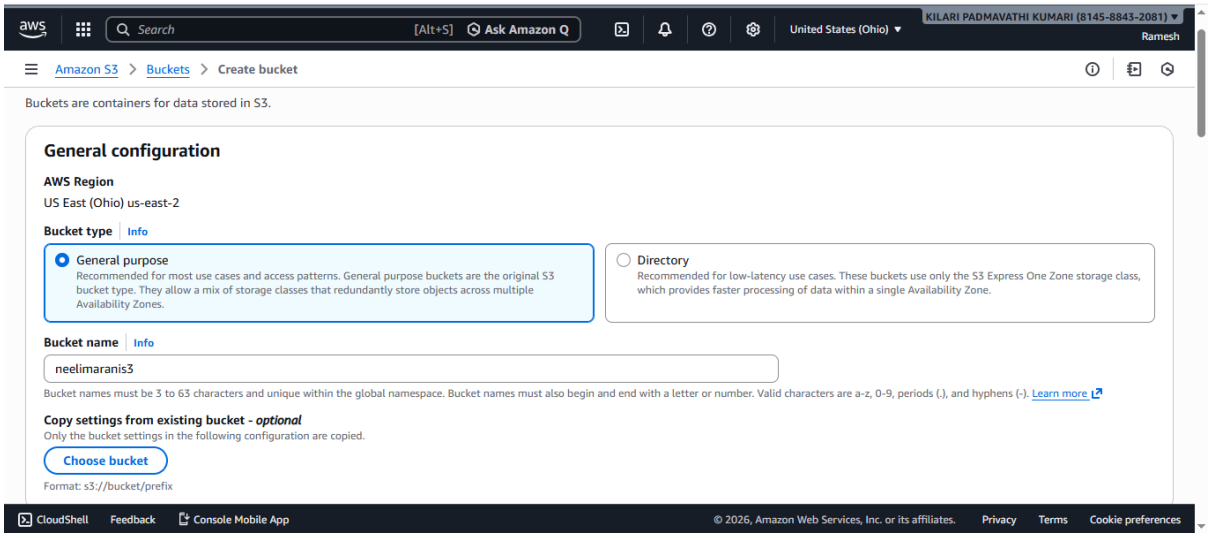
S3-TASK1

Amazon S3 (Simple Storage Service) is a highly scalable, durable, and secure cloud object storage service from Amazon Web Services (AWS).

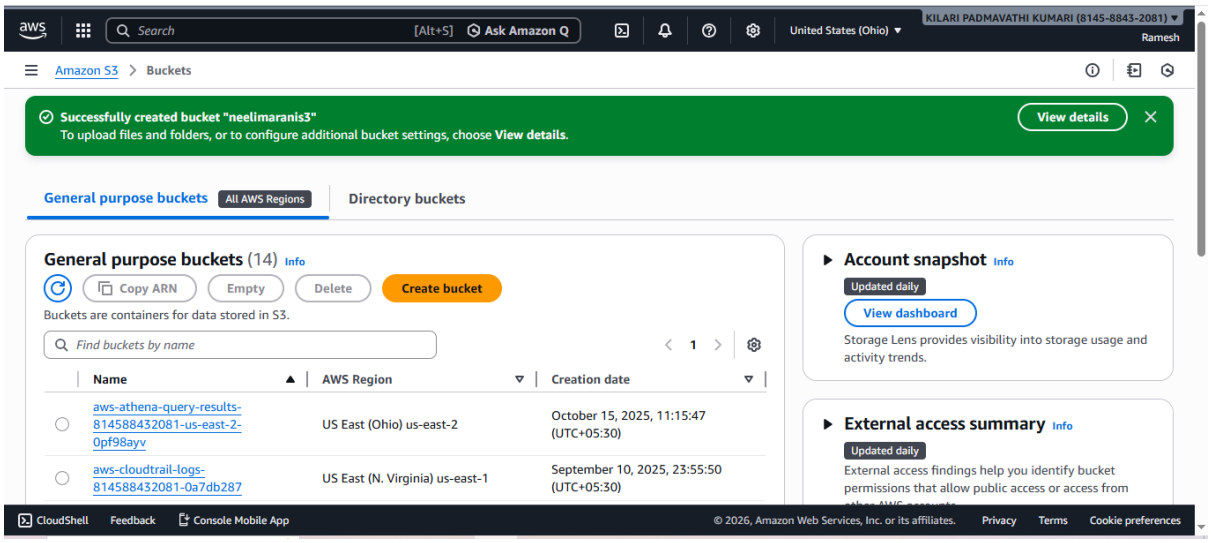
1. Create an S3 bucket and upload some objects to S3.

- Open Amazon S3
- Click on Bucket
- Click on Create bucket.

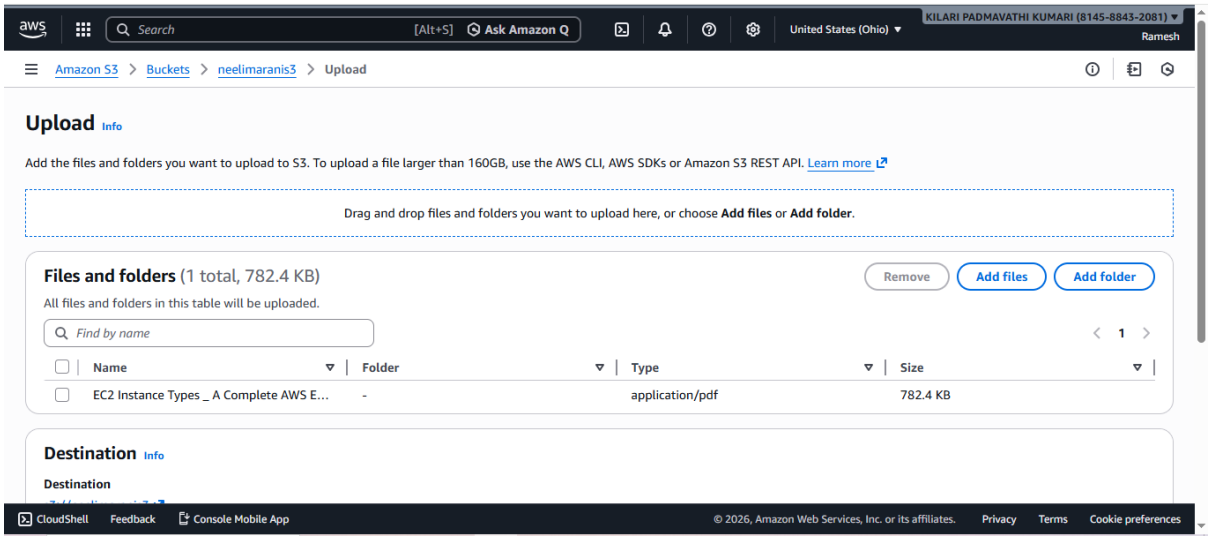


- General configuration
- AWS region what we have choose before
- Bucket type →general purpose
- Bucket Name →neelimaranis3(Unique Name)
- Click on create bucket.

S3-TASK1

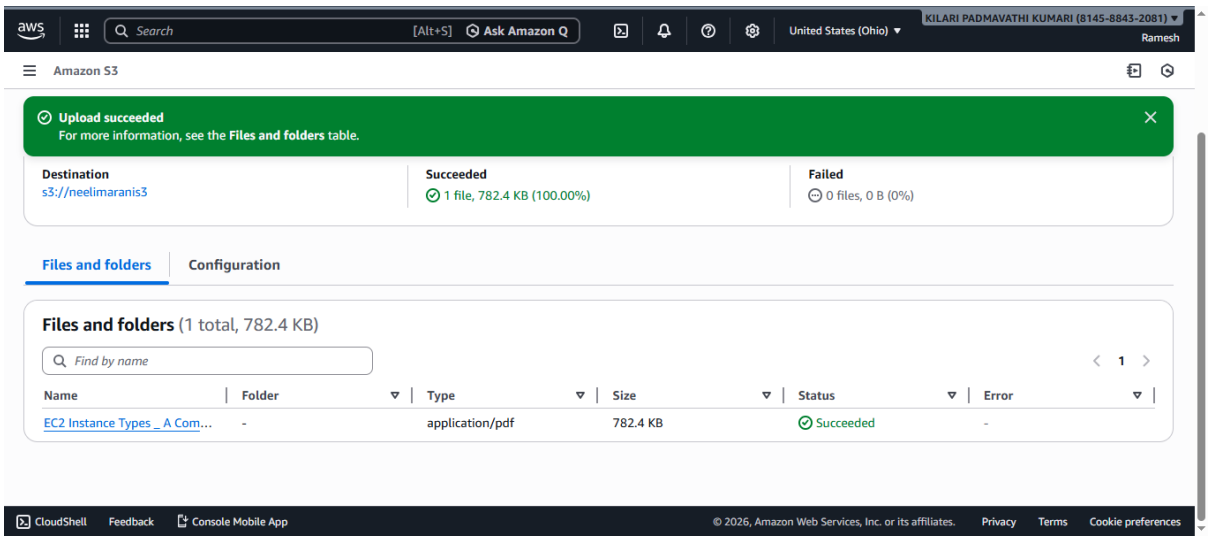


- The above image shows bucket created successfully.



- To upload objects into S3 bucket
- Go to buckets choose bucket which is already created
- Click on upload and
- Click on add files and browse from pc.

S3-TASK1



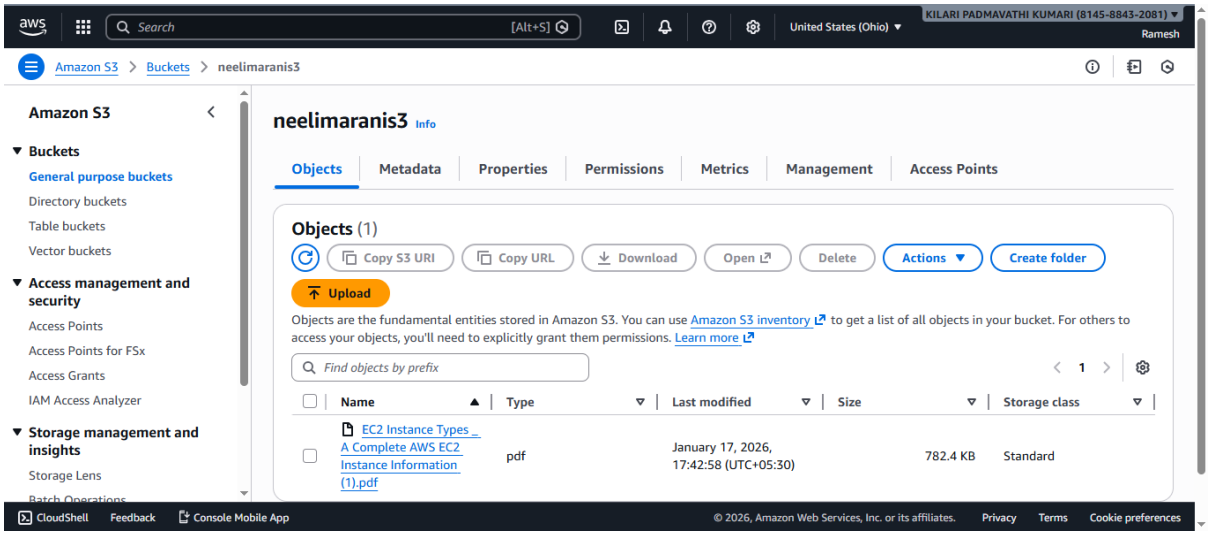
- The above image shows upload files successful and we can verify it.

