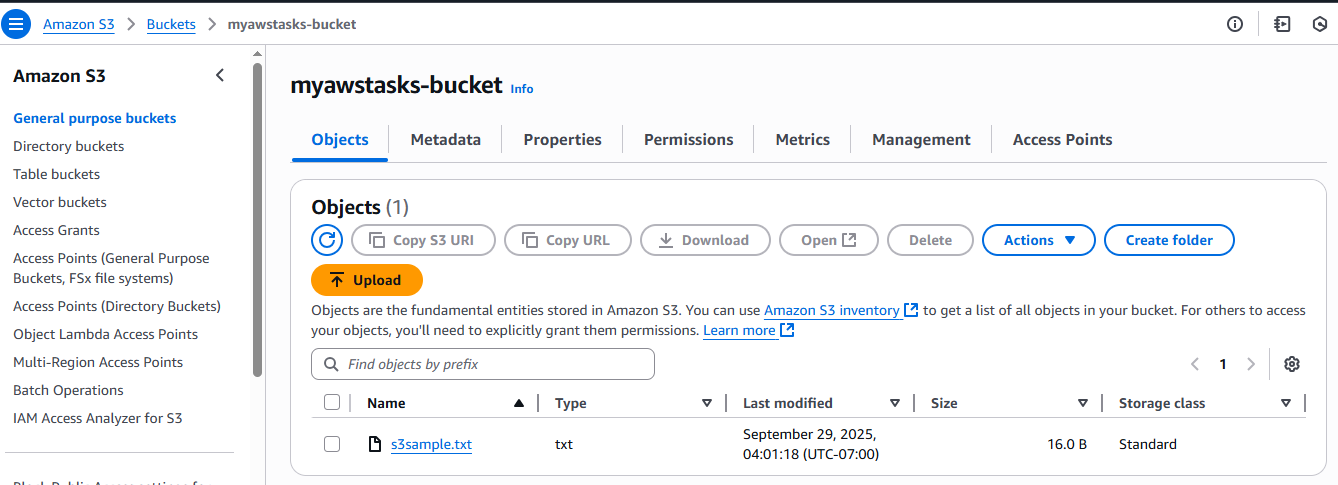
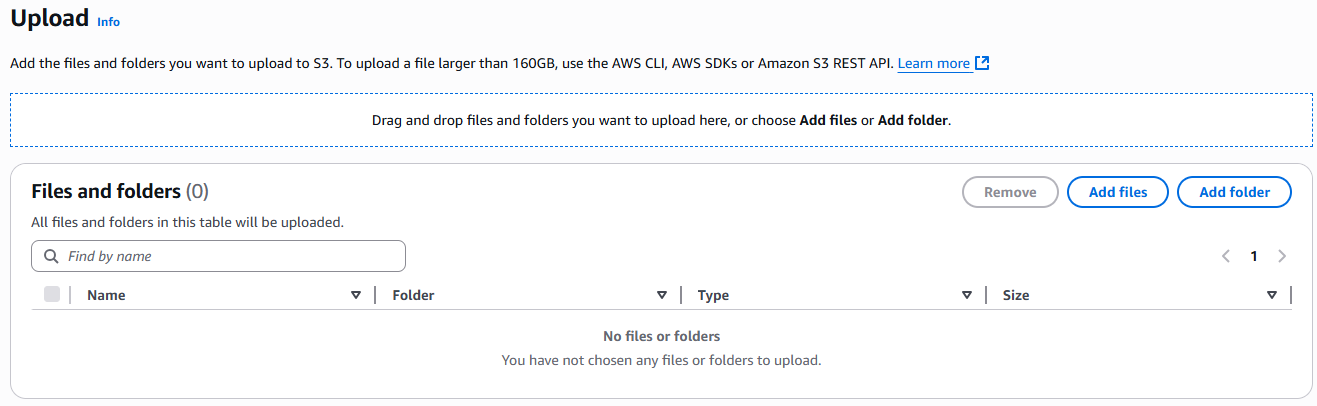
**S3 TASKS**

**1.Create an S3 bucket and upload some objects to S3.**

**Using AWS Console**

1. **Login** to the AWS Management Console.
2. Go to **S3** service.
3. Click **Create bucket**.
   * **Bucket name**: Must be unique globally (e.g., my-first-s3-bucket-demo-2025).
   * **Region**: Select a region (e.g., us-east-1).
   * Keep defaults unless you need versioning/encryption.
4. Click **Create bucket**.
5. Open the bucket → **Upload** → **Add files**.
6. Select files from your computer (e.g., test.txt, image.jpg) and click **Upload**.





