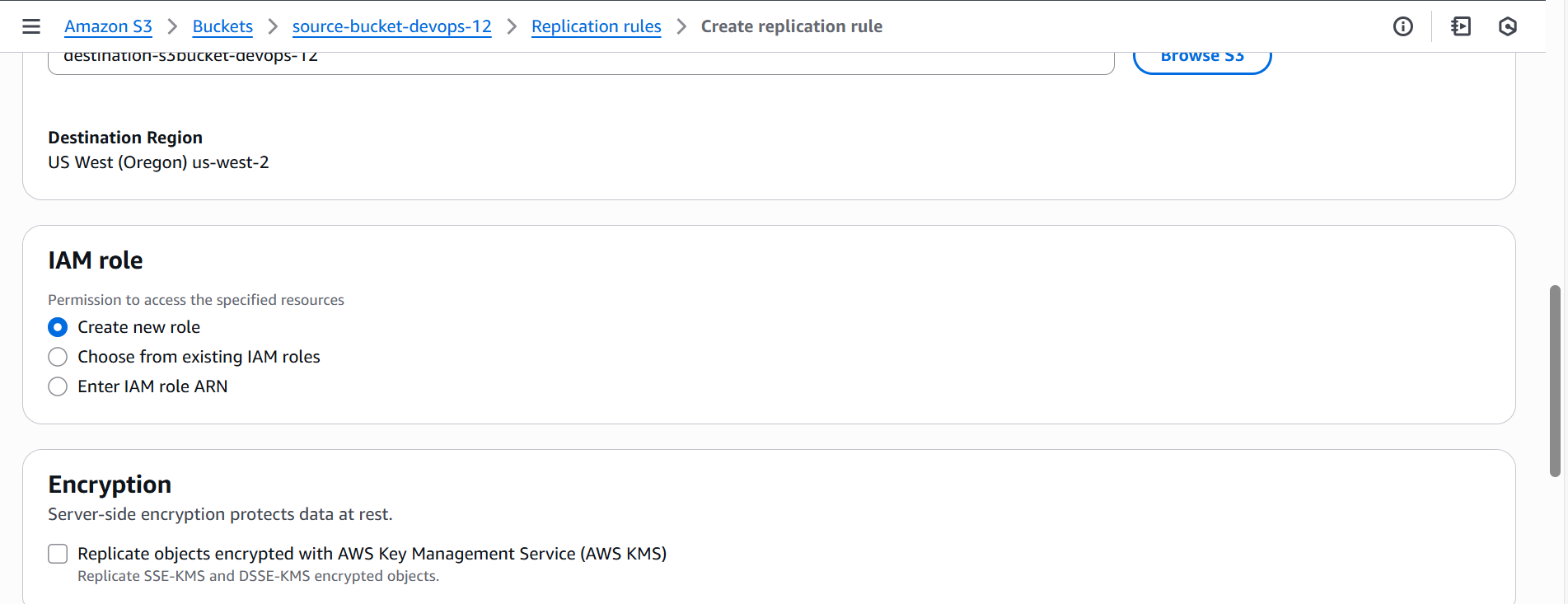
"Resource": "arn:aws:s3:::destination-bucket/\*"

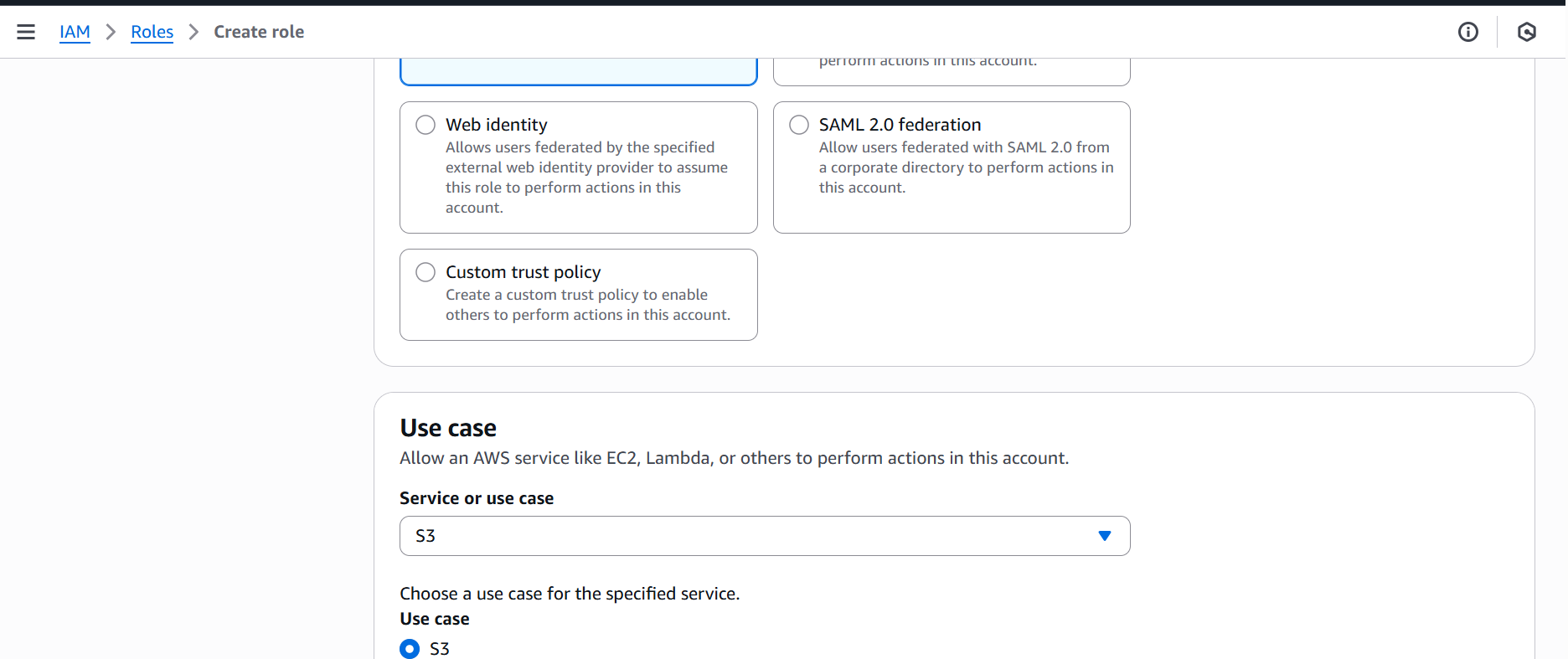
}

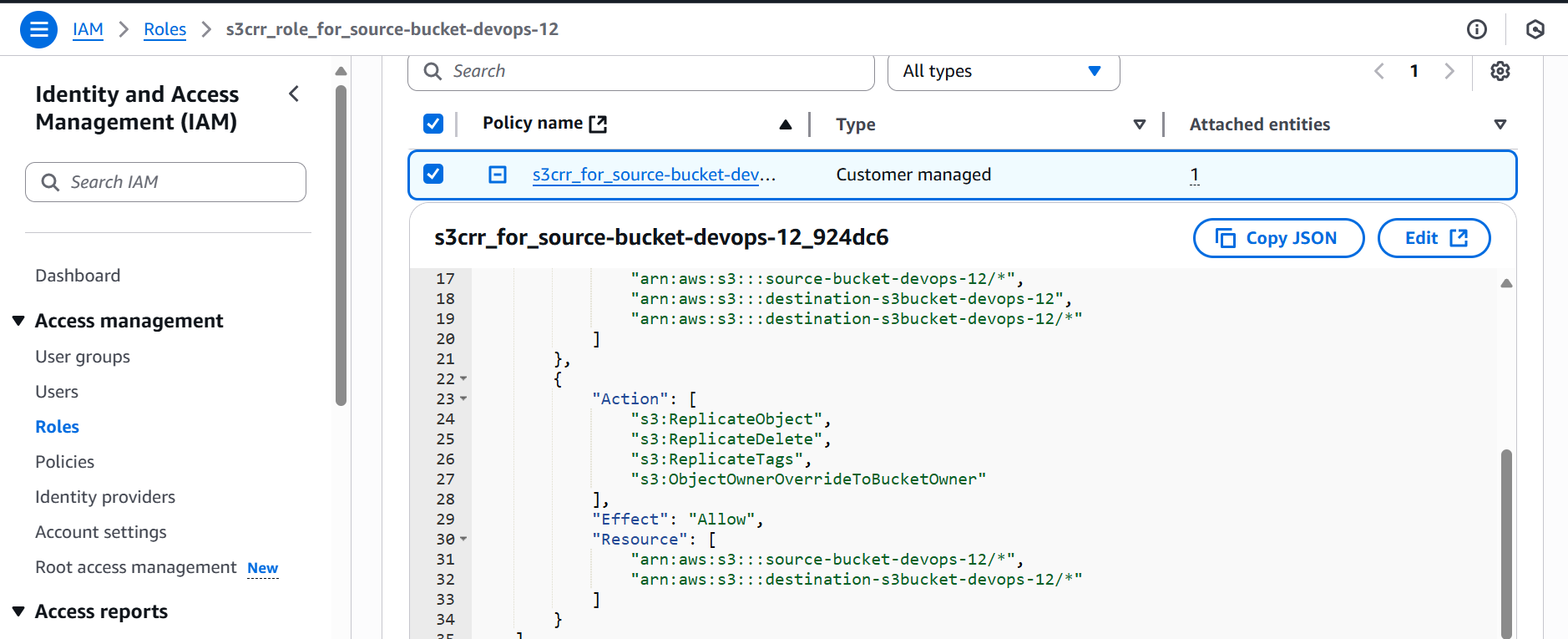
]