S3-TASK1

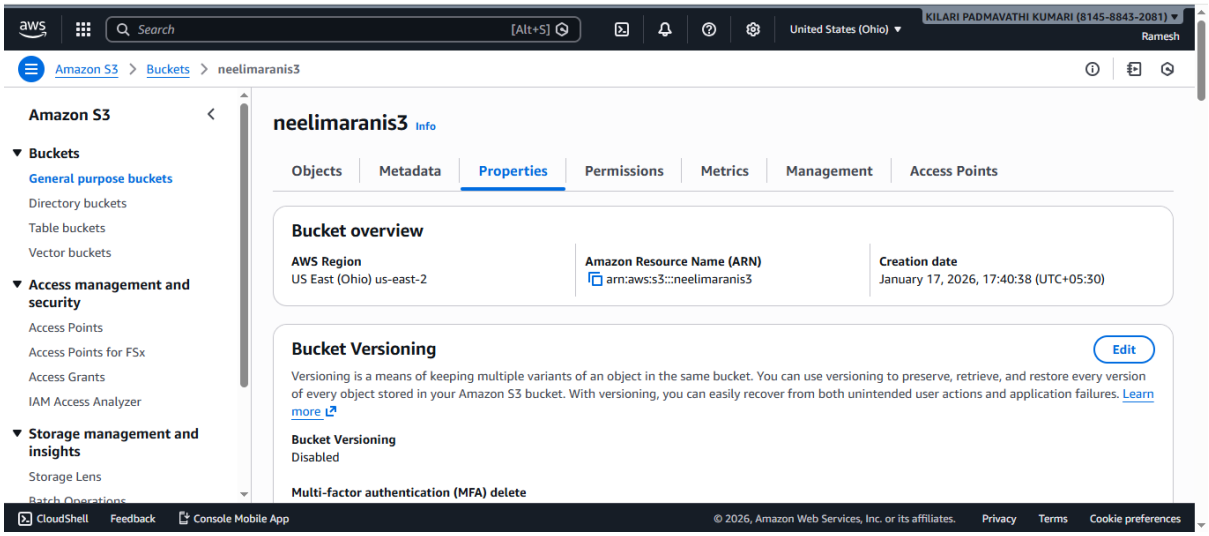
2. Deploy a static website in the S3 bucket.

