1. Create s3 bucket and upload some objects to s3.

.Click **"Create bucket."**

.Enter a **globally unique bucket name**.

.Select a **region**.

.Set desired options (Versioning, Encryption, Permissions).

.Click **"Create bucket."**

A screenshot of a computer

AI-generated content may be incorrect.

.Open your newly created bucket.

.Click **"Upload."**

.Click **"Add files"** or **"Add folder."**

.Choose the files from your computer.

.Set permissions if needed (optional).

.Click **"Upload."**

A screenshot of a computer

AI-generated content may be incorrect.

.Go to the browser and search with url: https://moranithinbucket.s3.us-east-1.amazonaws.com/feb.PNG

A screenshot of a computer

AI-generated content may be incorrect.

1. Deploy static website in s3 bucket.

Go to [S3 Console](https://s3.console.aws.amazon.com/s3/home).

.Click **"Create bucket."**

.Enter a **globally unique bucket name**.

.Choose the region.

.Uncheck **Block all public access** (you need public access for static website hosting).

.Acknowledge the warning.

. Click **"Create bucket."**

A screenshot of a computer

AI-generated content may be incorrect.

**Enable Static Website Hosting**

.Open your bucket.

.Go to **"Properties."**

.Scroll down to **"Static website hosting."**

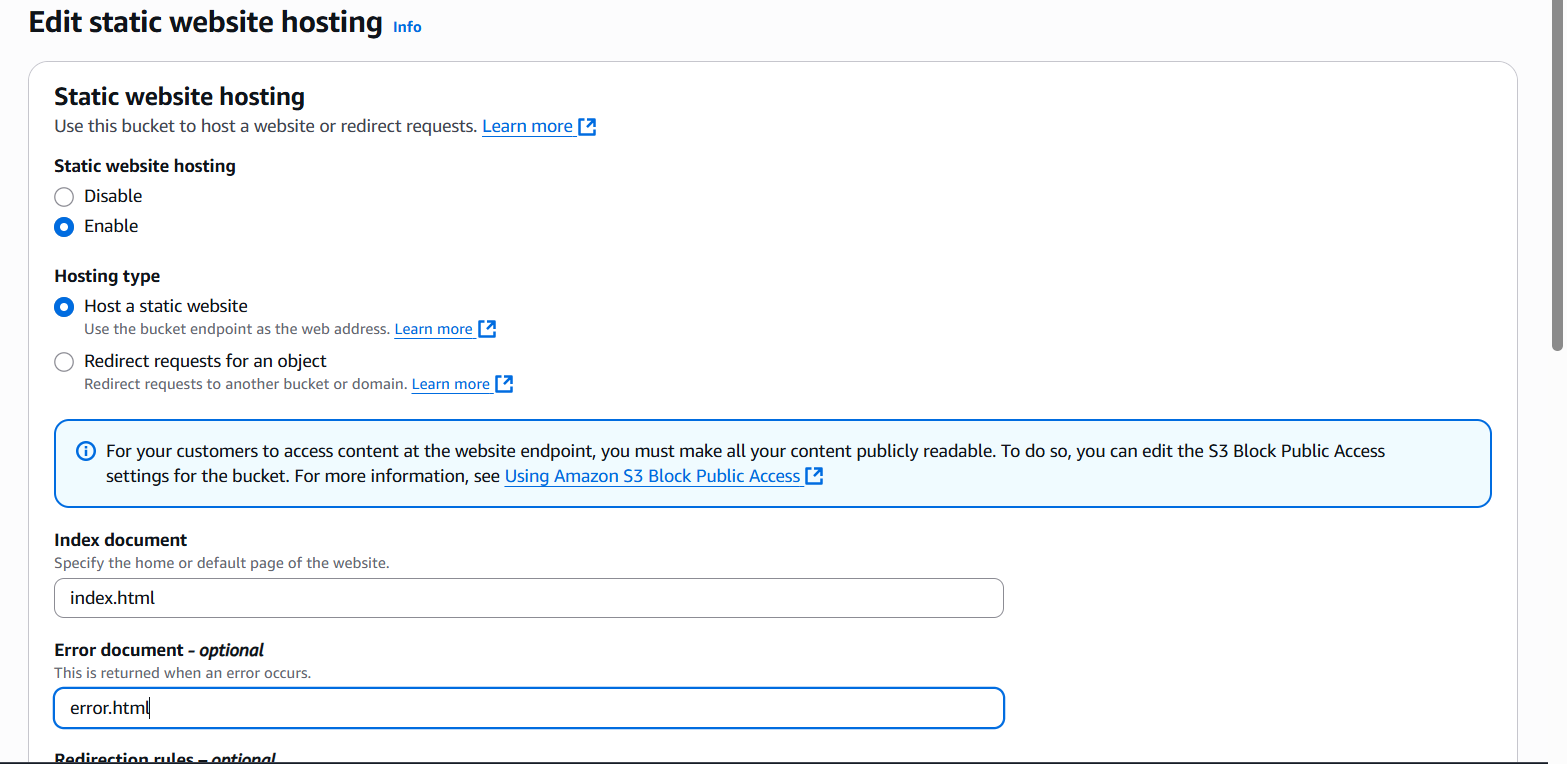
.Click **"Edit."**

.Enable static website hosting.

.Enter **Index document** (e.g., index.html).

.(Optional) Enter **Error document** (e.g., error.html).

.Click **"Save changes."**



.Create 2 files in gitbash

A computer screen shot of a program code

AI-generated content may be incorrect.

**Upload Your Website Files**

.Go to the **"Objects"** tab of your bucket.

.Click **"Upload."**

.Add your files (index.html, css, js).

.Before finishing, go to **"Permissions"**, and make sure files are **public**

A screenshot of a computer

AI-generated content may be incorrect.

**Access Your Static Website**

.Go back to **"Properties" → "Static website hosting."**

.Copy the **Bucket website endpoint** URL.

.Open it in your browser — your site is live!