}

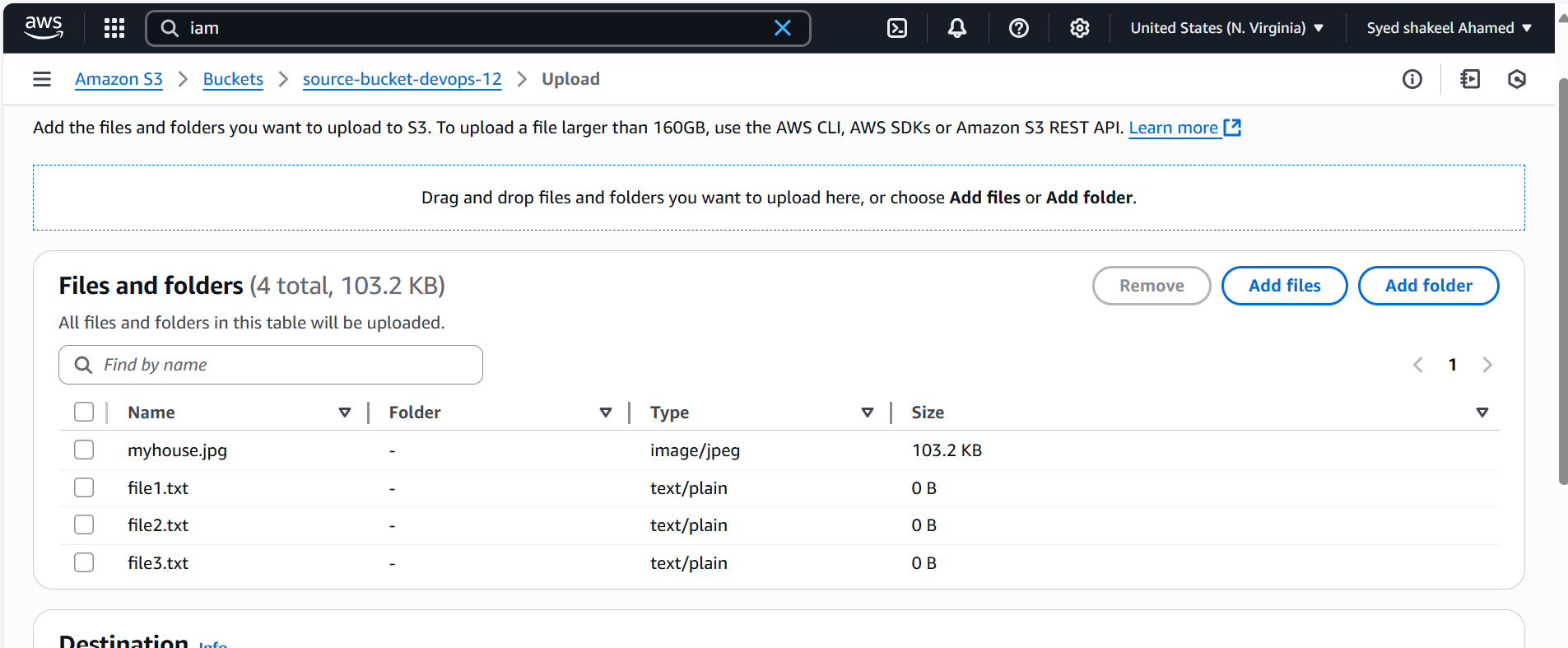
- Click Create Role and Note the Role ARN.

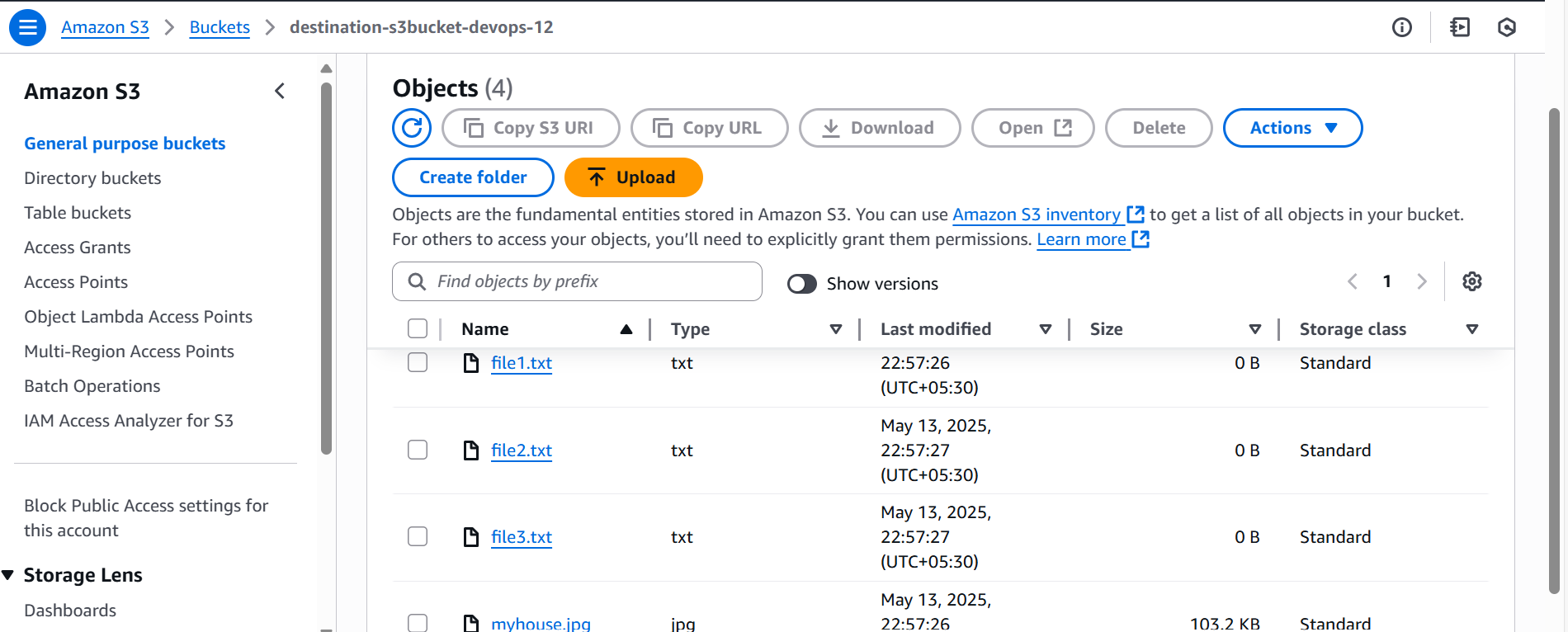


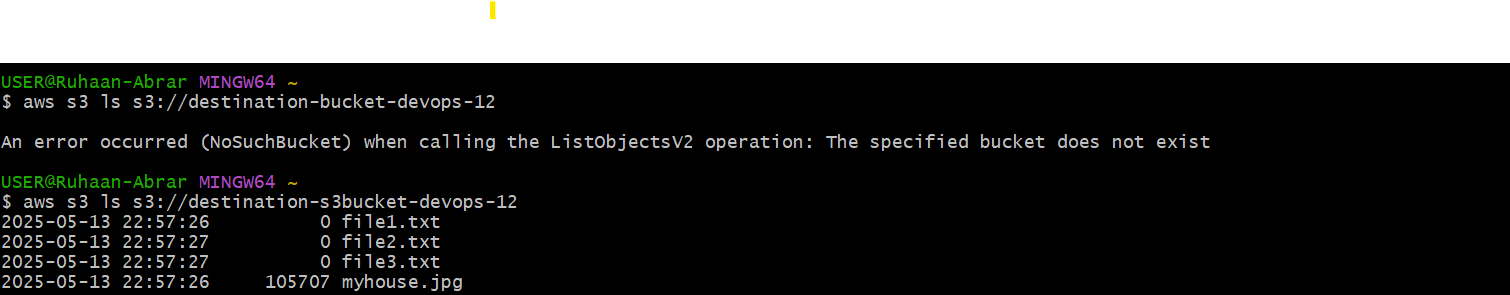












**4) Configure bucket policy, only Admin user can see the objects of s3 bucket.**

Restrict S3 Access to Admin User Only

Prerequisites

- An IAM user with the AdministratorAccess policy.

- A private S3 bucket (e.g., secure-bucket).

- A custom bucket policy restricting access to only the Admin user.