Step 1: Enable Static Website Hosting

Go to **AWS Console → S3 → Your Bucket**

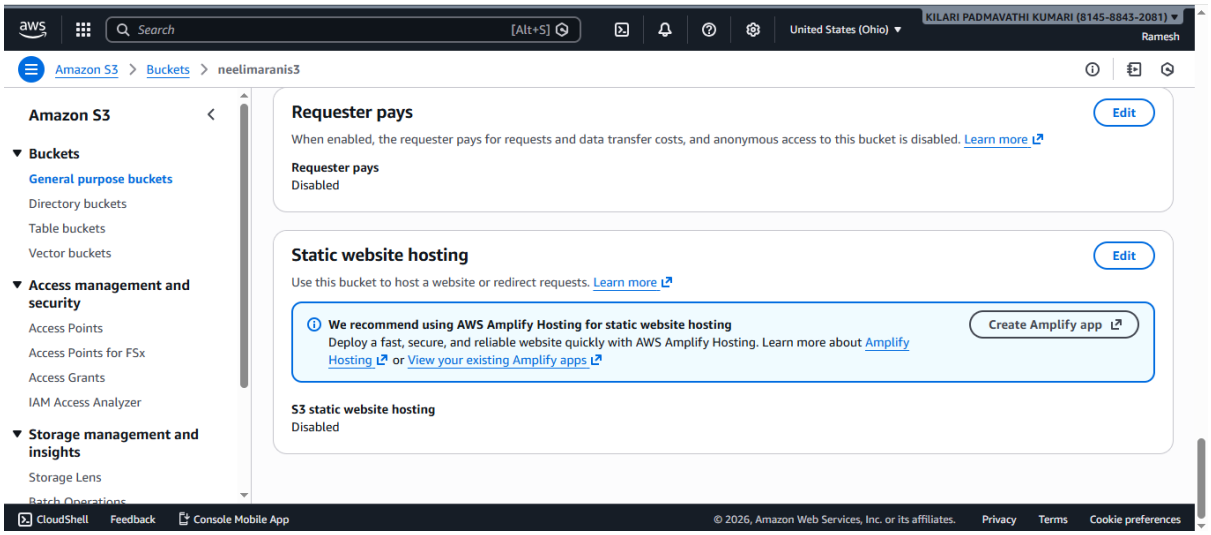


- Click **Properties** tab

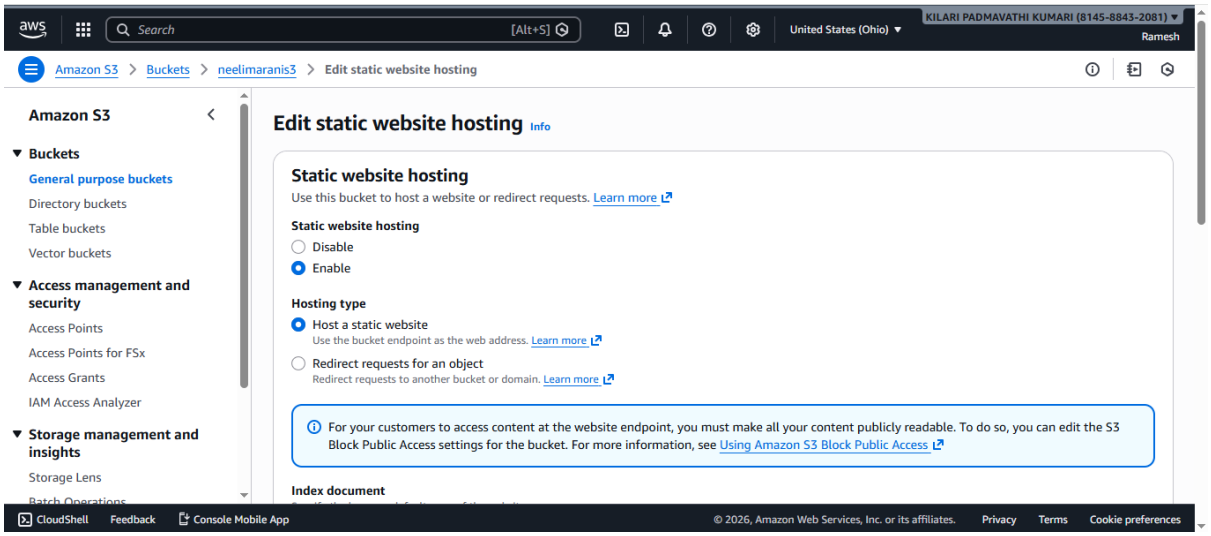


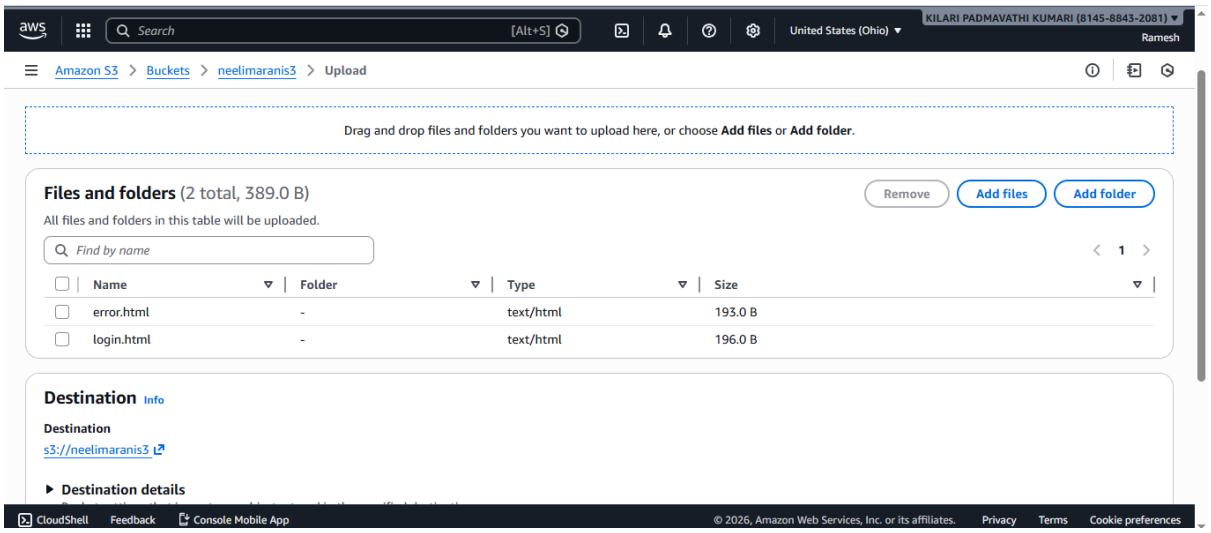
S3-TASK1

- Scroll down to **Static website hosting** → Click **Edit**

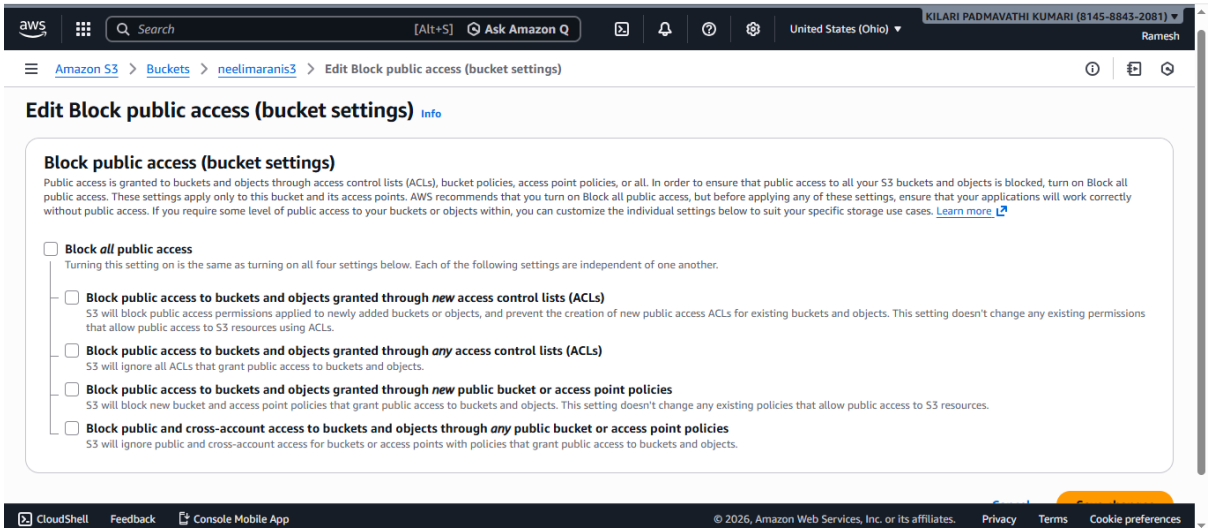


- Select **Enable**





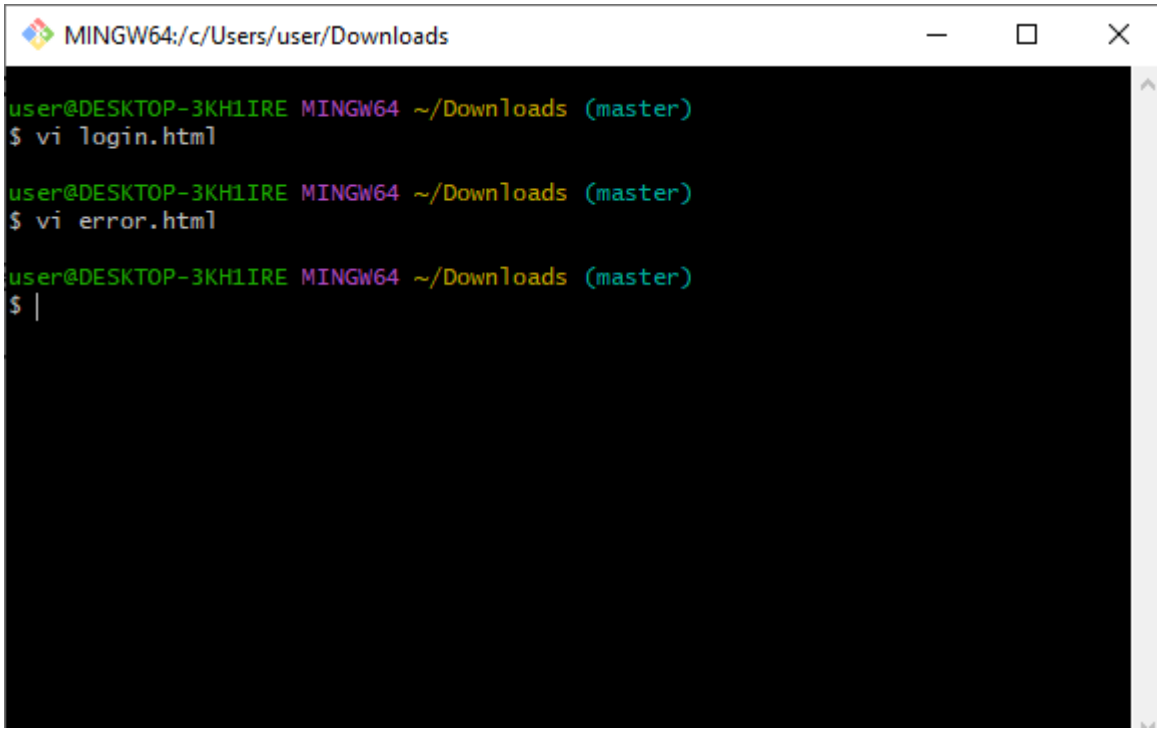
S3-TASK1



- Save changes and confirm it.

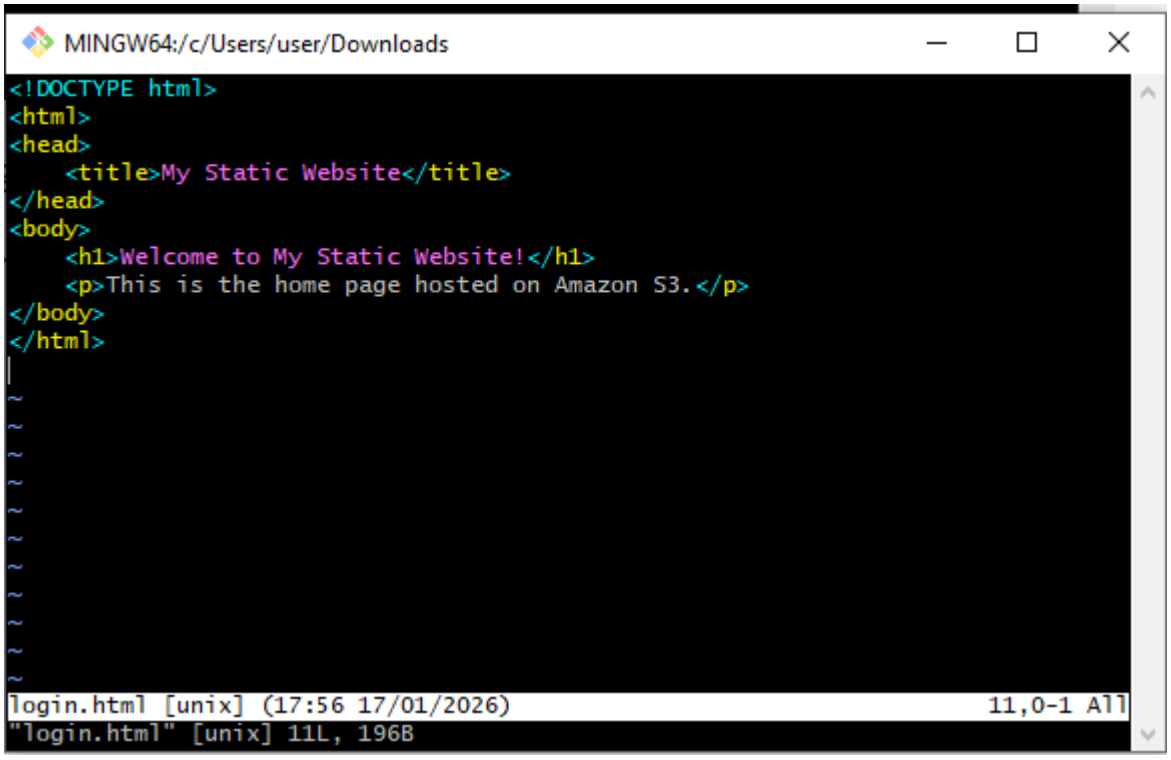
  Choose **Host a static website**

  Enter:

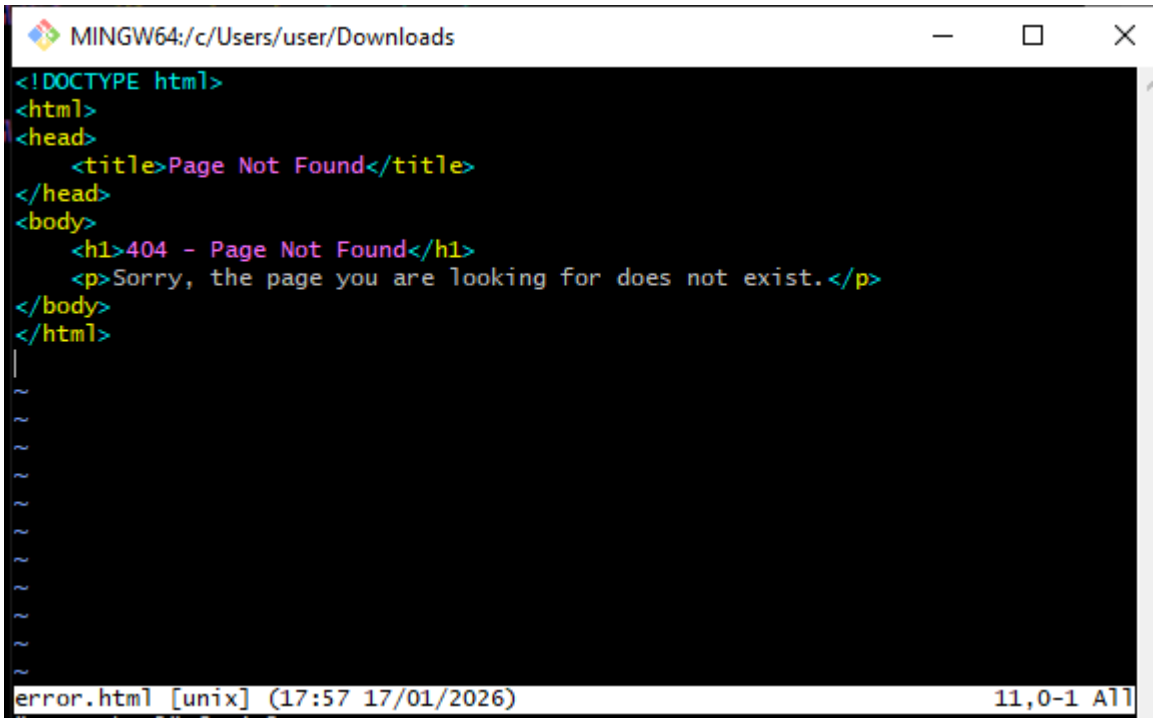


- **Index document:** login.html

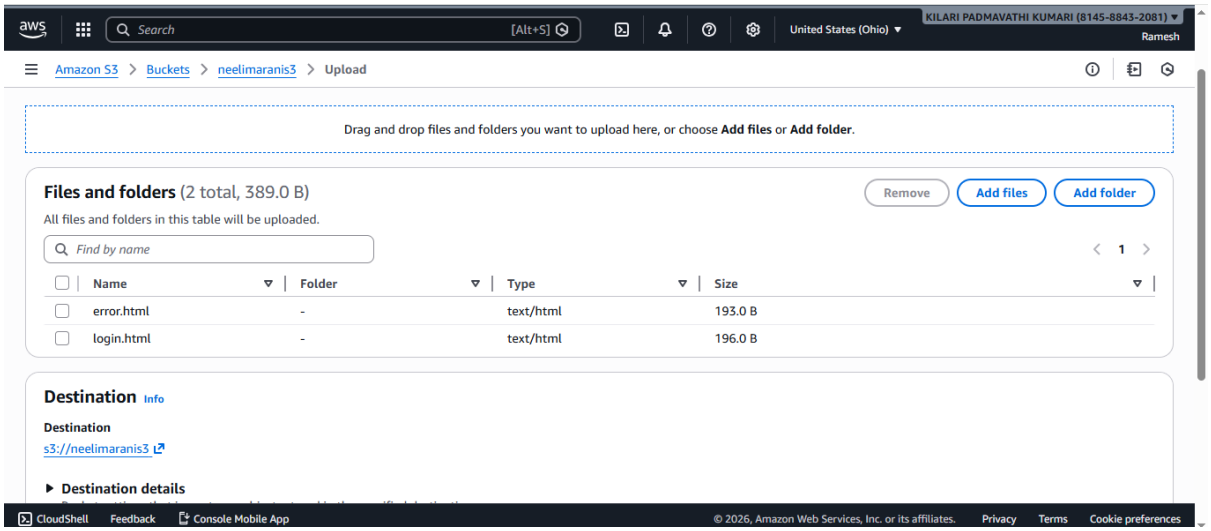
S3-TASK1



- **Error document:** error.html (optional)



S3-TASK1



- Click **Save changes**

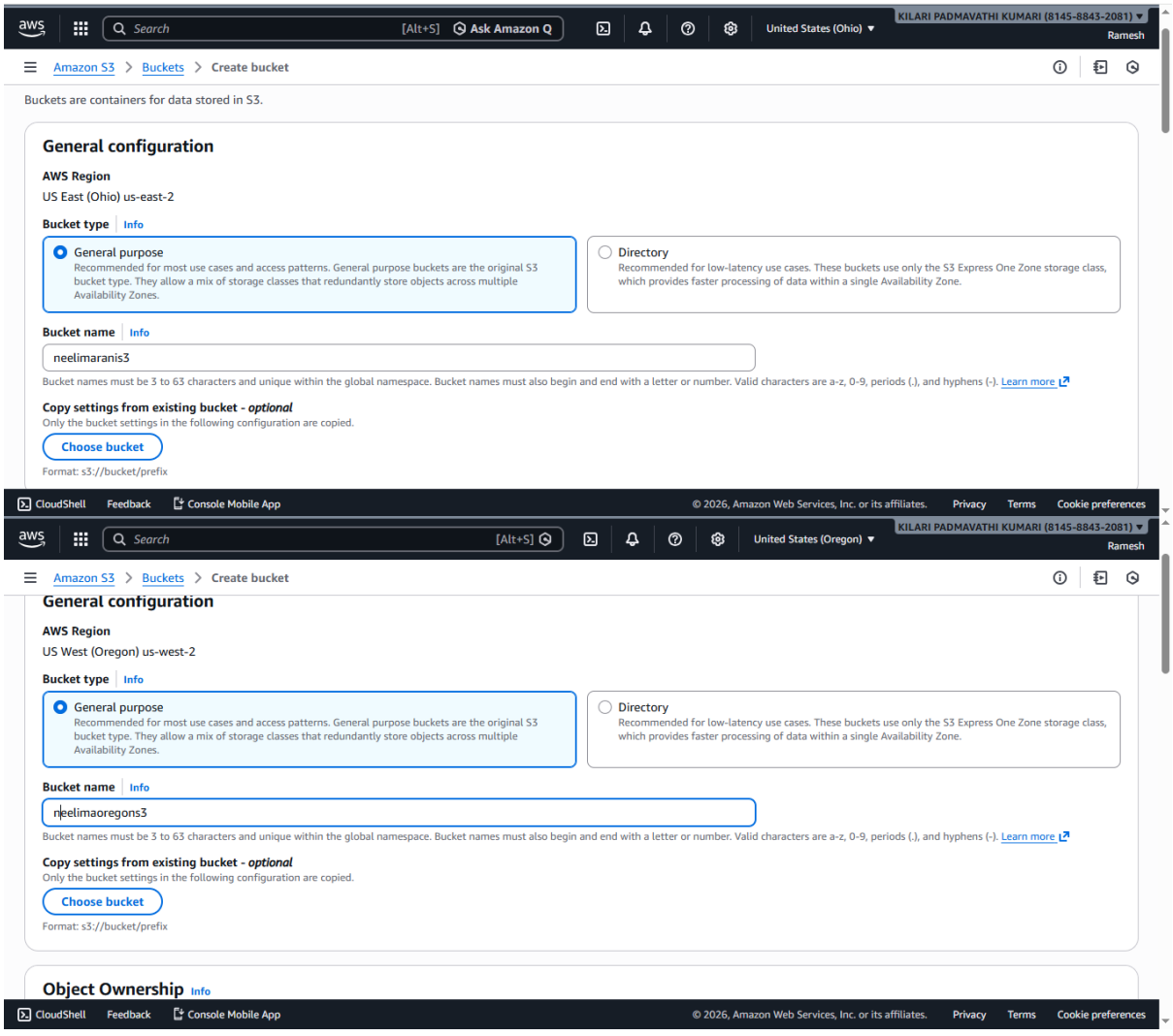
S3-TASK1

3. Enable cross-region replication on S3 buckets.

You need **two buckets**:

a. **Source bucket**
b. **Destination bucket** (in a different region)

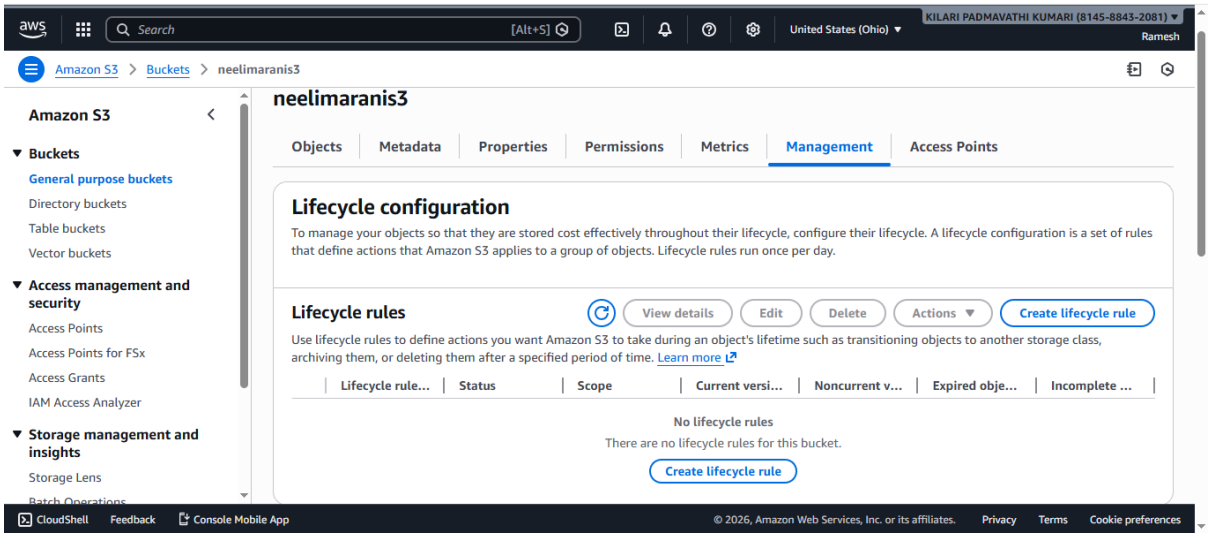
Both buckets must have **versioning enabled** (CRR requires versioning).



S3-TASK1

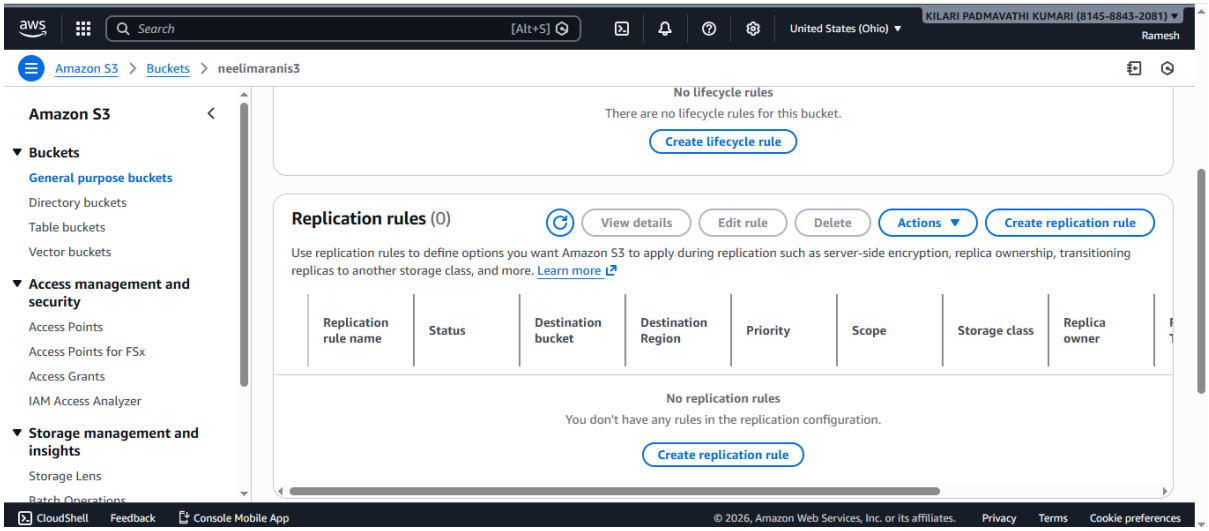
Step 1: Go to Source Bucket

- Log in to **AWS Console → S3 → Select your source bucket** (e.g.,neelimaranis3)
- Click **Management** tab



Step 2: Create Replication Rule

- Scroll down to **Replication rules** → Click **Create rule**
- Enter a **Rule name**, e.g., CRR-to-oregon