**2.Deploy a static website in the S3 bucket.**

**1. Prepare Your Website Files**

Create a folder locally with at least:

* index.html (homepage)
* error.html (optional, error page)

Example index.html:

<!DOCTYPE html>

<html>

<head>

<title>My First S3 Website</title>

</head>

<body>

<h1>Hello from S3 Static Website!</h1>

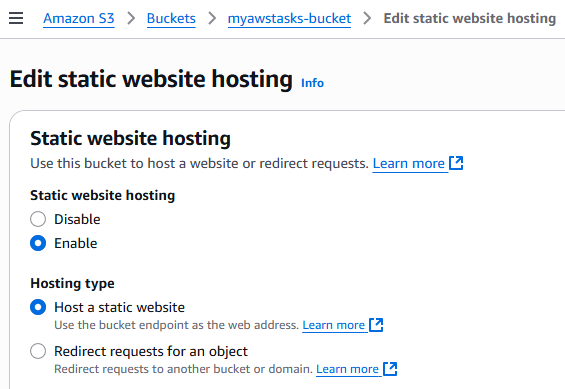
</body>

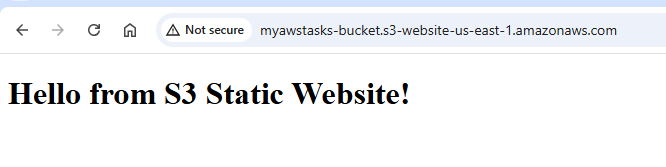
</html>

**2. Using AWS Console**

1. **Create an S3 bucket** (name must be unique, e.g. my-static-site-2025).
   * **Region**: Choose your preferred region.
   * **Uncheck** *Block all public access* → Confirm.
   * Enable bucket for public access (needed for website hosting).
2. Upload your files (index.html, error.html).
3. Go to **Properties** → Scroll to **Static website hosting**.
   * Enable it.
   * Select **Host a static website**.
   * Enter index.html for Index document.
   * Enter error.html for Error document.
   * Save.
4. Copy the **Endpoint URL** (something like:  
   http://my-static-site-2025.s3-website-us-east-1.amazonaws.com).

Now your site is live 🎉





**3.Enable cross-region replication on S3 buckets.**

# Configure S3 Cross-Region Replication in Console

### **Step 1: Enable Versioning on Both Buckets**

1. Go to the **S3 Console**.
2. Open your **source bucket** → **Properties** tab → **Bucket Versioning** → **Enable**.
3. Do the same for your **destination bucket**.

### **Step 2: Open Replication Settings on Source Bucket**

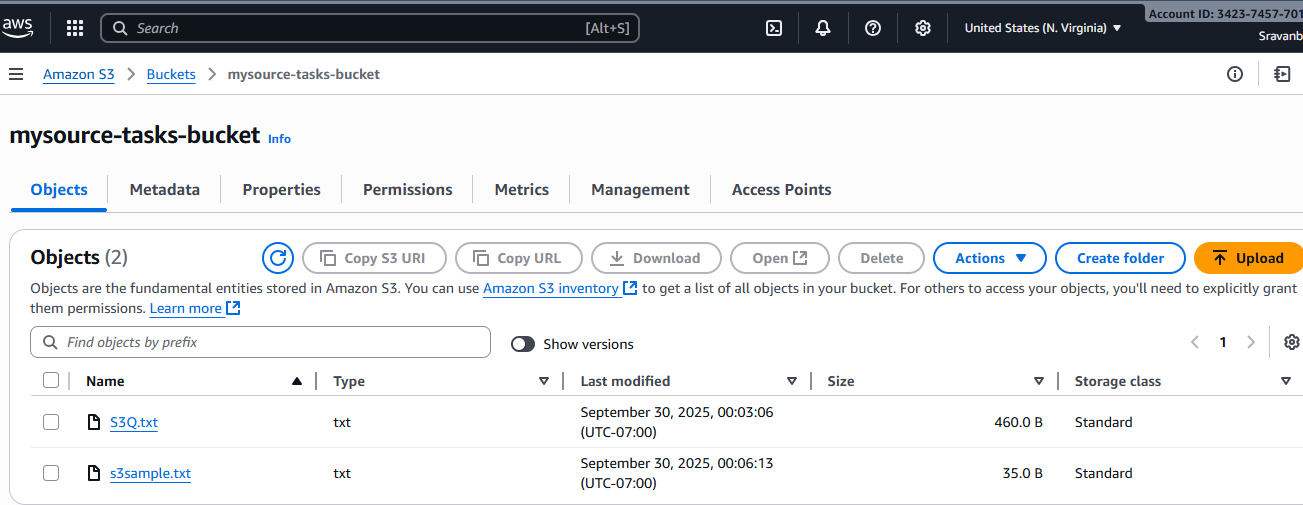
1. In the **source bucket**, go to the **Management** tab.
2. Scroll to **Replication rules** → click **Create replication rule**.