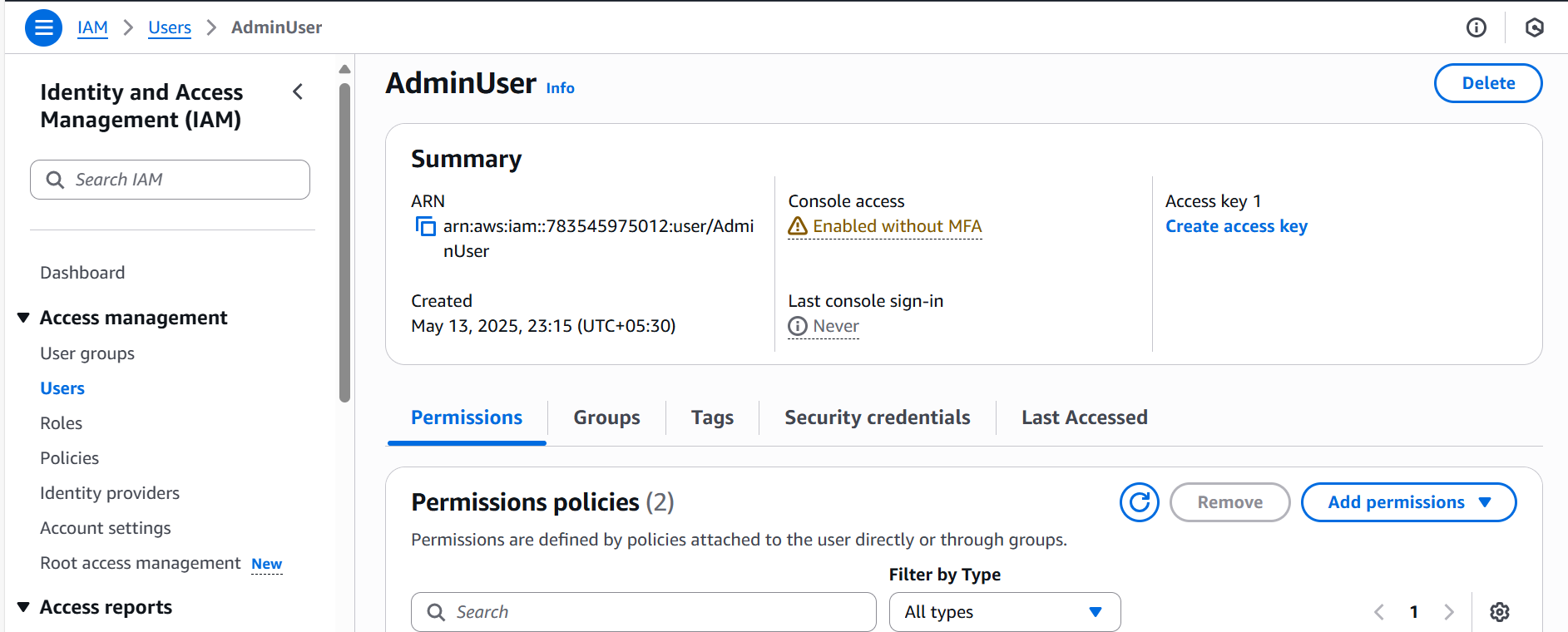
Identify Admin User

- Open the AWS IAM Console.

- Navigate to Users.

- Select the Admin User (e.g., AdminUser).

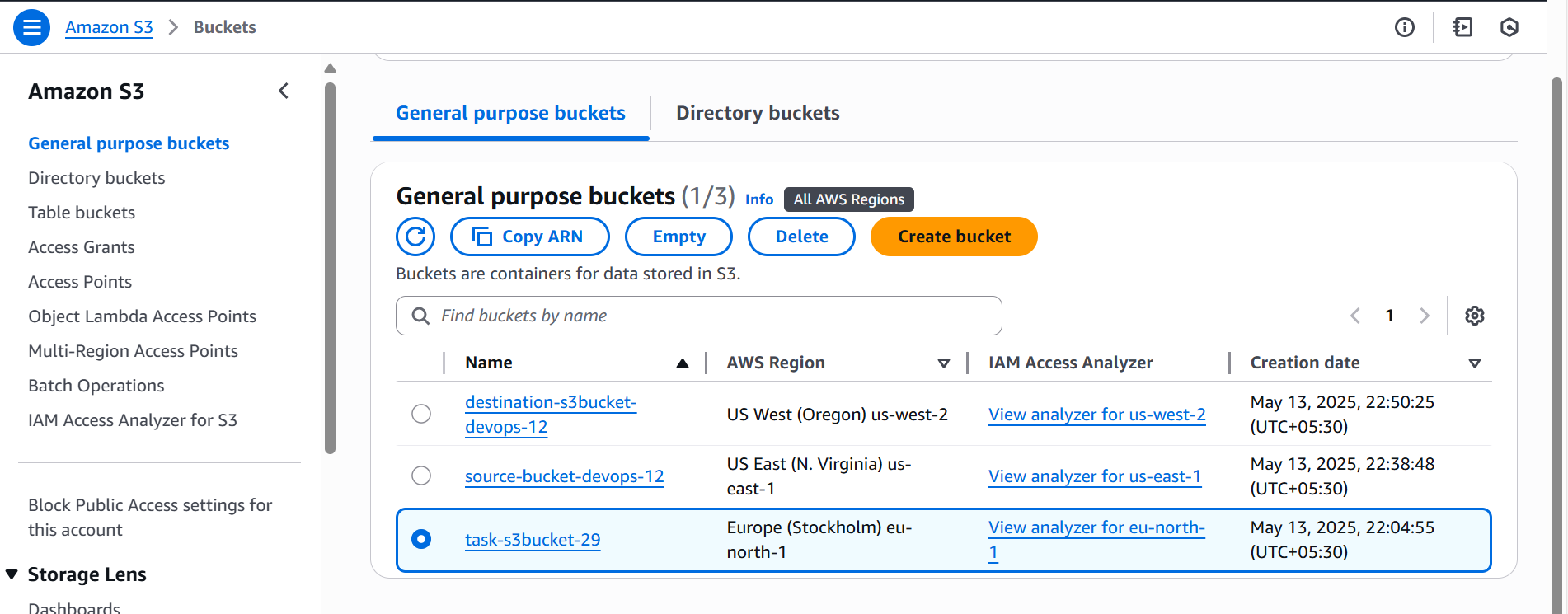
- Copy the IAM User ARN (e.g., arn:aws:iam::783545975012:user/AdminUser).

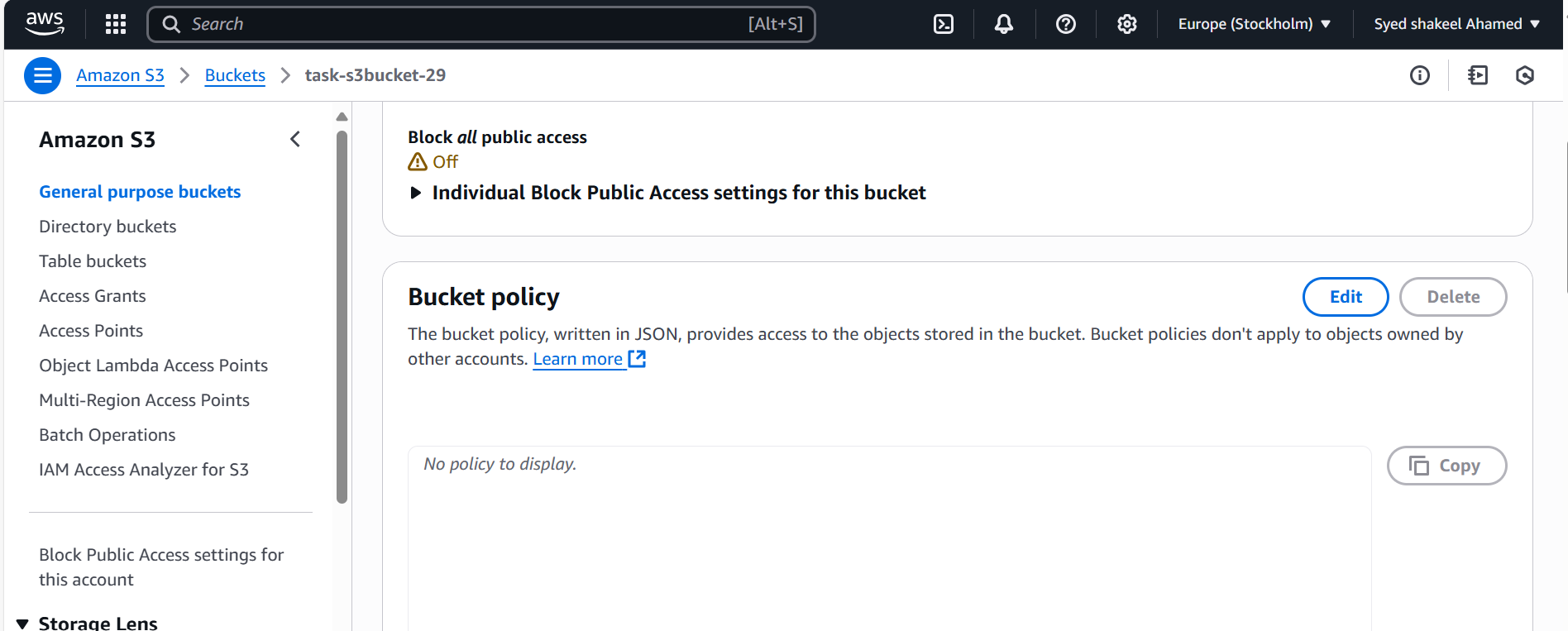


Configure S3 Bucket Policy

- Open the AWS S3 Console.