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Enable cross region replication on s3 buckets.

**Create Source and Destination Buckets**

.Go to [S3 Console](https://s3.console.aws.amazon.com/s3/home).

.Create **Source Bucket** (e.g., source-bucket-xyz) in Region A.

.Create **Destination Bucket** (e.g., destination-bucket-abc) in Region B.

A screenshot of a computer

AI-generated content may be incorrect.

**Enable Versioning on Both Buckets**

.Open each bucket.

.Go to **Properties.**

.Enable **Versioning.**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Set Up Replication Rule in Source Bucket**

.Go to your **Source Bucket** → **Management.**

.Click **"Create replication rule."**

.Enter rule name.

.Scope:

* + **Entire bucket** or specific prefix/tags.

Destination:

* + Choose **"Specify a bucket in another account"** (optional) or **"This account."**
  + Select **Destination Bucket**.

.IAM Role:

* + Choose **Create new role** (or select existing replication role).

.(Optional) Replicate delete markers and existing objects.

.Save the rule.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Validate**

.Upload an object to the **Source Bucket**.

.Wait for replication (can take minutes).

.Check **Destination Bucket** in other region — object should appear.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Configure bucket policy,only Admin user can see the objects of s3 bucket.

**Step 1: Confirm IAM user Admin exists and get ARN**

Go to **IAM → Users**.

Confirm the user Admin exists.

Copy its **full ARN**,

Step 2: Configure bucket policy to allow only this user

**Step 3: Apply the policy**

**Option 1: From AWS Console**

Go to **S3 → Select bucket moranithinlifecyclebucket → Permissions → Bucket policy**.

Paste the above JSON.

Save.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Test the access**

**Test as Admin user:**

Login to AWS Console using Admin user.

Try **list objects** or **upload files** → ✅ Should work.

A computer screen shot of a black screen

AI-generated content may be incorrect.