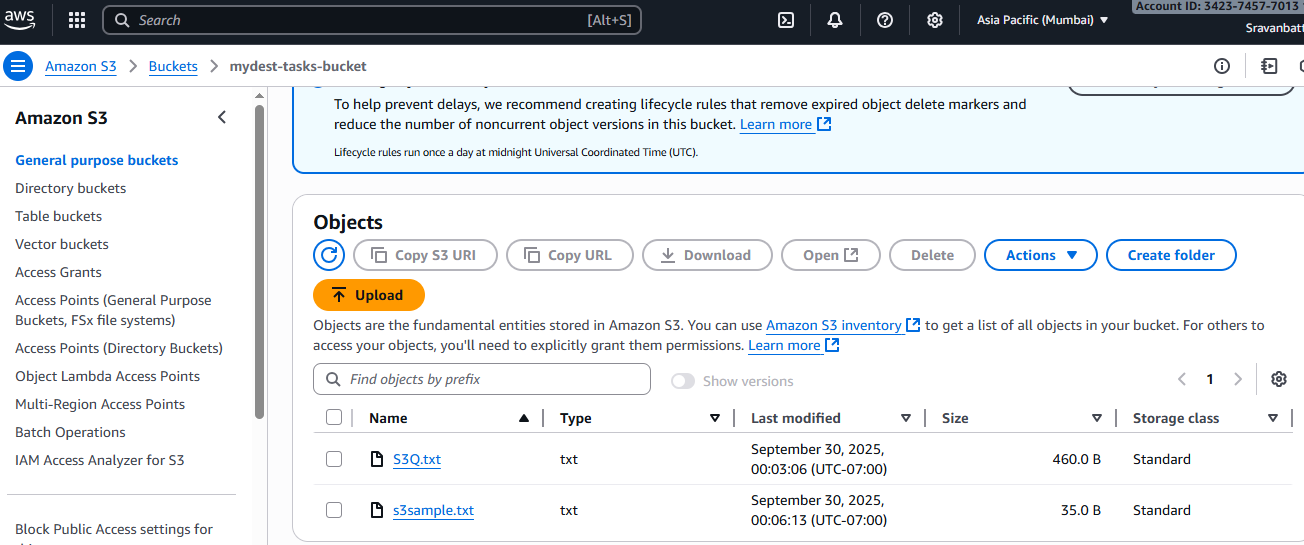
### **Step 3: Configure the Rule**

1. **Rule name** → e.g. CRR-Rule.
2. **Status** → Keep it enabled.
3. **Choose rule scope** → Apply to all objects (or you can set prefix/tags if needed).
4. **Destination**:
   * Select **Choose a bucket in this account**.
   * Enter your **destination bucket name**.
   * Make sure region is different (for cross-region).
   * Choose storage class (default: Standard).
5. **IAM role**:
   * Choose **Create IAM role**.
   * 
   * Select the role you created earlier → S3ReplicationRole.

### **Step 4: Review and Save**

1. Review the settings.
2. Click **Save**.





**4.Configure a bucket policy so only the Admin user can see the objects of the S3 bucket.**

**Step 1: Create Admin User**

1. Go to the **IAM console** → **Users** → **Add users**
2. Enter username → Admin
3. Select **Access type**:
   * ✅ AWS Management Console access (for login)
   * ✅ Programmatic access (if using CLI/SDK)
4. Set password (auto or custom).
5. Click **Next: Permissions**
6. Skip group for now → **Next → Create user**.

**Step 2: Attach Full Access Policy to Admin**

1. Open the created **Admin user** → **Permissions** → **Add permissions**
2. Attach **AmazonS3FullAccess** (or a custom policy you prefer).

Now Admin has the right permissions.

