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

**Using AWS Console:**

1. **Go to S3 → Create bucket.**
   * **Bucket name: my-static-website-bucket**
   * **Region: us-east-1 (or any)**
   * **Uncheck Block all public access (if hosting a website).**
2. **Click Create bucket.**

**Upload Objects:**

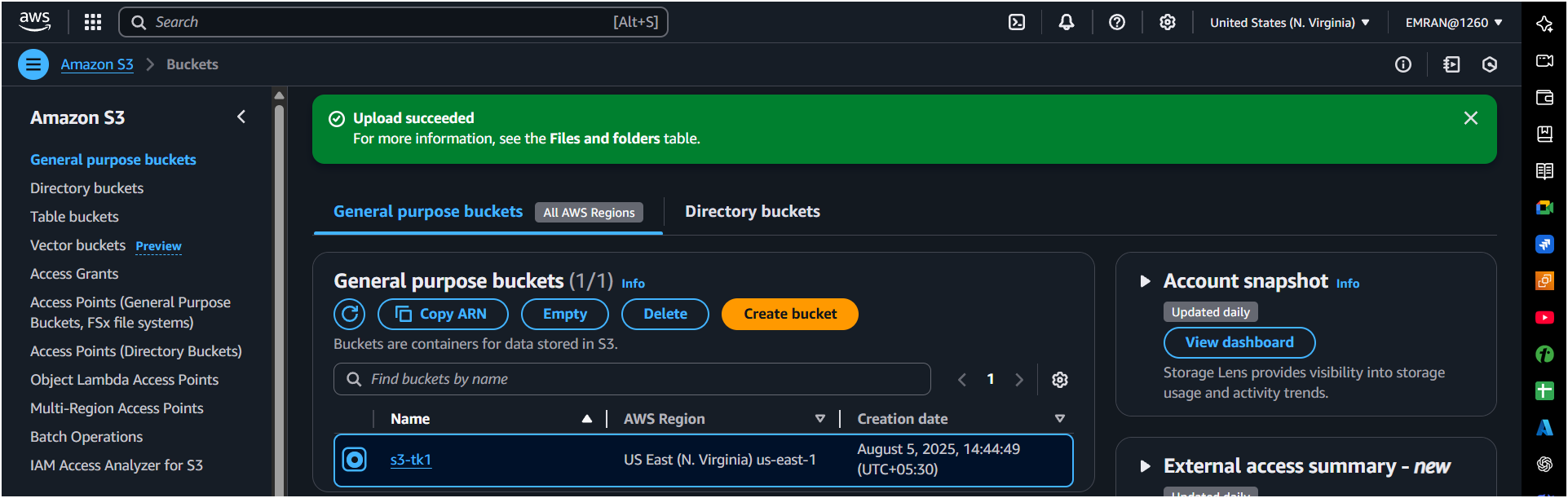
**AWS CLI Commands:**

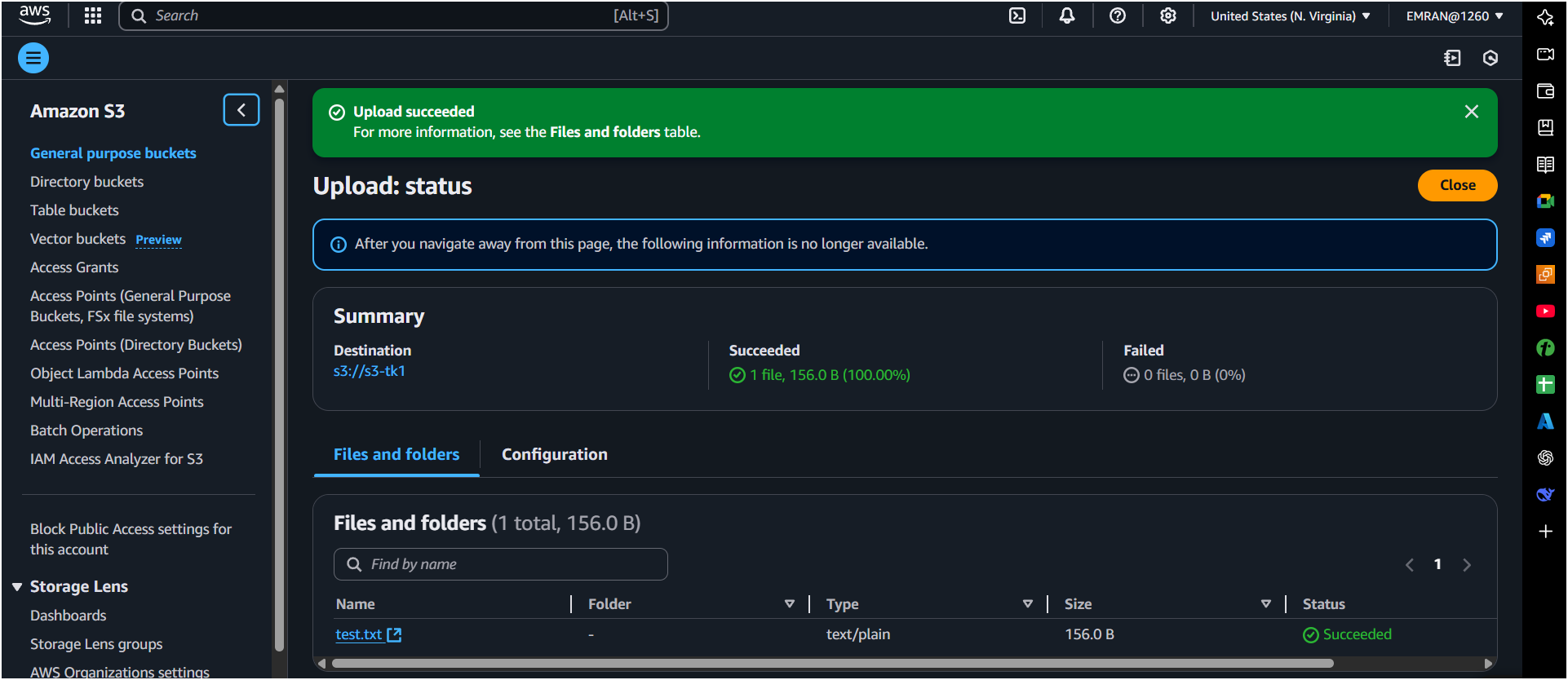
* **Go to your bucket → Upload → Add files → Upload sample files.**

**aws s3 mb s3://my-static-website-bucket --region us-east-1**

**aws s3 cp file1.txt s3://my-static-website-bucket/**

**aws s3 cp file2.txt s3://my-static-website-bucket/**





1. **Deploy static website in s3 bucket.**

**Steps:**

1. **Go to bucket → Properties → Static Website Hosting → Enable.**
   * **Hosting type: Host a static website.**
   * **Index document: index.html.**
2. **Upload index.html and error.html files.**
3. **Make objects public using Bucket Policy**