- Go to Permissions → Bucket Policy





* Add the following policy:

{

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

"Statement": [

{ "Effect": "Deny",

"Principal": "*",*

*"Action": "s3:GetObject",*

*"Resource": "arn:aws:s3:::secure-bucket/*", "Condition":

{

"StringNotEquals": { "aws:username": "AdminUser"

}

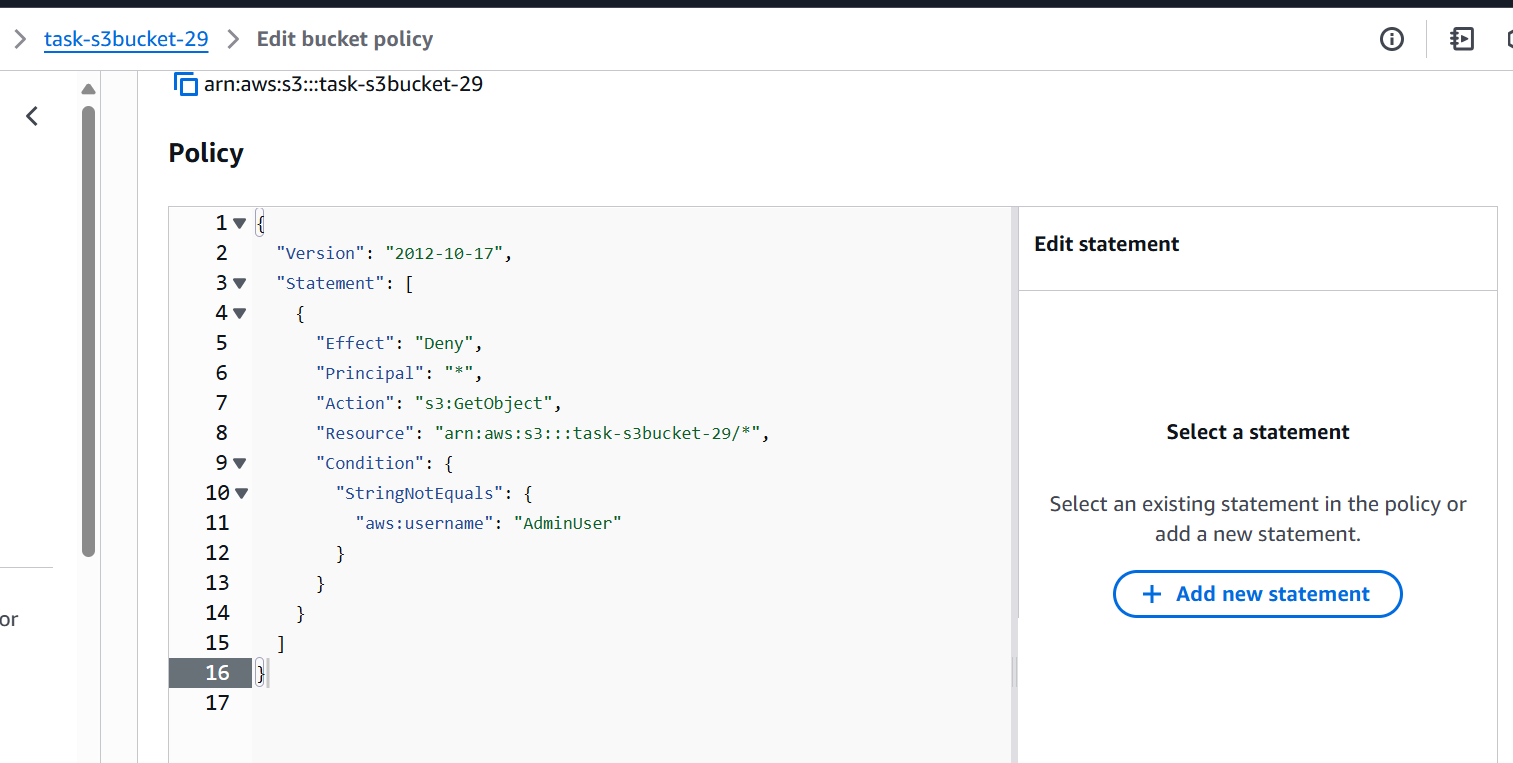
}

}

]

}

Click Save Changes.



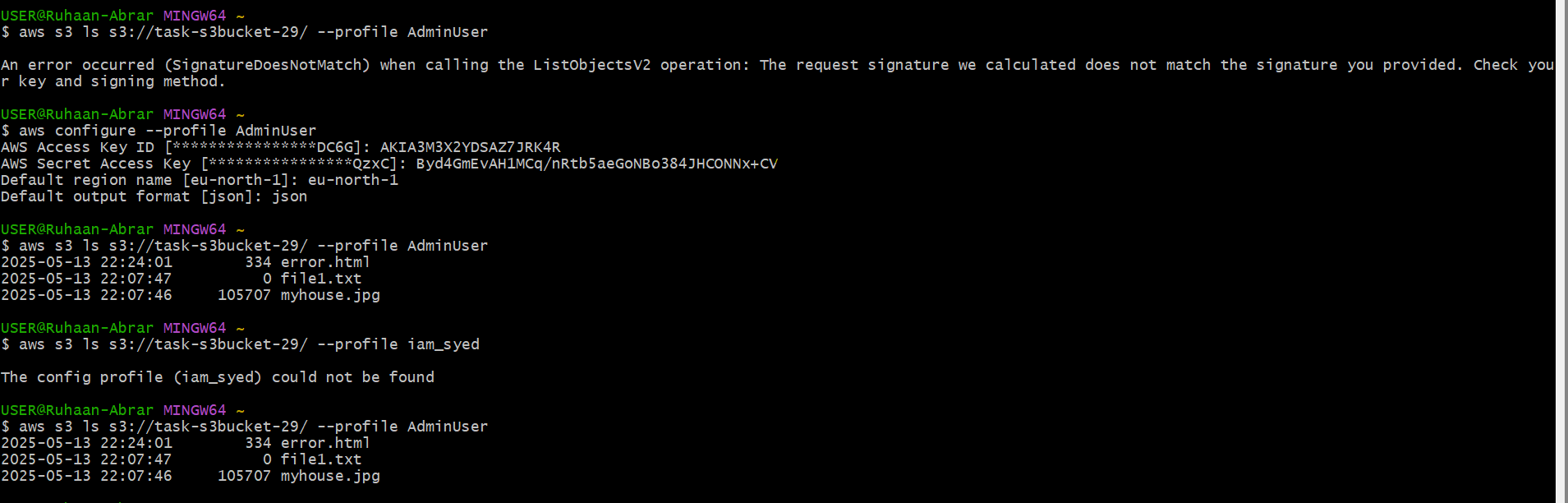
Verify Access

Test with Admin User

aws s3 ls s3://secure-bucket/ --profile AdminUser

We will get the output

And also checked with differe IAM user but got the access-denied error



**5) Setup lifecycle policies to automatically transition or delete objects based on specific criteria**

Setting Up S3 Lifecycle Policies

Prerequisites

- An S3 bucket with objects stored.

Define Lifecycle Requirements

Before setting up lifecycle rules, determine:

- Transition: Move objects to lower-cost storage classes (e.g., Glacier, Intelligent-Tiering).

- Expiration: Automatically delete objects after a specific period.

- Noncurrent Version Expiration: Remove older versions when using versioning.

Create a Lifecycle Rule

Using AWS Console

- Open the AWS S3 Console.

- Navigate to the Bucket where you want lifecycle policies.

- Go to Management → Lifecycle Rules.

- Click Create Lifecycle Rule.

- Enter Rule Name (e.g., ArchiveOldObjects).

- Choose Scope:

- Apply to all objects OR specific prefixes (logs/, data/).

- Configure Transition (Move objects to lower-cost storage):

- Move objects to Standard-IA after 30 days.

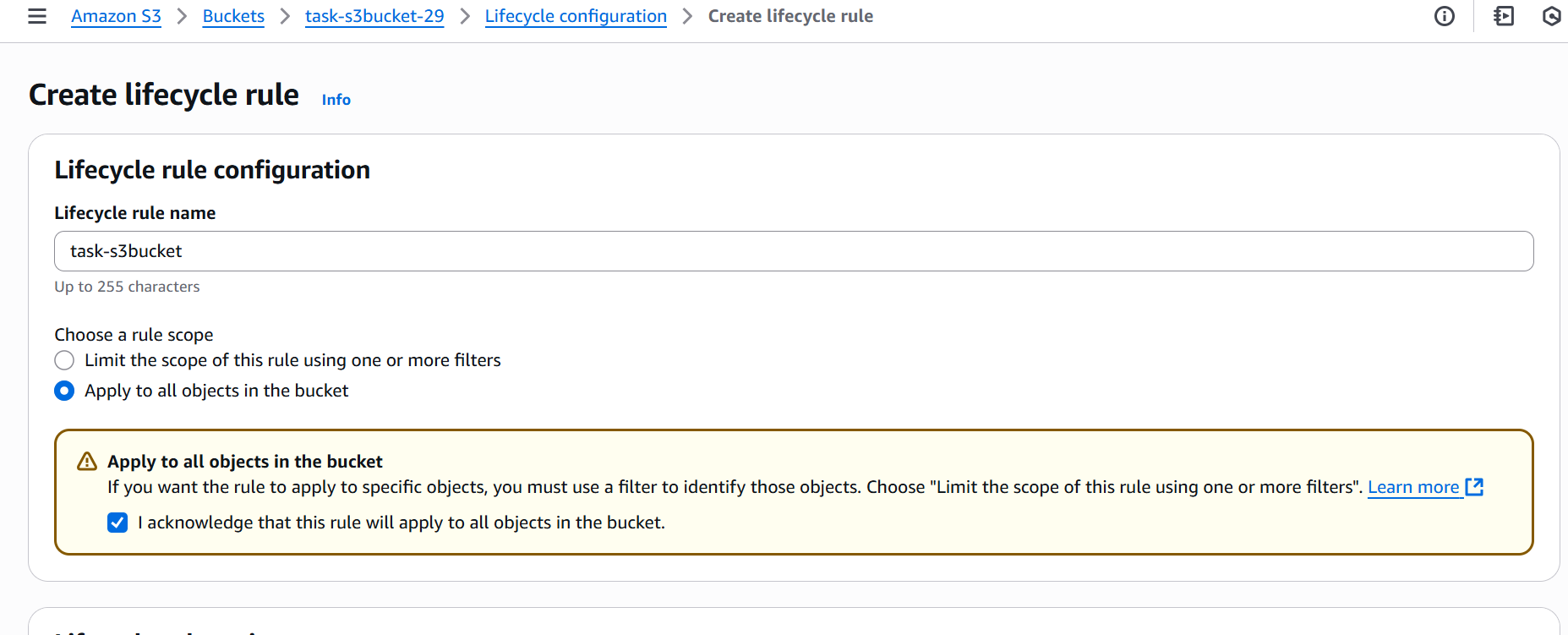
- Move objects to Glacier after 90 days.