**Step 3: Restrict Bucket Policy to Only Admin**

Suppose bucket name is **my-private-bucket**.  
You need to know the **Admin user ARN** (found in Admin’s IAM user details → ARN). Example:

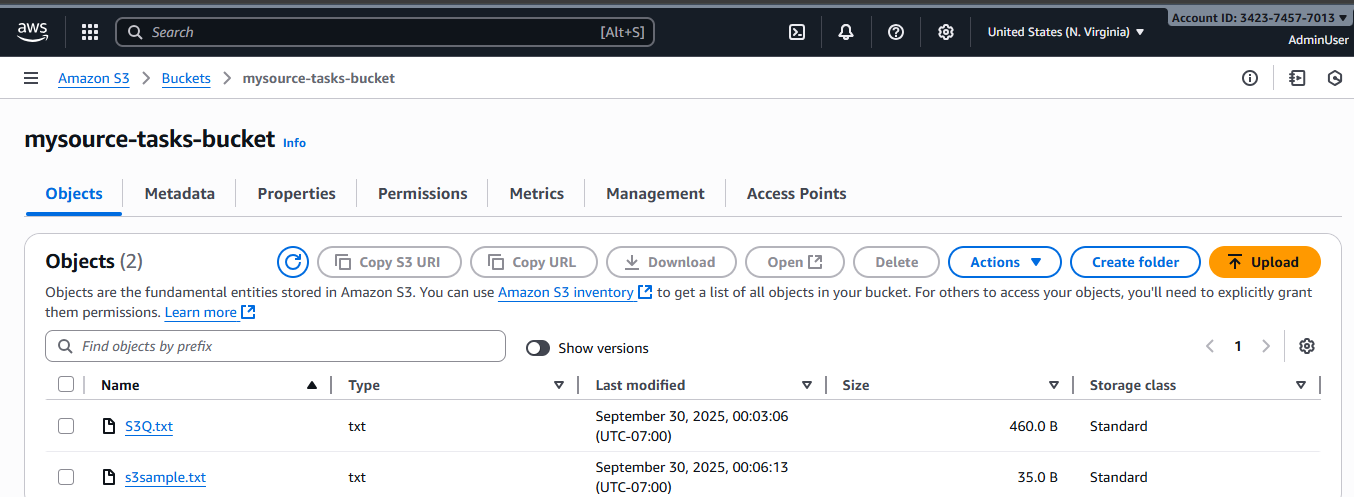
1. Copy the **User ARN** (it looks like:

arn:aws:iam::123456789012:user/Admin

## ****Step 2: Create Bucket Policy****

Go to **S3 → Your Bucket → Permissions → Bucket Policy** and paste something like this (replace placeholders):





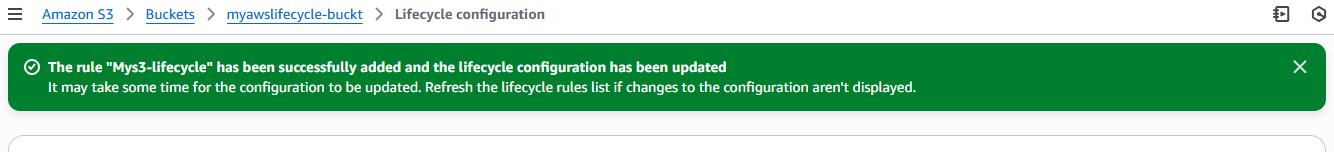
**5.Set up lifecycle policies to automatically transition or delete objects based on specific criteria.**

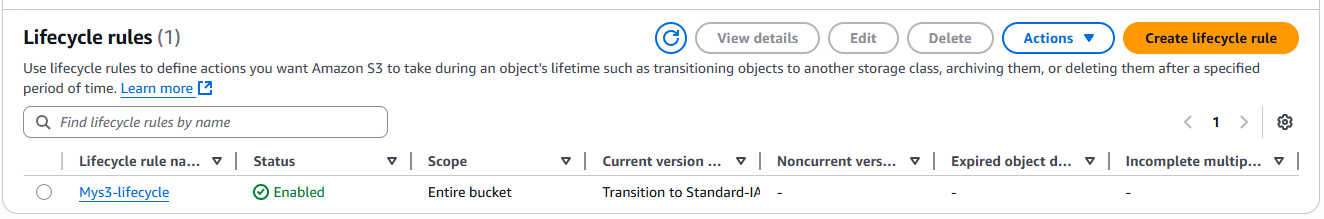
### Step-by-Step: Create a Lifecycle Rule in AWS S3 Console