**{**

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

**"Statement": [**

**{**

**"Sid": "PublicReadGetObject",**

**"Effect": "Allow",**

**"Principal": "\*",**

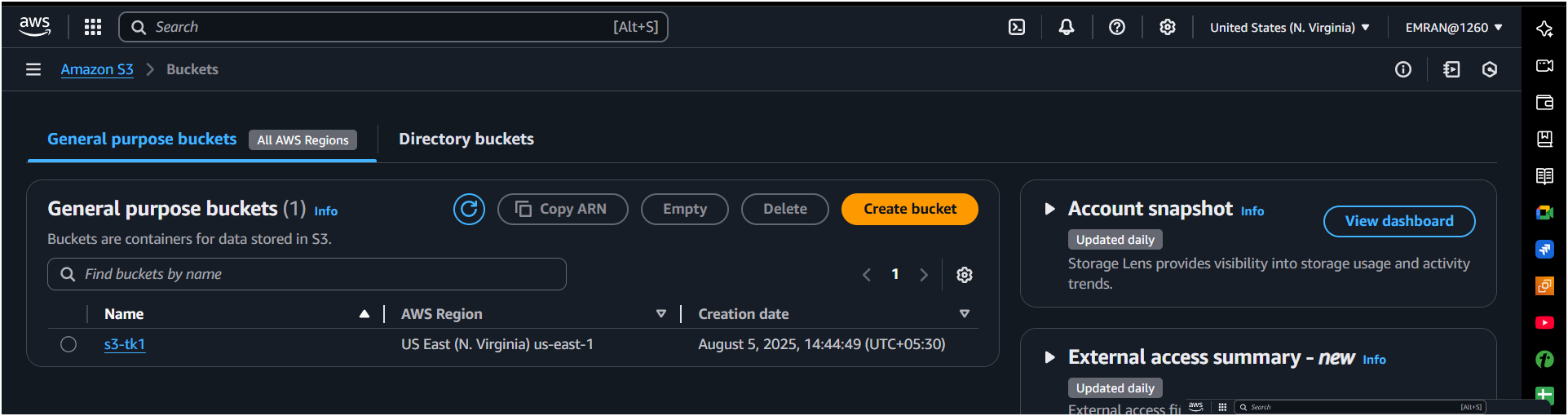
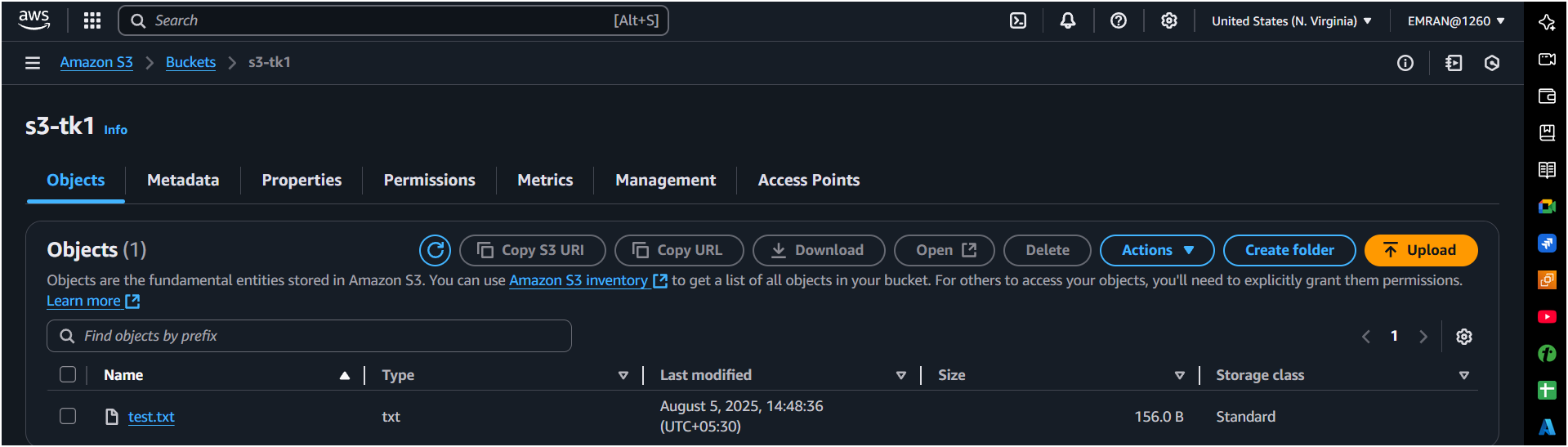
**"Action": "s3:GetObject",**