1. Setup lifecycle policies to automatically transition or delete objects based on specific criteria.

**Step 1: Create an S3 Bucket**

1. Go to **AWS Management Console → S3 → Create bucket**.
2. **Bucket name**:  
   Example: moranithin-lifecycle-bucket.
3. **Region**:  
   Choose US East (N. Virginia) (us-east-1) (or your preferred region).
4. Keep **Block all public access ON** (recommended).
5. Click **Create bucket**.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Create Lifecycle Rule (Corrected)**

1. Go to **S3 → Click your bucket → Management tab → Lifecycle rules**.
2. Click **Create lifecycle rule**.
3. Enter rule name:  
   Example: CorrectTransitionAndDeleteRule.
4. **Scope of rule**:  
   Select **Apply to all objects in the bucket**.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Add Transitions**

**4.1 First transition (Standard-IA after 30 days)**

* Check **Transition current versions of objects between storage classes**.
* Click **Add transition**.
  + Days after object creation: **30**
  + Destination storage class: **Standard-IA**

**4.2 Second transition (Glacier after 60 days)**

* Click **Add transition again**.
  + Days after object creation: **60**
  + Destination storage class: **Glacier Flexible Retrieval**

**Step 5: Configure expiration (delete after 365 days)**

* Check **Expire current versions of objects**.
  + Days after object creation: **365**

A screenshot of a computer

AI-generated content may be incorrect.

**Step 6: Review and create rule**

1. Confirm your settings:
   * ✔ Day 30 → Standard-IA
   * ✔ Day 60 → Glacier Flexible Retrieval
   * ✔ Day 365 → Delete
2. Click **Create rule**.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Push some objects in s3 using AWS CLI.
   * + Run: aws configure
     + Then give access key, securitykey,region,json.

THEN RUN:

=>aws sts get-caller-identity.