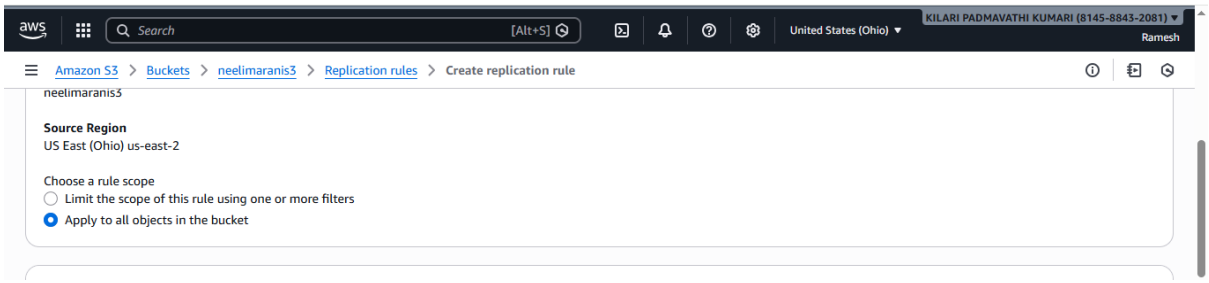


S3-TASK1

Step 3: Choose Rule Scope

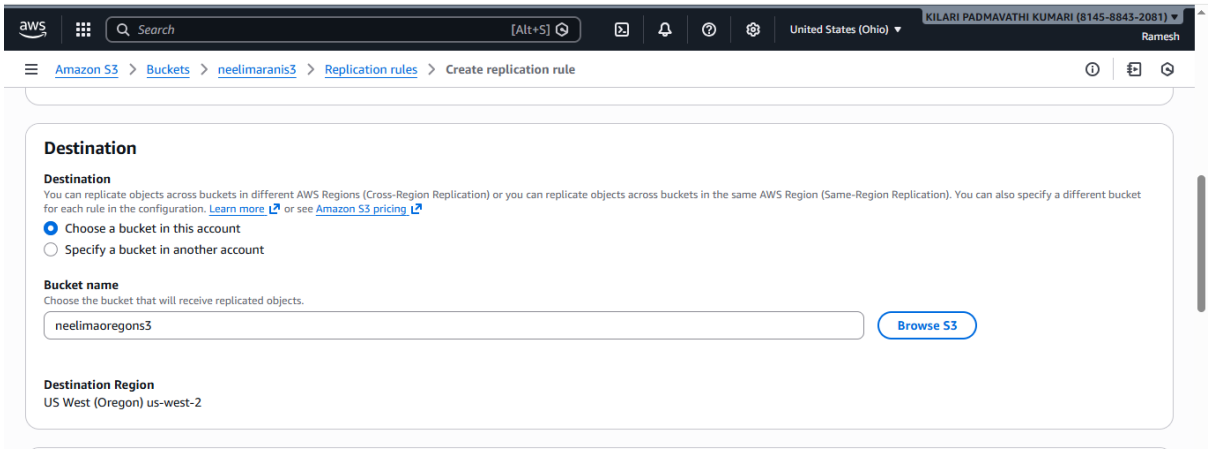
Rule scope:

- Select **Apply to all objects**
- Or choose **Filter by prefix or tags** if you only want certain objects replicated



Step 4: Choose Destination Bucket

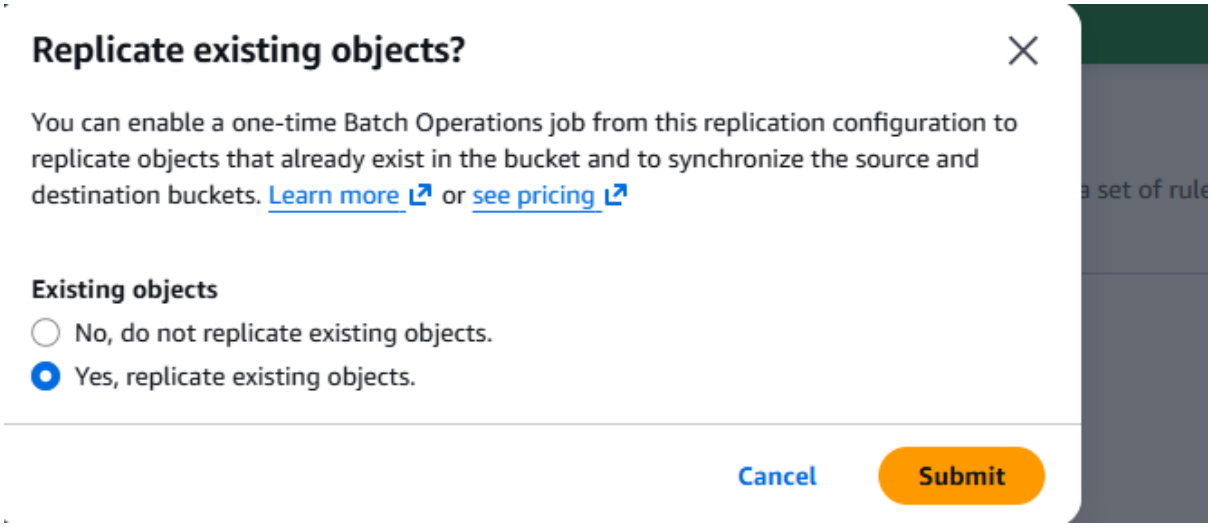
- Destination → Select Another AWS Region
- Bucket → Select your destination bucket (e.g., neelimaoregons3-bucket)
- Storage class in destination → leave as Standard (or choose another like Glacier if needed)

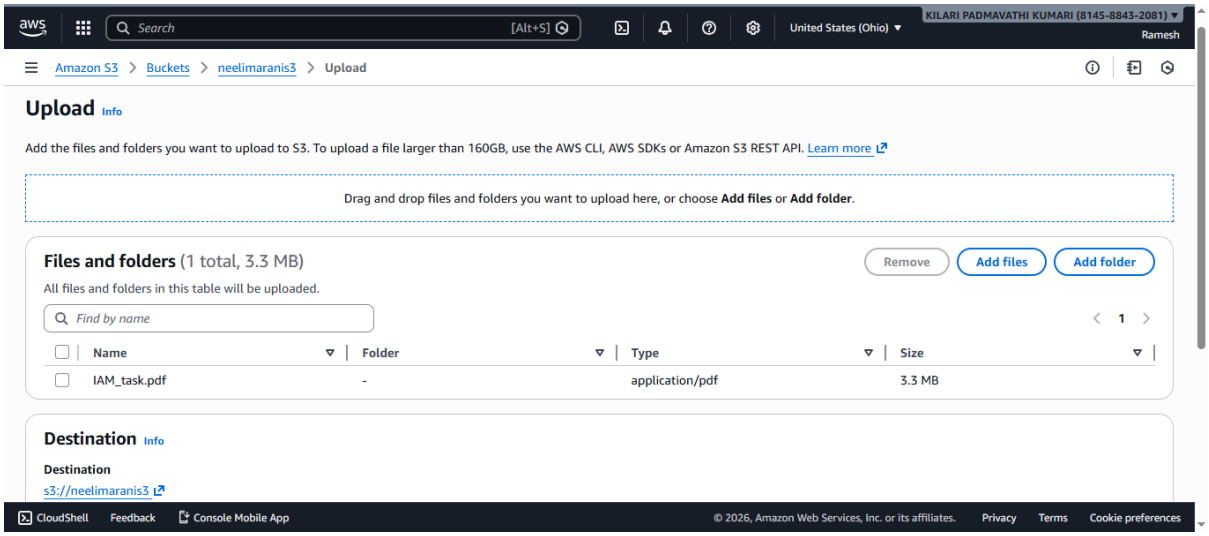


S3-TASK1

Step 5: IAM Role for Replication

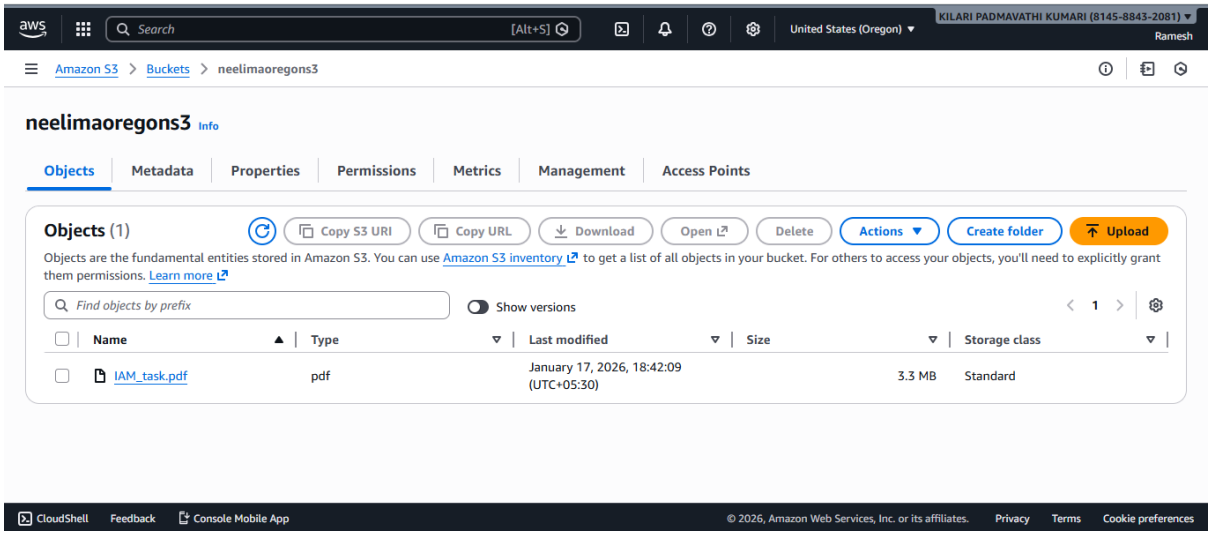
1. Choose or create a role:
   - Select Create a new role
   - AWS will create a role automatically (example: S3ReplicationRole)
   - This role allows S3 to replicate objects to the destination bucket
2. Click Save





- Here we have upload files in source bucket nothing neelimaranis3.