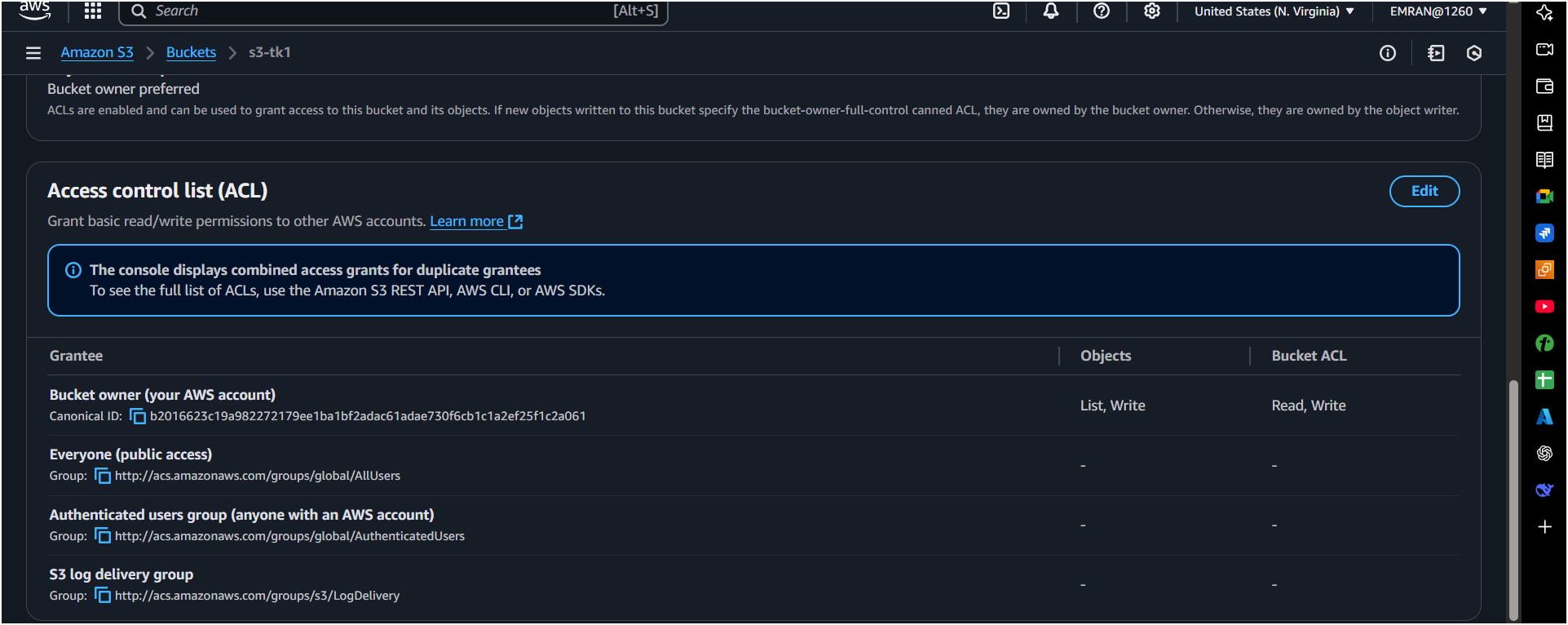
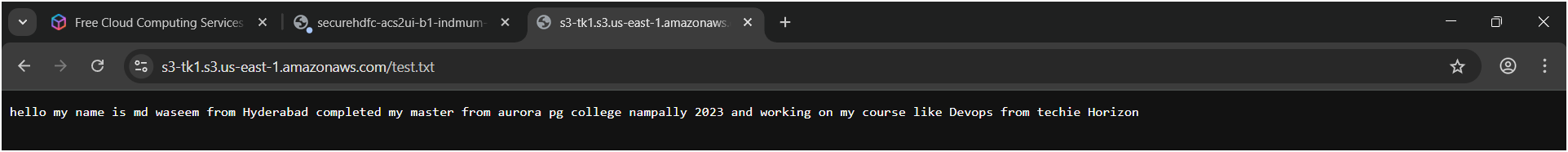
**"Resource": "arn:aws:s3:::my-static-website-bucket/\*"**