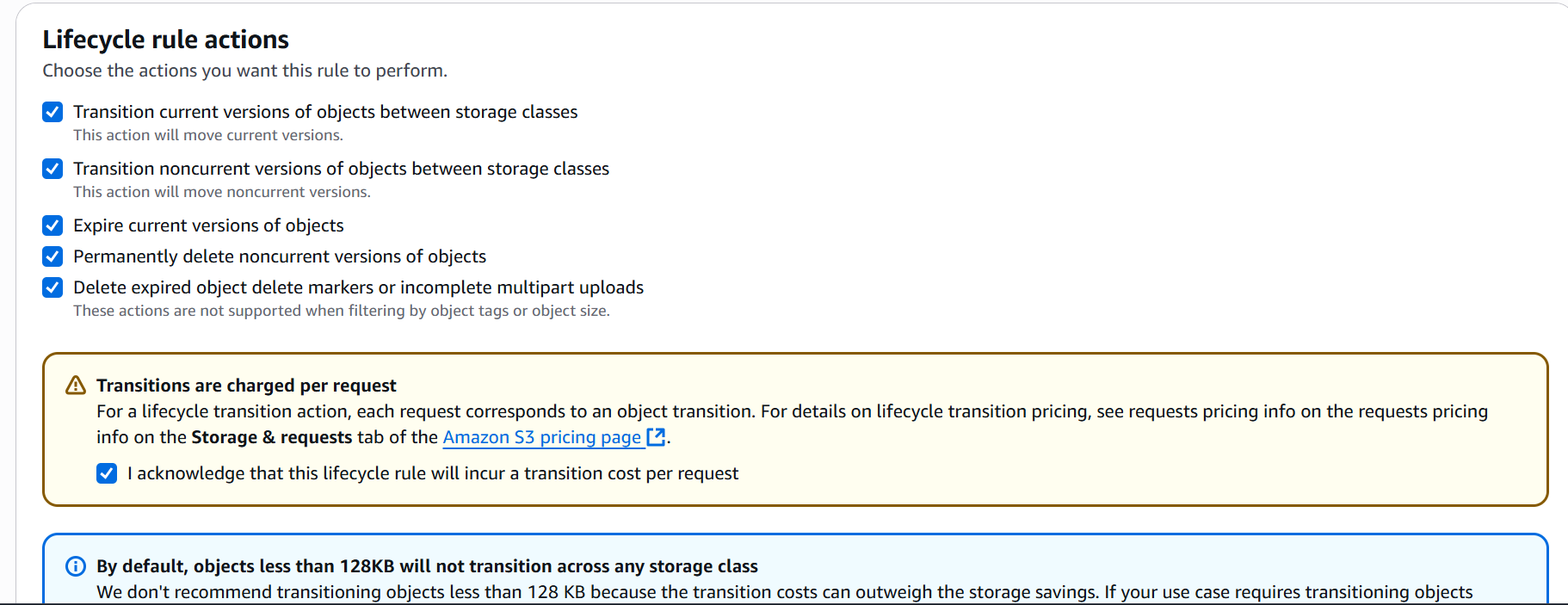
- Configure Expiration:

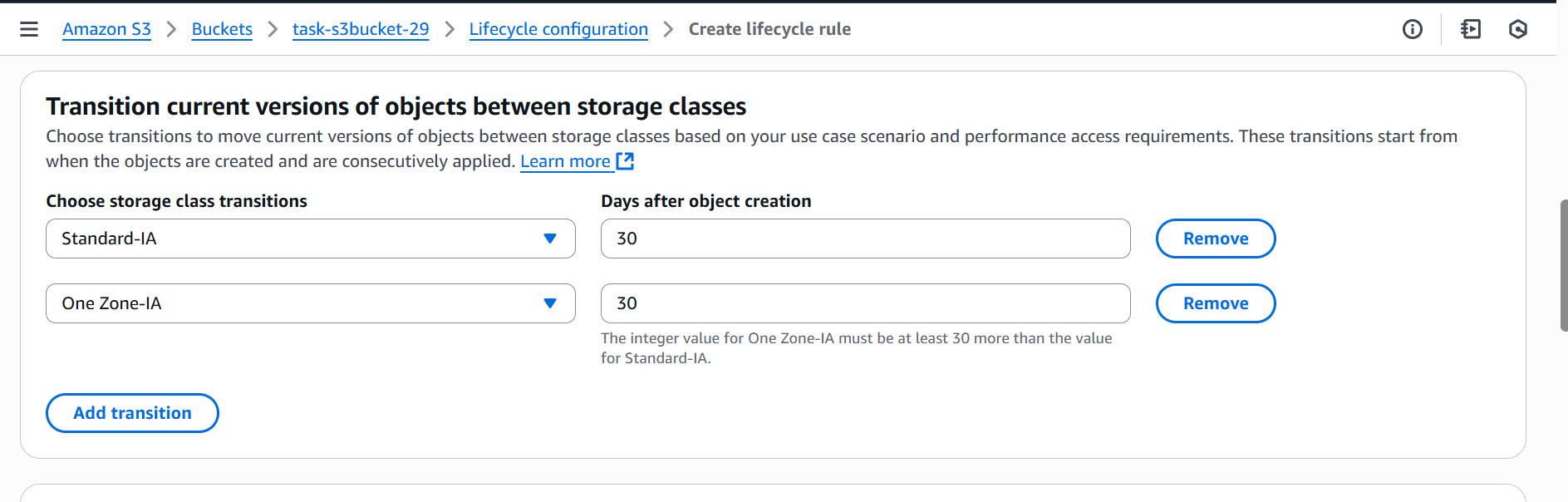
- Delete objects after 365 days.

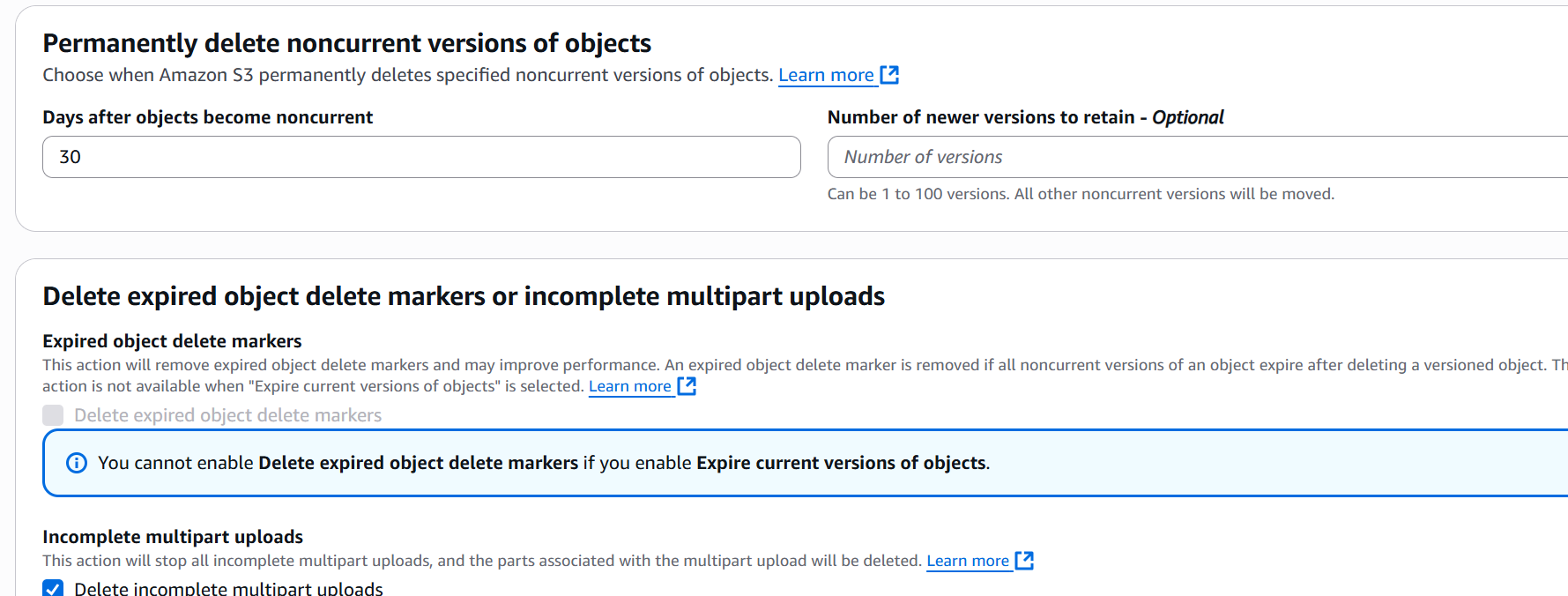
- Click Save Changes.