Create some test files locally

Touch car > file1.txt

Touch bus > file1.txt

Touch bike > file1.txt

Upload (push) files to S3 bucket

aws s3 cp file1.txt s3://moranithin-lifecycle-bucket/

aws s3 cp file2.txt s3://moranithin-lifecycle-bucket/

aws s3 cp file3.txt s3://moranithin-lifecycle-bucket/

Verify files in the bucket

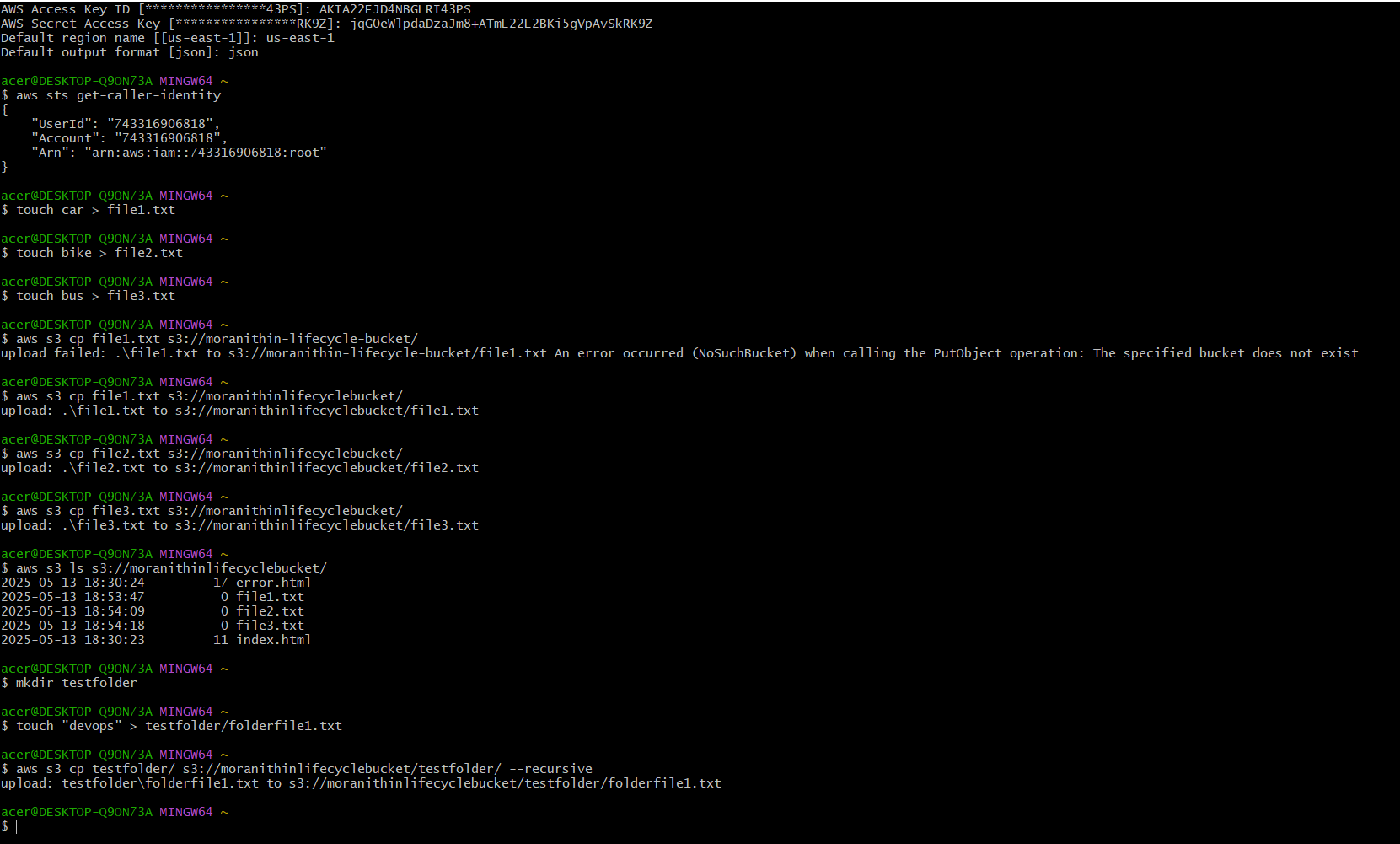
aws s3 ls s3://moranithin-lifecycle-bucket/

Upload a full folder recursively

mkdir testfolder

echo "File in folder" > testfolder/folderfile1.txt

aws s3 cp testfolder/ s3://moranithin-lifecycle-bucket/testfolder/ --recursive



A screenshot of a computer

AI-generated content may be incorrect.

1. Write a bash script to create s3 bucket.

Run: touch creates3bucket.sh

=> vi creates3bucket.sh

#!/bin/bash

# Script to create an S3 bucket intelligently for any region

BUCKET\_NAME=$1

REGION=$2

if [ -z "$BUCKET\_NAME" ] || [ -z "$REGION" ]; then

echo "Usage: $0 <bucket-name> <region>"

exit 1

fi

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

# Special case for us-east-1 where LocationConstraint is not required

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

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

# Check if the bucket was created successfully

if [ $? -eq 0 ]; then

echo "✅ Bucket created successfully."

else

echo "❌ Failed to create bucket."

Fi

=>Run:

=> chmod +x creates3bucket.sh

=> ./creates3bucket.sh morareddynewbucket us-east-1

A computer screen shot of text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Upload one 1 gb of file to s3 using cli.

Step 1: Create a 1 GB file using Windows command inside Git Bash

cmd /c "fsutil file createnew testfile-1gb.bin 1073741824"

Upload the file to S3 using AWS CLI

aws s3 cp /c/Users/acer/Desktop/testfile-1gb.bin s3://morareddynewbucket/

A black screen with many colorful text

AI-generated content may be incorrect.

After uploading file it will show in your bucket

A screenshot of a computer

AI-generated content may be incorrect.