S3-TASK1



- Here it is reflected in neelimaoregons3 due to replication rule

S3-TASK1

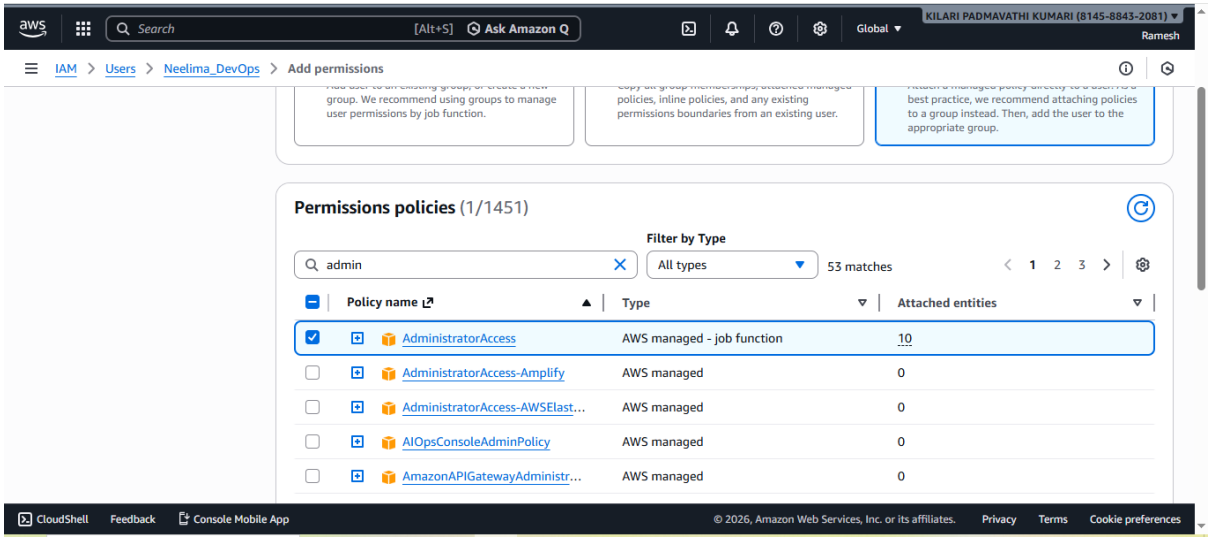
S3 Cross-Region Replication – Short Steps

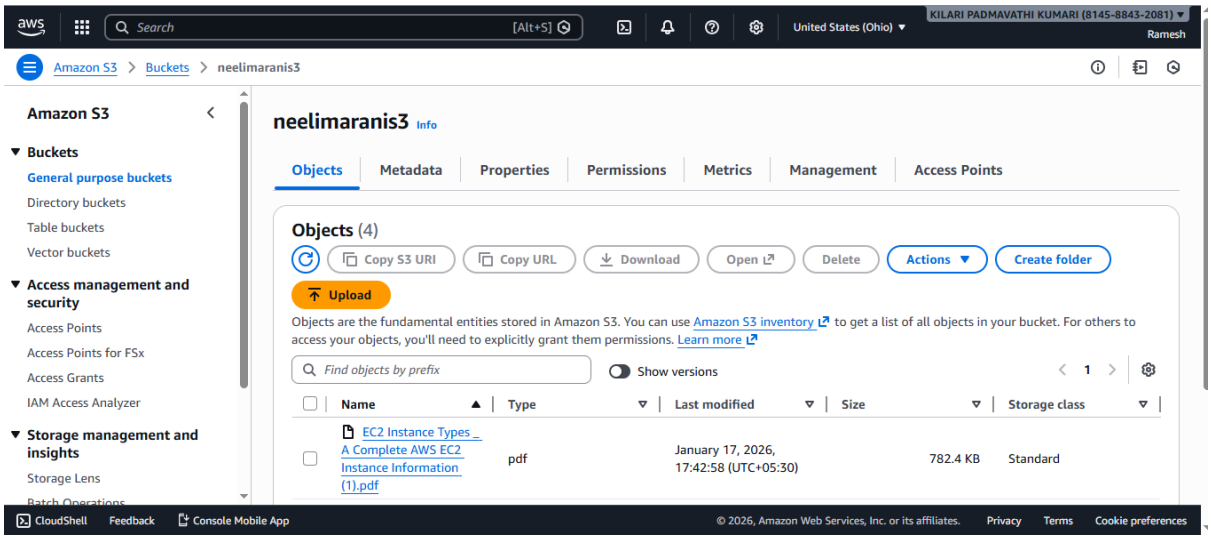
1. Create 2 buckets
    - Source bucket → Region 1 (example: us-west-2)
    - Destination bucket → Region 2 (example: ap-south-1)
2. Enable Versioning
    - Go to S3 → Source bucket → Properties → Versioning → Enable
    - Go to S3 → Destination bucket → Properties → Versioning → Enable
3. Open Source Bucket
    - S3 → Source bucket → Management tab
4. Create Replication Rule
    - Click Replication rules → Create rule
    - Rule scope: Apply to all objects
5. Select Destination
    - Choose Another AWS Region
    - Select the destination bucket
6. IAM Role
    - Choose Create new role (AWS creates it automatically)
7. Create Rule
    - (Optional) Enable Replicate existing objects
    - Click Create rule
8. Verify
    - Upload a file to source bucket
    - File appears automatically in destination bucket

S3-TASK1

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

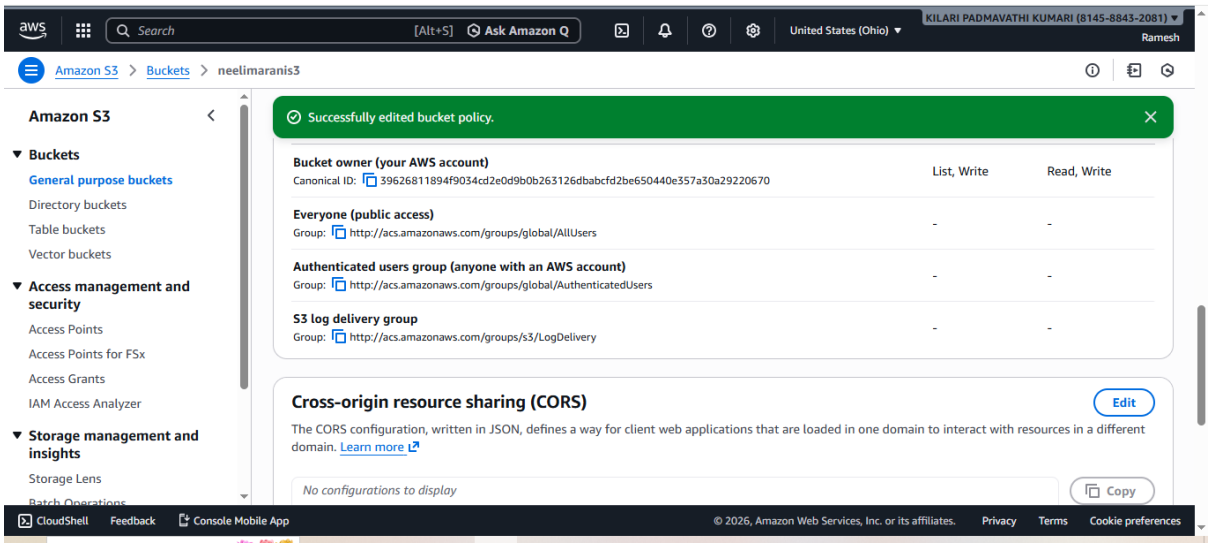
- In this admin user can see the objects means user should AdminstratorAcces.
- IAM→user→Neelima_Devops→ Add permissions add AdminstratorAccess.



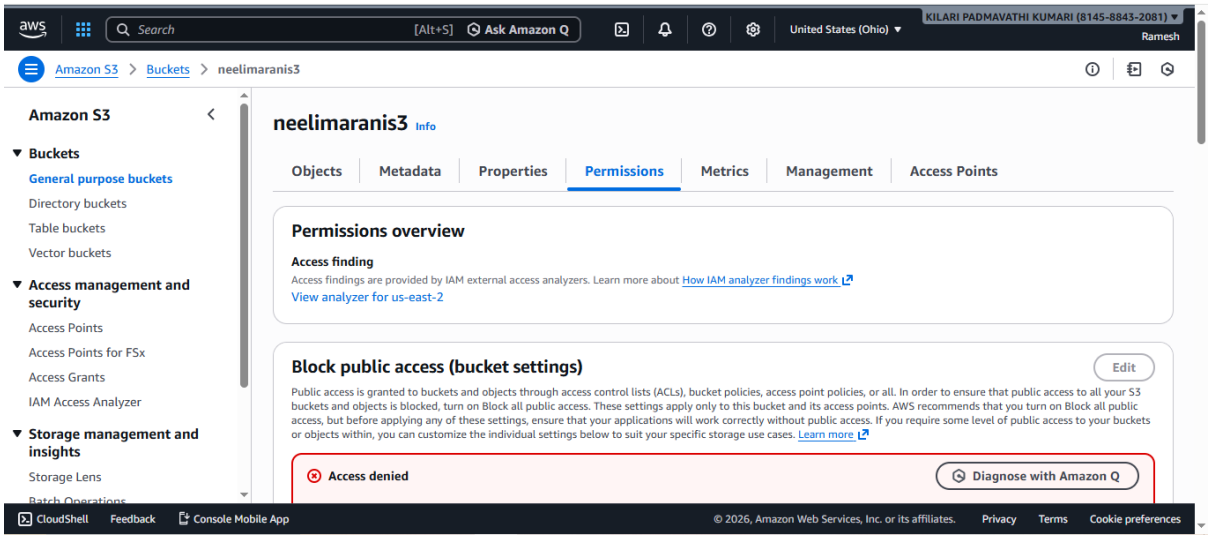


- Go to Amazon s3
- Click on buckets
- Select required bucket what we have created

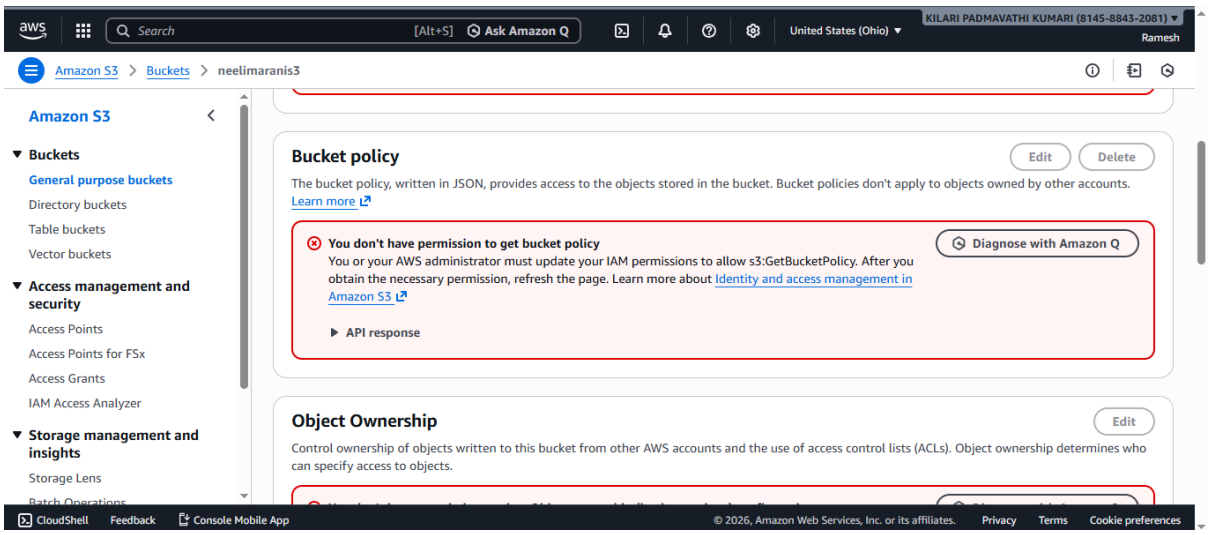
S3-TASK1



- Go to edit bucket policy and policy to it
- The above image shows successfully edit bucket policy.
- Now only admin user can access bucket .