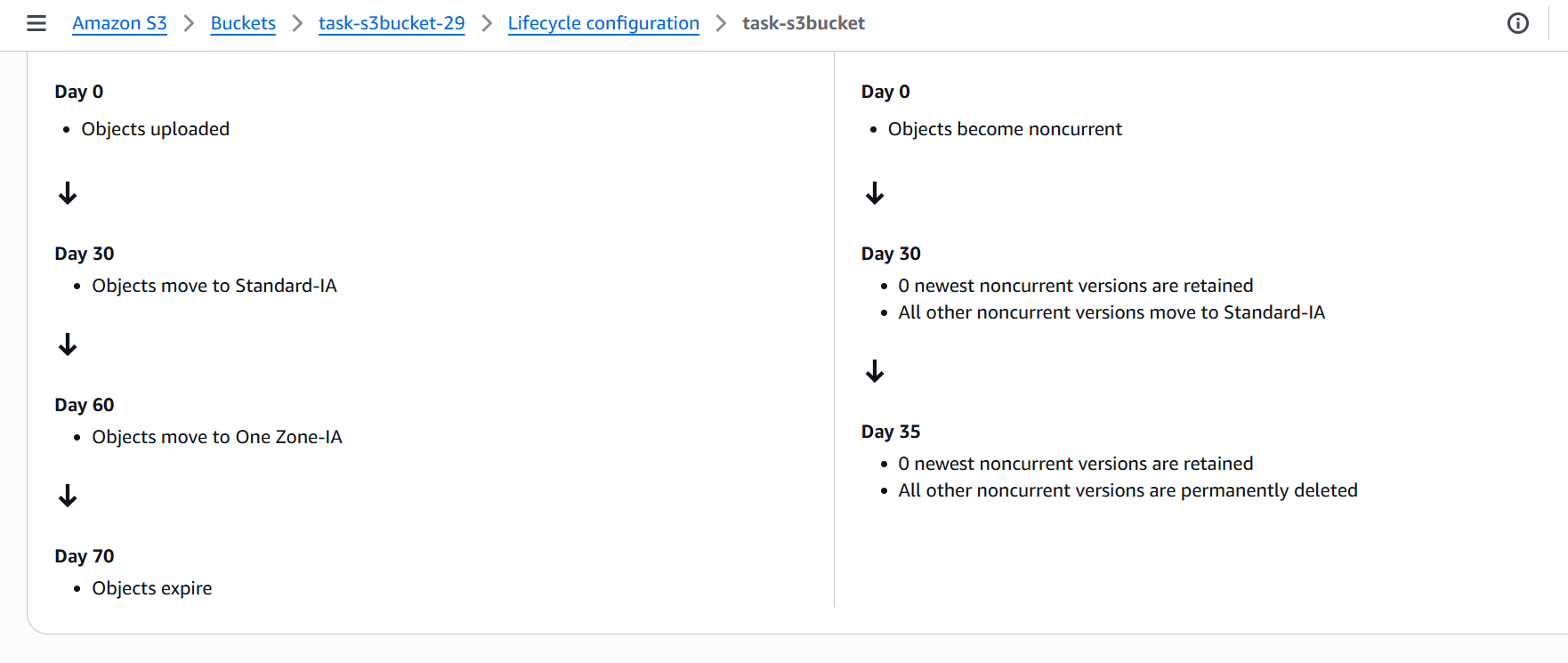












**6) Push some objects in s3 using AWS CLI**

Uploading Objects to S3 Using AWS CLI

Prerequisites

- AWS CLI installed (aws --version to verify).

- IAM user with necessary S3 write permissions (s3:PutObject).

- S3 bucket created (e.g., task-s3bucket-29).

- AWS CLI profile properly configured (aws configure).

Upload a Single Object To upload a single file (example.txt) to an S3 bucket:

aws s3 cp /local/path/example.txt s3://task-s3bucket-29/

Upload Multiple Files Uploading all files in a directory

aws s3 cp /local/path/ s3://task-s3bucket-29/ --recursive

This ensures all files in the local directory are pushed.

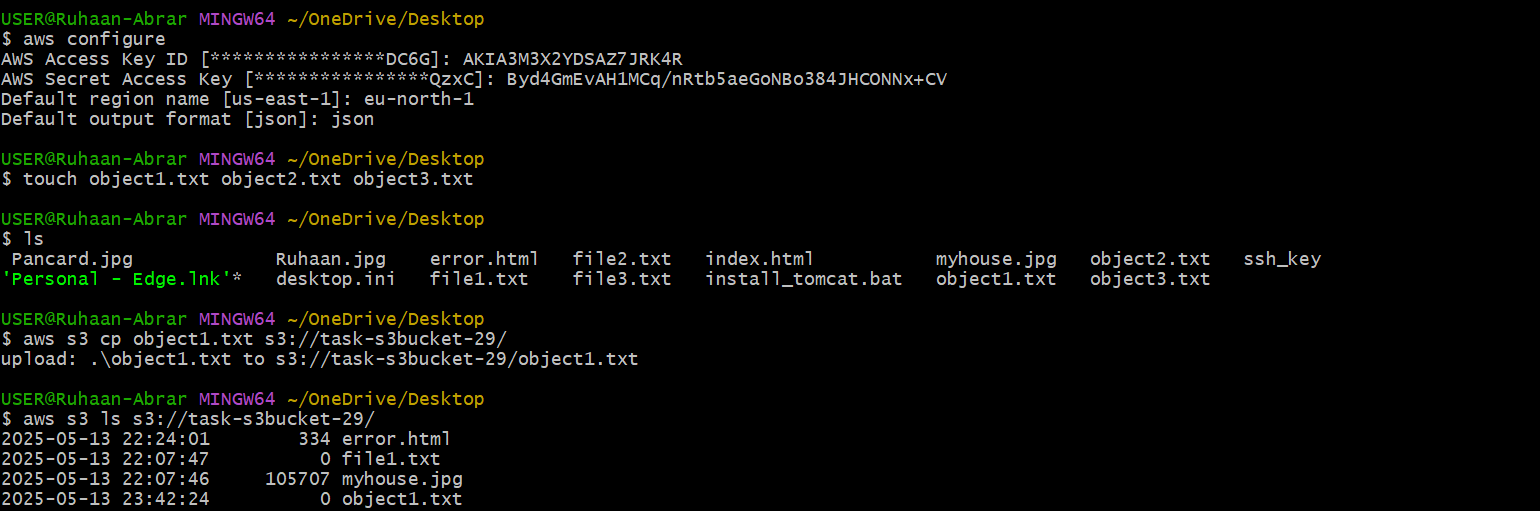
Uploading specific file types To upload only .txt files:

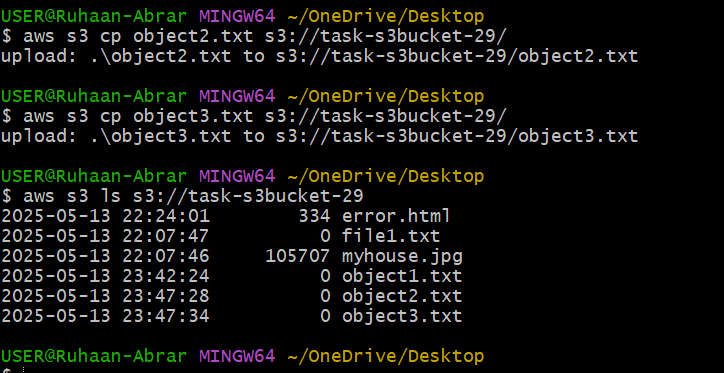
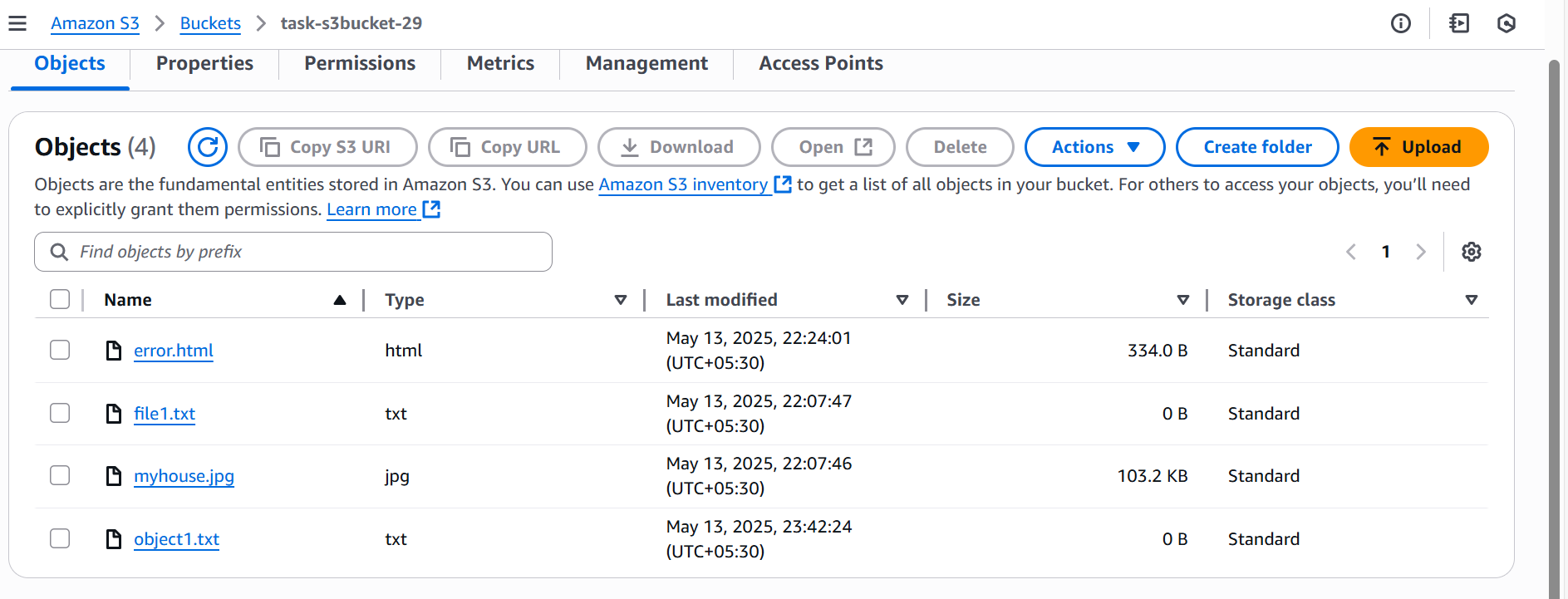
aws s3 cp /local/path/ s3://task-s3bucket-29/ --recursive --exclude "*" --include "*.txt"