**}**

**]**

**}**

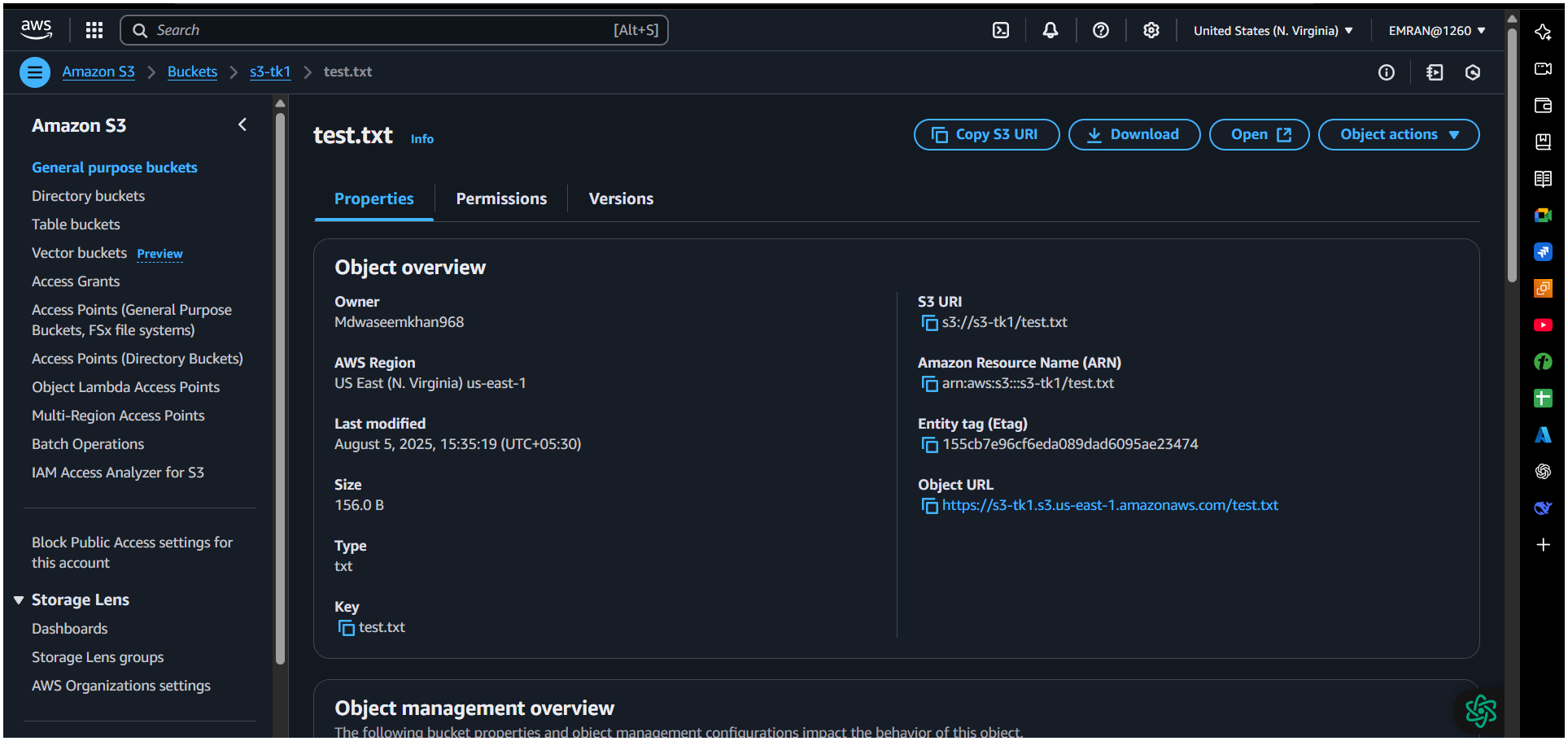
1. **Enable cross region replication on s3 buckets.**