1. **Access S3 Console**  
   Navigate to the Amazon S3 Console.
2. **Select Your Bucket**  
   Click on the name of the bucket where you want to apply the lifecycle policy.
3. **Go to the Management Tab**  
   In the bucket details page, click on the **Management** tab.
4. **Create a New Lifecycle Rule**  
   Click on **Create lifecycle rule**.
5. **Configure Rule Name and Scope**
   * **Rule name**: Enter a descriptive name, e.g., AutoTransitionAndExpire.
   * **Scope**: Choose whether to apply the rule to all objects or a subset (e.g., using a prefix or tags).
6. **Set Transition Actions**
   * Check **Transition current versions of objects between storage classes**.
   * Click on **Add transition**.
     + **Days after object creation**: Enter 30.
     + **Storage class**: Select STANDARD\_IA.
   * Click on **Add transition** again.
     + **Days after object creation**: Enter 90.
     + **Storage class**: Select GLACIER.
7. **Set Expiration Actions**
   * Check **Expire current versions of objects**.
   * **Days after object creation**: Enter 365.
8. **Set Additional Options (Optional)**
   * Check **Delete expired object delete markers or incomplete multipart uploads** to clean up expired delete markers and incomplete uploads.
9. **Review and Create**
   * Review your settings.
   * Click **Create rule** to apply the lifecycle policy.

### ✅ Summary of Actions

* **Transition current versions**:
  + To STANDARD\_IA after 30 days.
  + To GLACIER after 90 days.
* **Expire current versions**:
  + Delete objects after 365 days.
* **Additional cleanup**:
  + Remove expired delete markers and incomplete multipart uploads.





**7.Push some objects to S3 using the AWS CLI.**

**Prerequisites**

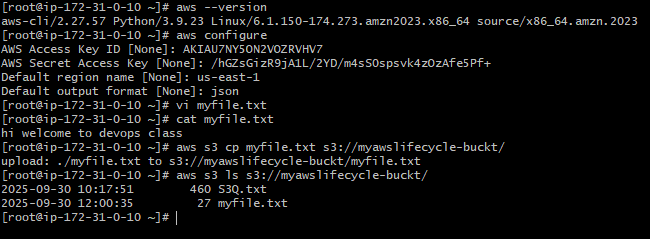
* AWS CLI installed (aws --version to check).
* Run aws configure and set:
  + **Access Key ID**
  + **Secret Access Key**
  + **Default Region**
  + **Output format (json/table/text)**

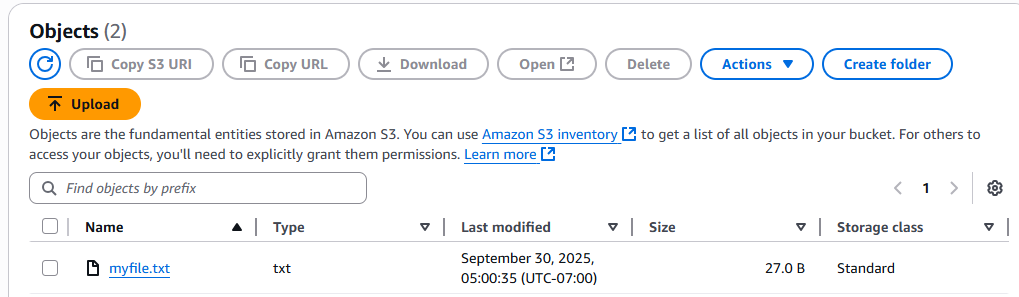
**Upload a Single File**

aws s3 cp myfile.txt s3://my-private-bucket/

## ****Verify Objects Uploaded****

aws s3 ls s3://my-private-bucket/





**8.Write a Bash script to create an S3 bucket.**

**Bash Script: create-s3-bucket.sh**

#!/bin/bash

# Exit if any command fails

set -e

# Ask for bucket name

read -p "Enter S3 bucket name: " BUCKET\_NAME

# Ask for AWS region

read -p "Enter AWS region (e.g., us-east-1): " REGION

echo "Creating S3 bucket: $BUCKET\_NAME in region: $REGION ..."

# Create bucket (region-specific)

if [ "$REGION" == "us-east-1" ]; then

# Special case: us-east-1 does not need LocationConstraint

aws s3api create-bucket --bucket "$BUCKET\_NAME" --region "$REGION"

else

aws s3api create-bucket \

--bucket "$BUCKET\_NAME" \

--region "$REGION" \

--create-bucket-configuration LocationConstraint="$REGION"

fi

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

**How to Use**

1. Save the script as create-s3-bucket.sh
2. Make it executable:
3. chmod +x create-s3-bucket.sh
4. Run it:
5. ./create-s3-bucket.sh
6. Enter:
   * **Bucket name** (must be globally unique across AWS)
   * **Region** (e.g., us-east-1, ap-south-1, etc.)