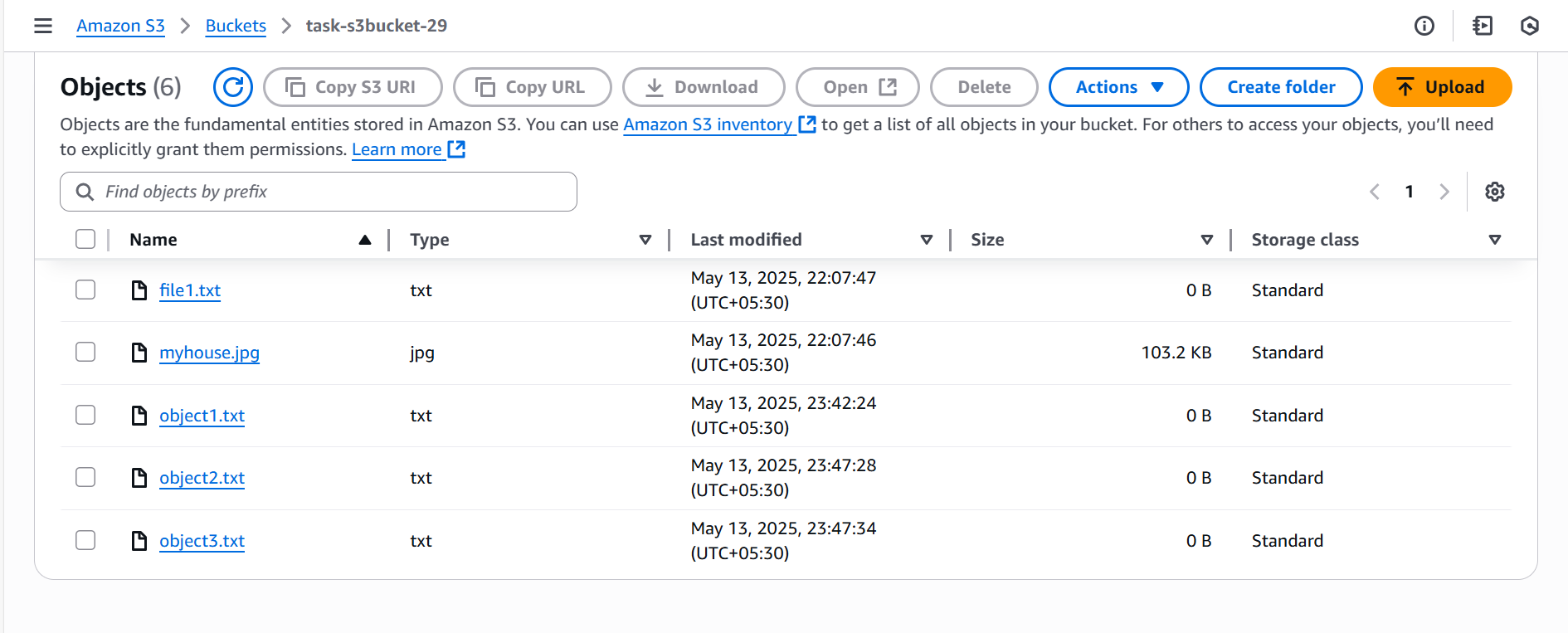
Verify Uploaded Objects

List all uploaded files:

aws s3 ls s3://task-s3bucket-29/







**7) Write a bash script to create s3 bucket.**

Configure AWS CLI using Access key and secret key and proper region

Create one script file using

vi create\_s3bucket.bash

enter the bash script in the file

#!/bin/bash

BUCKET\_NAME="s3bucket-bashscript"

REGION="us-east-1"

aws s3 mb s3://$BUCKET\_NAME --region $REGION

aws s3 ls | grep $BUCKET\_NAME

echo "Bucket '$BUCKET\_NAME' created successfully!"

save the script file

Now change the modification to execute the file

chmod +x create\_s3bucket.bash

Run the file

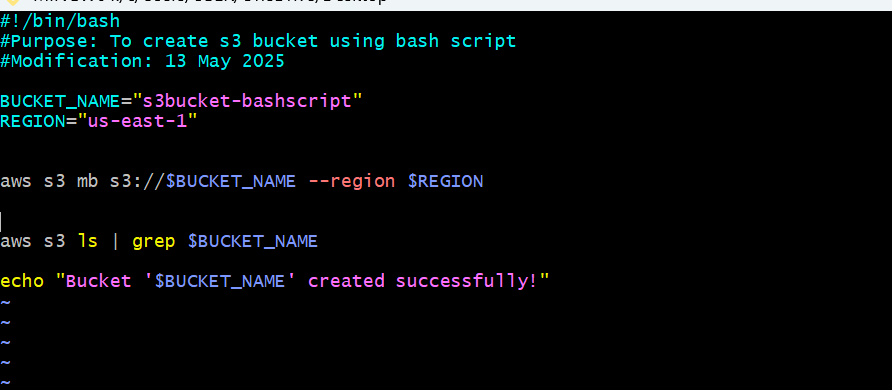
./create\_s3bucket.bash

Once the script executed successfully

check the Amazon S3 bucket list

Successfully the bucket will be created with region provided





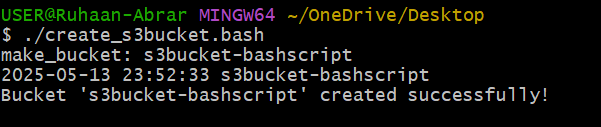
**1. aws s3 mb s3://$Bucket\_NAME --region $REGION**

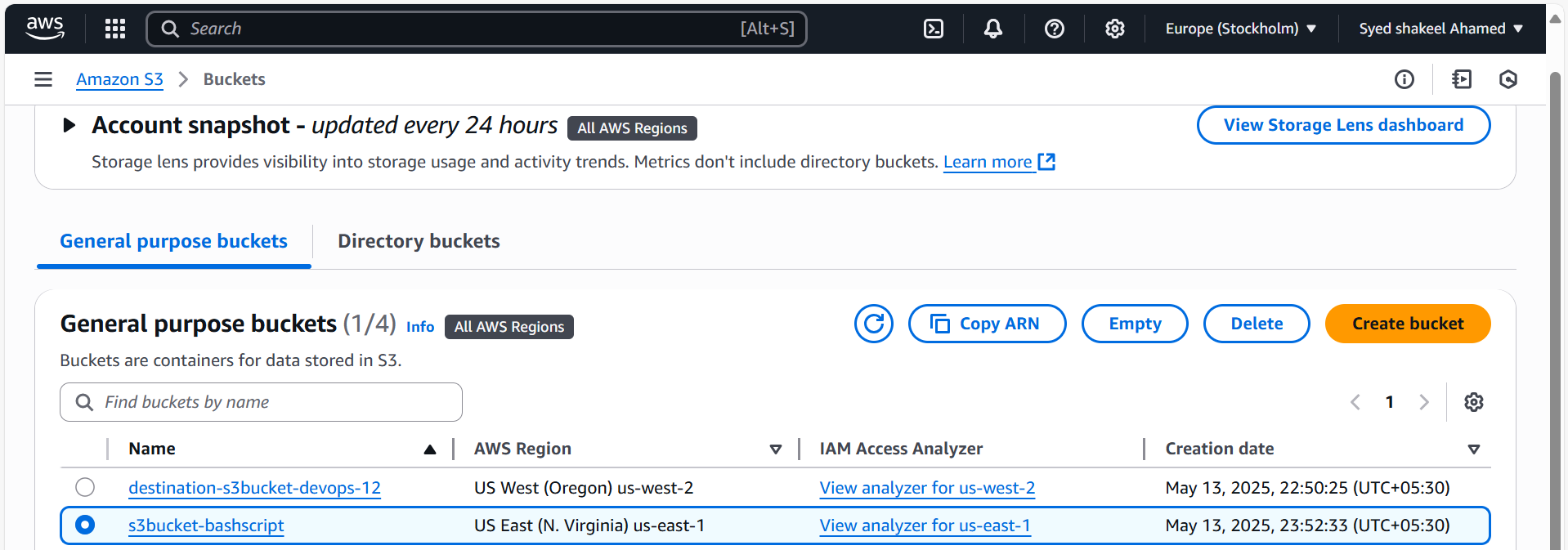
* aws s3 → Calls the **AWS CLI** and specifies interaction with **Amazon S3**.
* mb → **Make Bucket** command, used to create a new S3 bucket.
* s3://$Bucket\_NAME → Defines the **S3 bucket name** dynamically (using $Bucket\_NAME as a variable).
* --region $REGION → Specifies the **AWS region** where the bucket will be created (e.g., us-east-1).