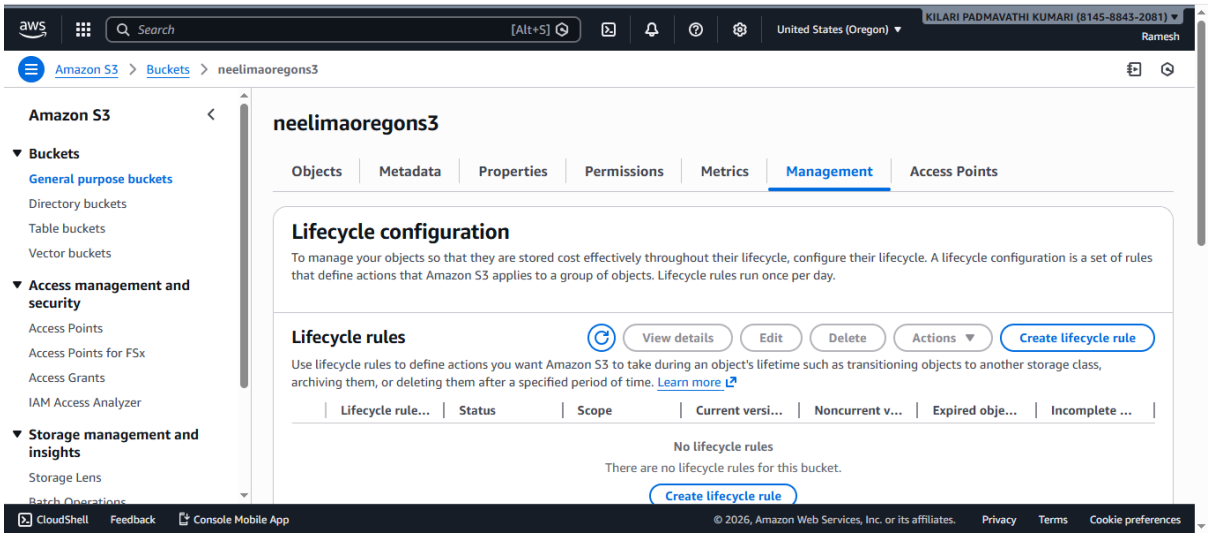
S3-TASK1



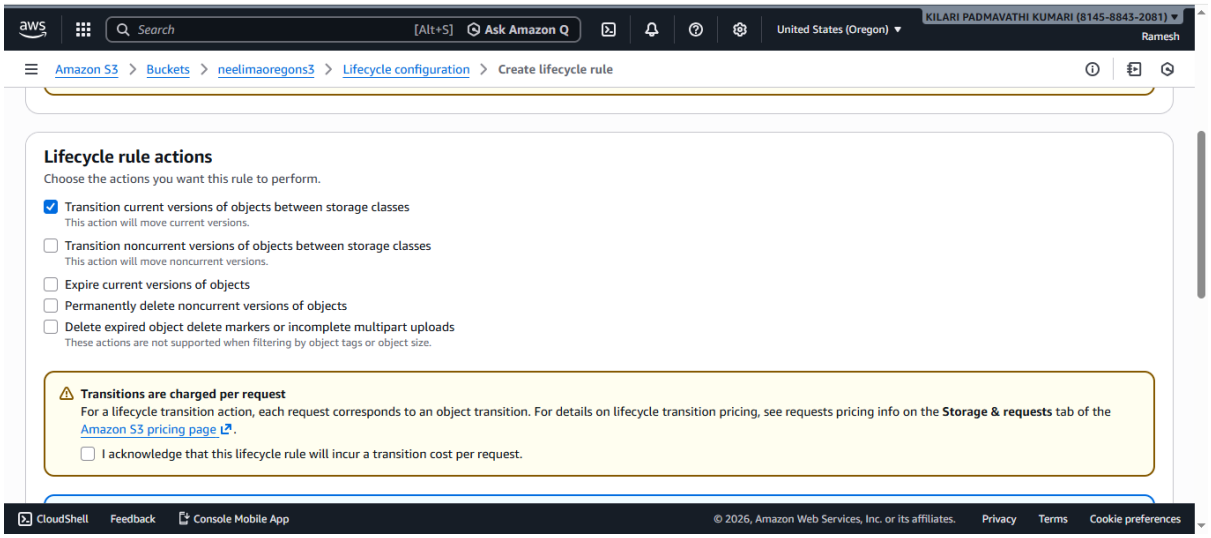
- To verify it other users can’t access bucket and list the objects.

S3-TASK1

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

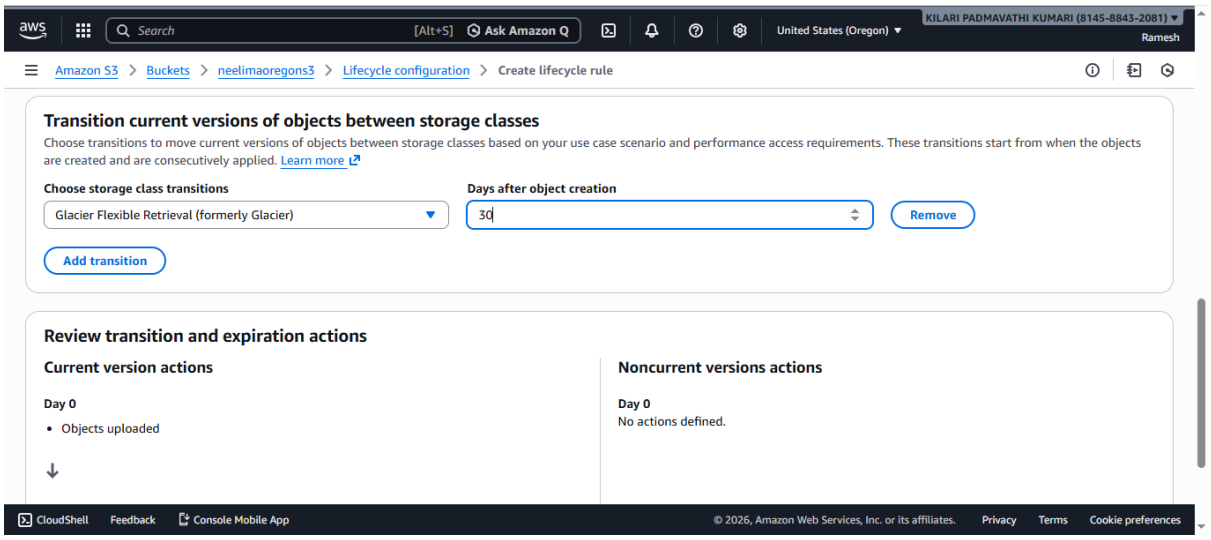


- Go to aws console and choose amazon s3
- Search for Buckets
- Choose bucket in which we want to make changes.
- After selecting bucket in that to go management tab
- Scroll and go to lifecycle configuration
- Click on create lifecycle rule.

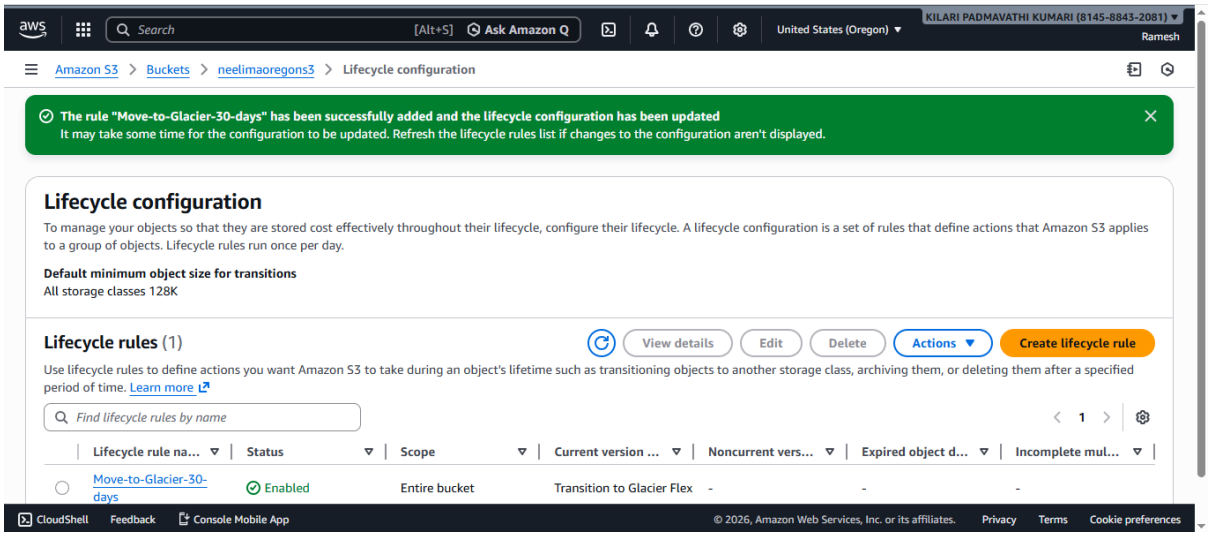


- In Lifecycle rule actions choose the actions we want this rule to perform
- Select Transition current versions of objects b/w storage classes.