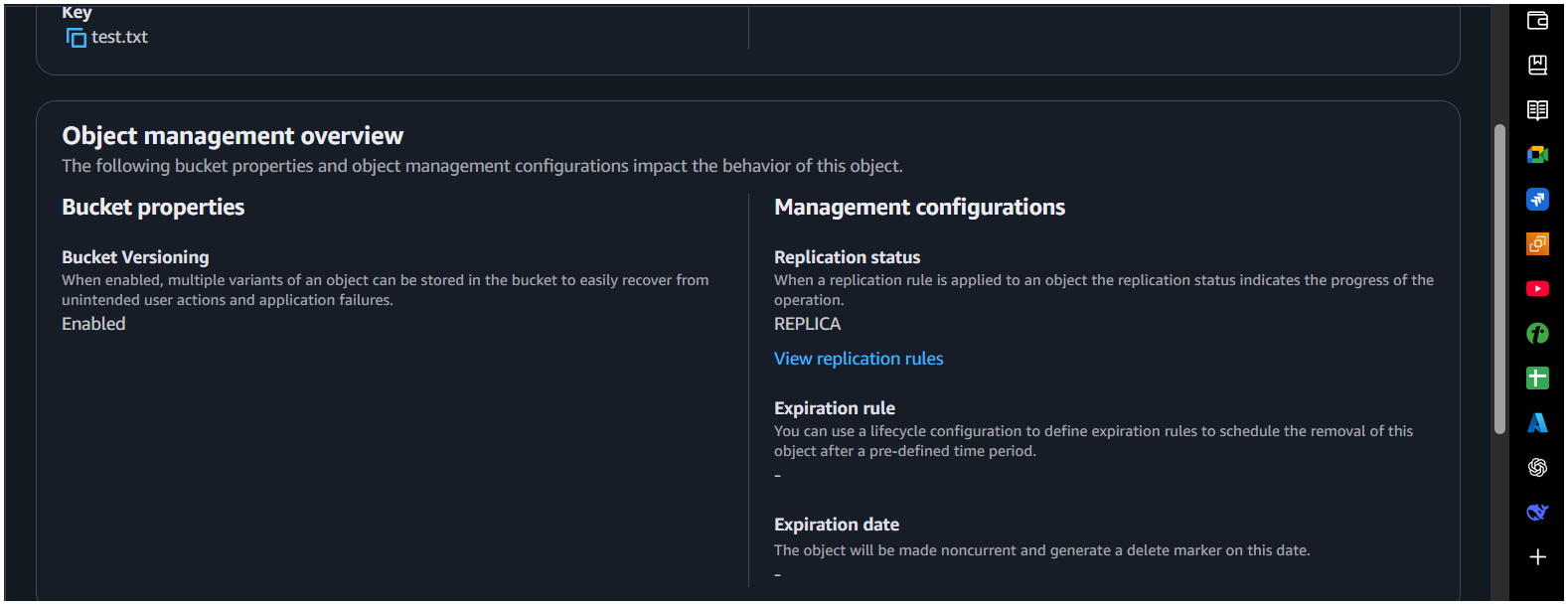
**Pre-requisites:**

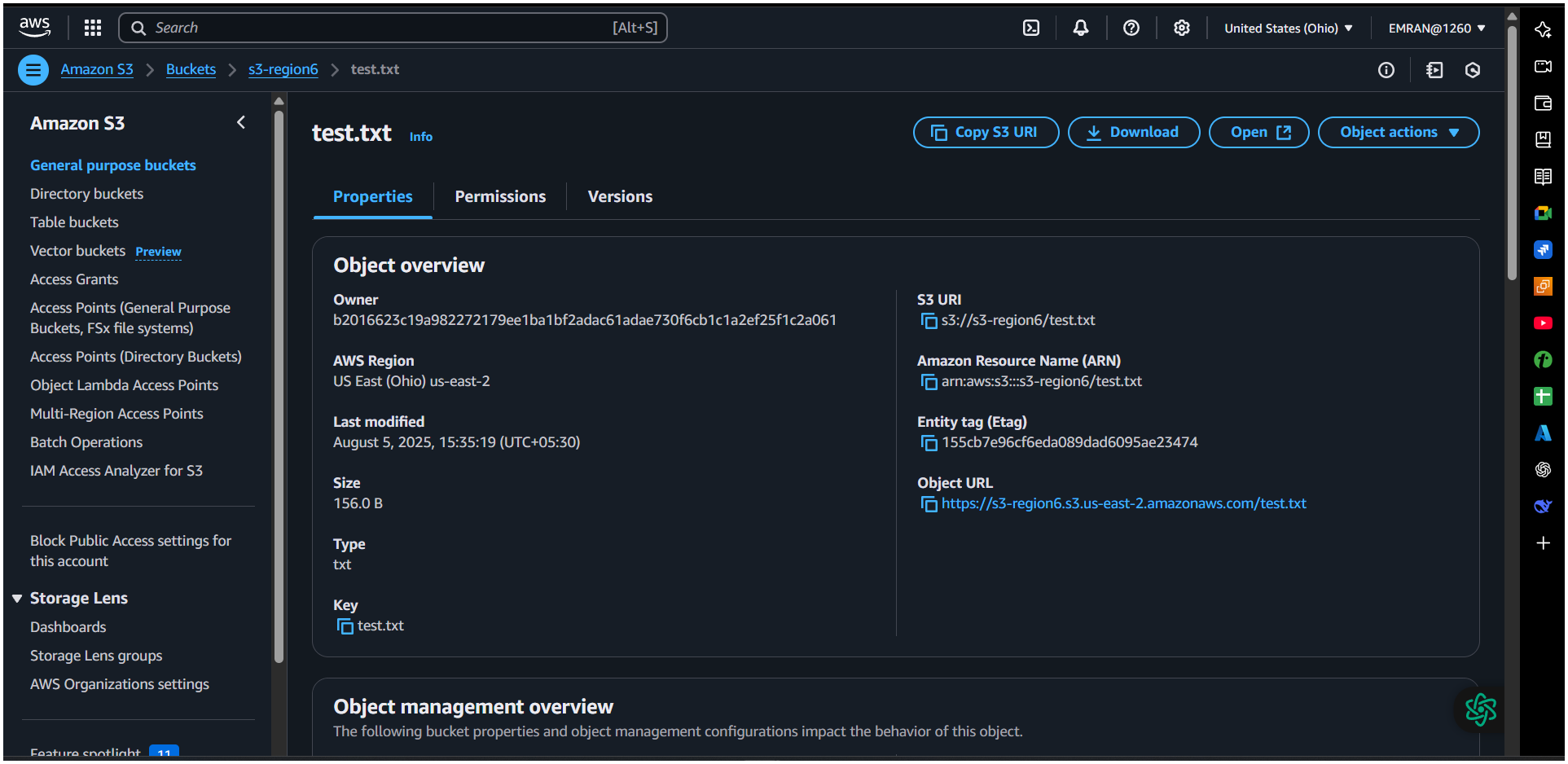
* **Versioning must be enabled on both source and destination buckets.**