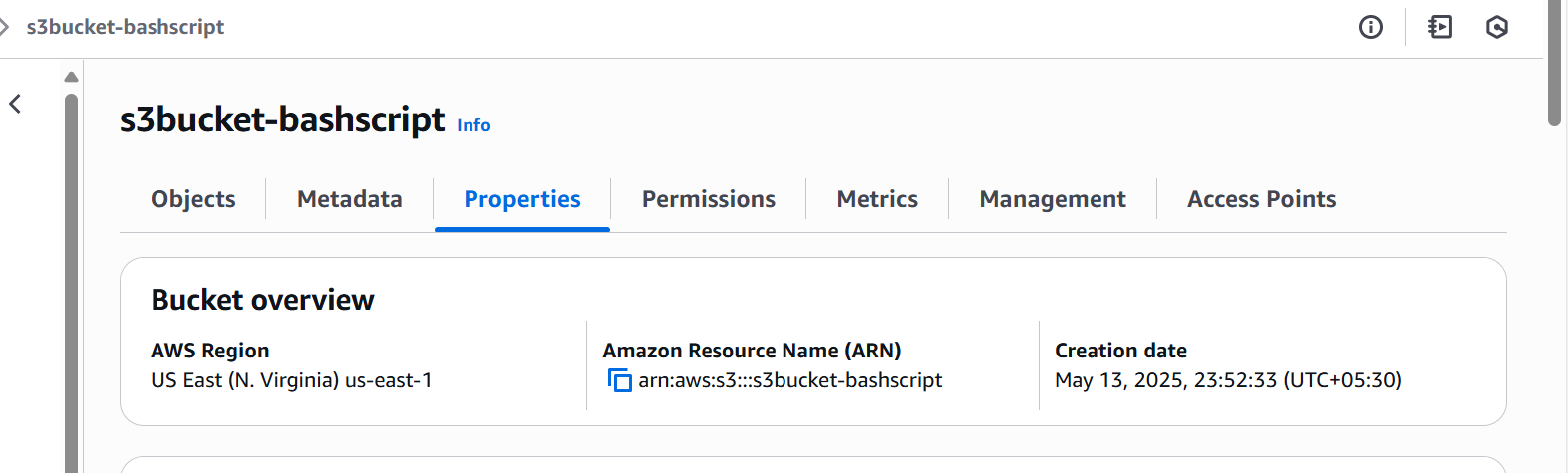
**2. aws s3 ls | grep $Bucket\_NAME**

* aws s3 ls → Lists **all buckets** in your AWS account.
* | grep $Bucket\_NAME → Filters the output to show **only the bucket** matching $Bucket\_NAME.









**8) Upload one 1 gb of file to s3 using cli.**

Open Command Prompt in Desktop

Create a testfile of 1GB using below command

fsutil file createnew testfile-1gb.bin 1073741824

fsutil file createnew → Uses the fsutil utility to create a new file.

- testfile-1gb.bin → The filename of the file to be created.

- 1073741824 → The file size in bytes (1GB = 1024 × 1024 × 1024 = 1,073,741,824 bytes).

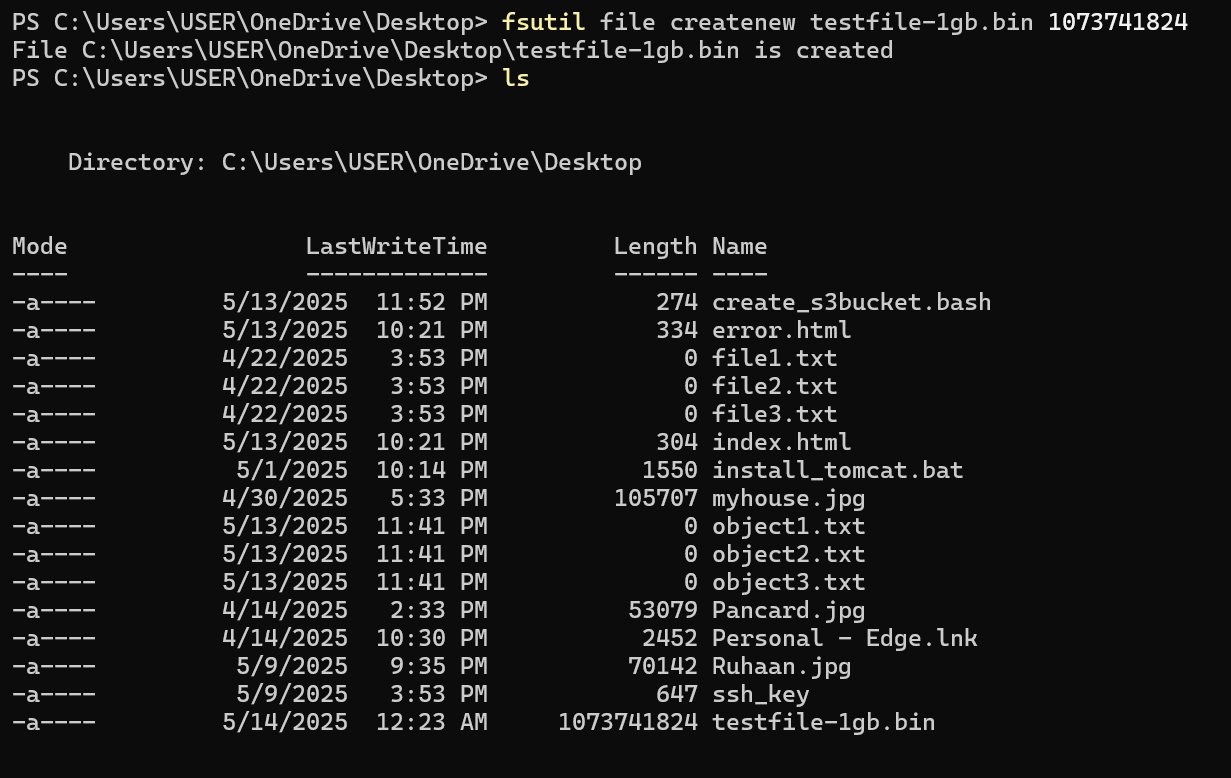
Effect of Running This Command

- Creates a binary file (testfile-1gb.bin) of exactly 1GB.

- The file is empty (filled with zeroes).

- Useful for testing large file uploads to S3 or other storage systems.

Once the file is created



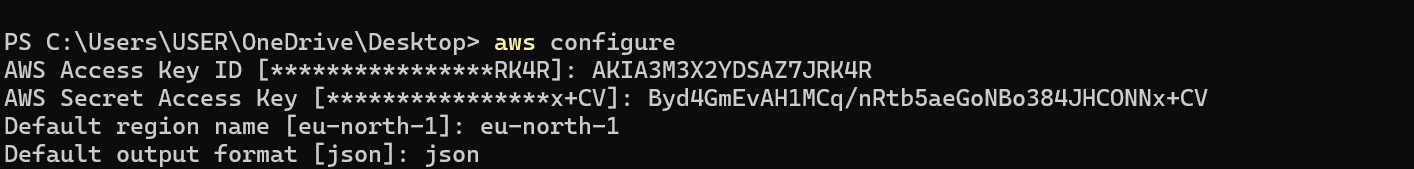
Configure AWS CLI on the terminal using

Access key

Secret Key

Region

Output format



Once the configuration is done

Try uploading the file to s3 bucket using below command

aws s3 cp testfile-1gb.bin s3://task-s3bucket-29/

1. aws s3

- aws → Calls the AWS CLI.

- s3 → Specifies the Amazon S3 service.

2. cp (Copy)

- cp stands for copy, meaning it transfers files from local storage to S3.

3. testfile-1gb.bin

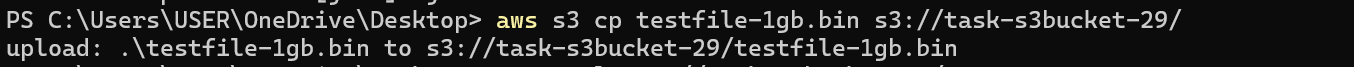
- This is the source file on your local machine (size: 1GB).

- Needs to exist before running the command.

4. s3://task-s3bucket-29/

- s3:// → Refers to an S3 bucket.

- task-s3bucket-29/ → Specifies the destination bucket in S3.



Verification

After the upload, list objects in the bucket:

aws s3 ls s3://task-s3bucket-29/

1. aws s3

* aws → Calls the AWS CLI tool.
* s3 → Specifies that you're interacting with Amazon S3.

1. ls (List Objects)

* ls stands for list, which fetches a list of objects stored in an S3 bucket.

1. s3://task-s3bucket-29/

* s3:// → Defines the S3 storage path.
* task-s3bucket-29/ → The S3 bucket name where objects are stored.