S3-TASK1



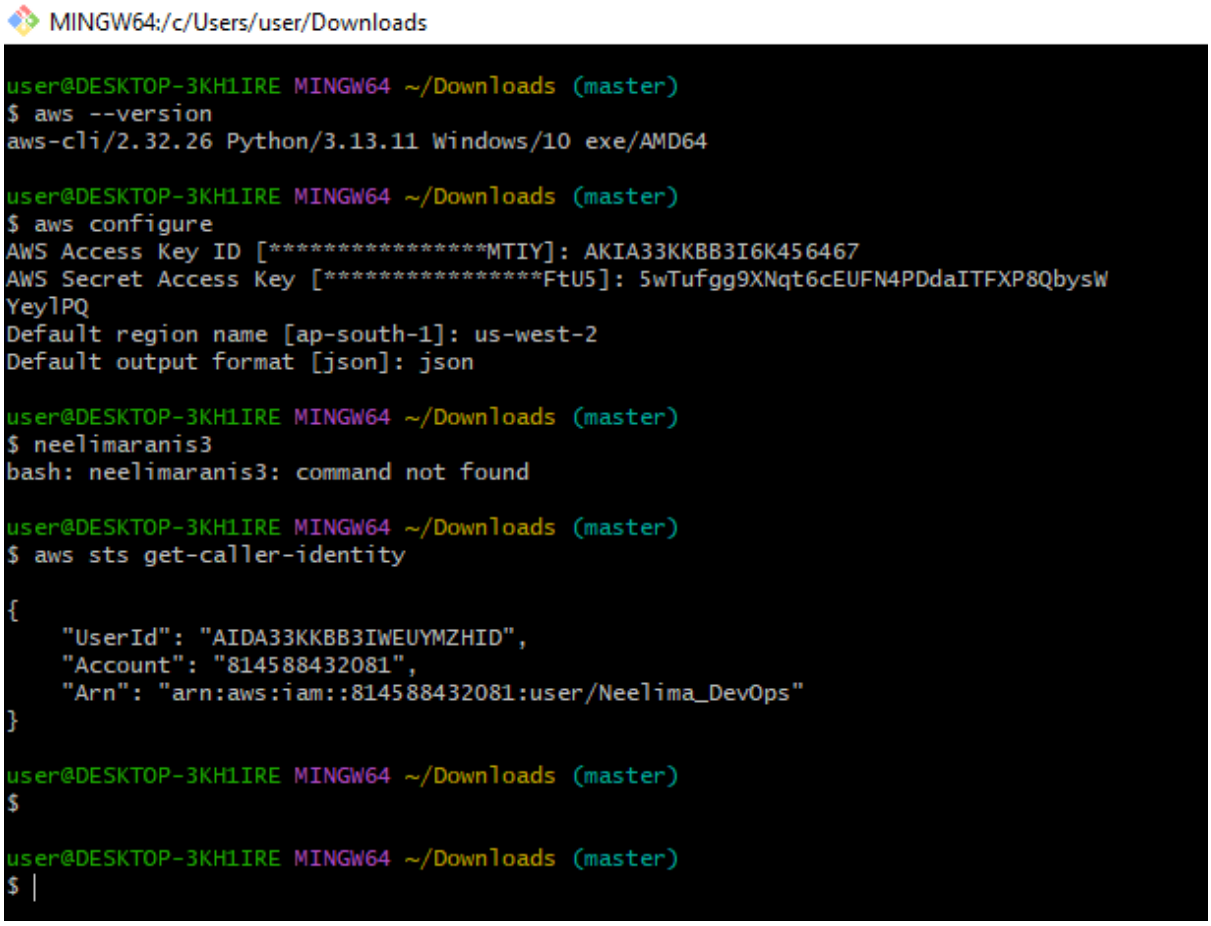
- In Transition current version of objects between storage classes
- Choose storage class transitions →Glacier Flexible Retrieval
- Days after object creations→30
- Click on create rule



- The above image shows Lifecycle configuration has been updated .

S3-TASK1

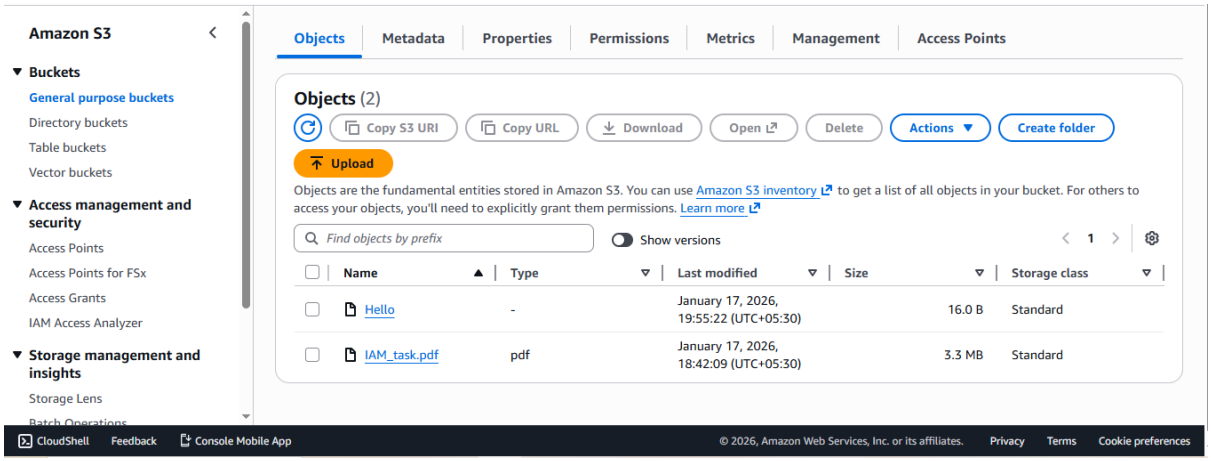
6. Push some objects to S3 using the AWS CLI.



- Open Git Bash check for **aws –version**
- Next configure aws with command **aws configure**
- After configure for details with command **Aws sts get-caller-identity**

- Create a File
- By using cp command we can push the objects into Buckets
  Command: aws s3 cp Hello s3://neelimaranis3/

S3-TASK1



- To verify Go to aws console
- Choose s3 and go to bucket were we have push the object
- Check the objects by clicking on object tab.

S3-TASK1

7. Write a Bash script to create an S3 bucket.

MINGW64:/c/Users/user/Downloads

```
#!/bin/bash
BUCKET_NAME="neelimaranis3-demo-$(date +%s)"
REGION="us-west-2"

echo "creating s3 bucket: $BUCKET_NAME IN $REGION"

aws s3api create-bucket --bucket $BUCKET_NAME --region $REGION --create-bucket-configuration LocationConstraint=$REGION

if [ $? -eq 0 ]; then
        echo "Bucket created successfully"
else
        echo "Bucket creation failed"
fi
```

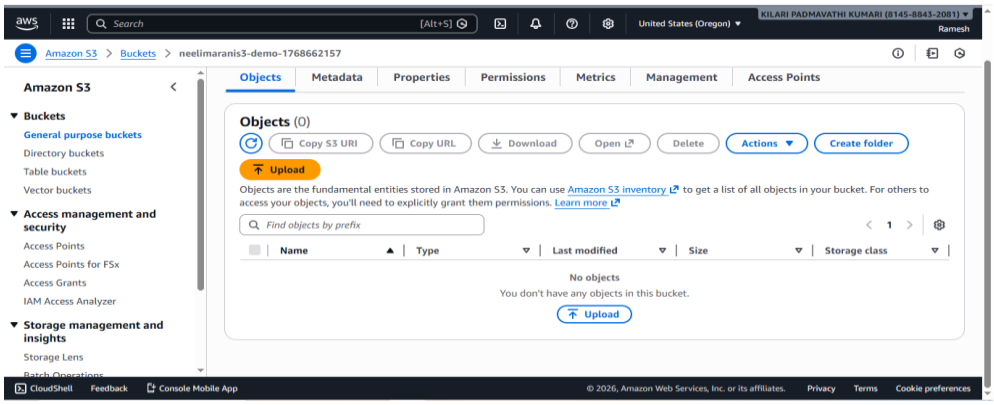
```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ vi s3bucket.sh

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ chmod 777 s3bucket.sh

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ./s3bucket.sh
creating s3 bucket: neelimaranis3-demo-1768662157 IN us-west-2
{
    "Location": "http://neelimaranis3-demo-1768662157.s3.amazonaws.com/",
    "BucketArn": "arn:aws:s3:::neelimaranis3-demo-1768662157"
}
Bucket created successfully

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$
```

- Write Bash script by using vi editor.
- Create a file and write script and save it.
- Give execute permission to file
- Run by command ./filename.



- To verify to s3 service and check buckets, in that we can see newly created bucket through CLI.

S3-TASK1

8. Upload a 1 GB file to S3 using the CLI.

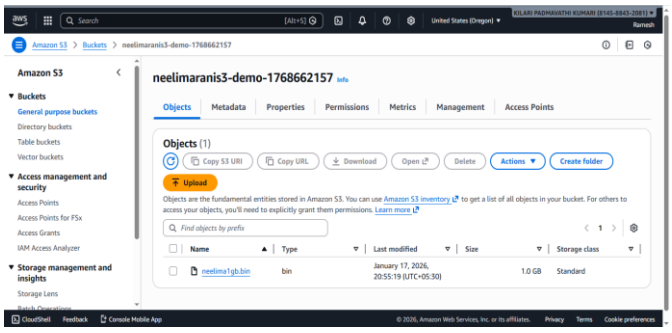
```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ dd if=/dev/zero of=neelima1gb.bin bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 3.14687 s, 341 MB/s

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ls -lh neelima1gb.bin
-rw-r--r-- 1 user 197121 1.0G Jan 17 20:53 neelima1gb.bin

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ aws s3 cp neelima1gb.bin s3://neelimaranis3-demo-1768662157/
upload: .\neelima1gb.bin to s3://neelimaranis3-demo-1768662157/neelima1gb.bin

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$
```

- To upload 1GB files we have command
  $ dd if=/dev/zero of=neelima1gb.bin bs=1M count=1024
- **dd**: The command-line utility used for converting and copying files.
- **if=/dev/zero**: Defines the input file (if). /dev/zero is a special file in Unix-like systems that provides as many null characters (zero-value bytes) as are read from it.
- **of=neelima1gb.bin**: Defines the output file (of). This is the name of the file that will be created.
- **bs=1M**: Sets the block size (bs) to 1 Megabyte. This tells dd to read/write 1MB at a time.
- **count=1024**: Specifies that the command should only copy 1024 blocks.
- Check with command ls -lh neelima1gb.bin
- **aws s3 cp neelima1gb**.bin s3://neelimaranis3-demo-1768662157/ →copy file to bucket.



- To verify, go to aws console choose buckets and select required bucket and check for objects tab and we can see file what we have create over the CLI.