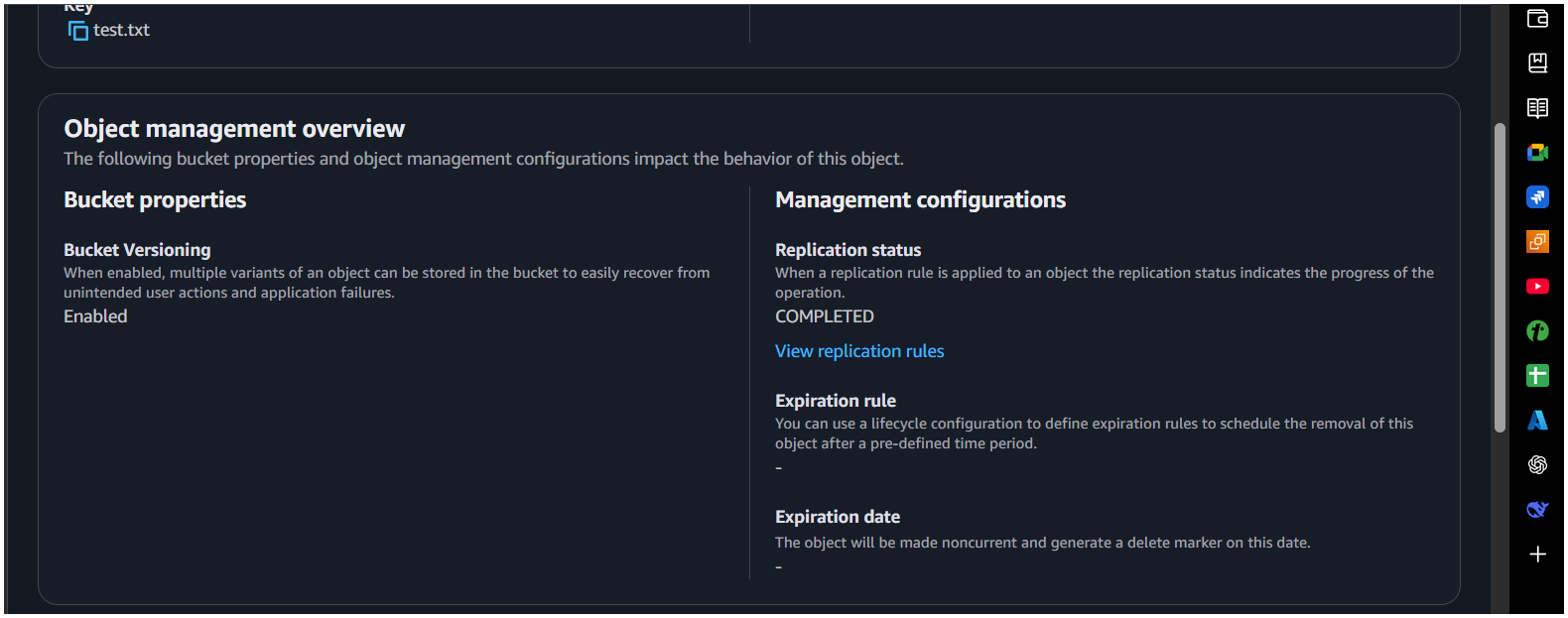
**Steps:**

1. **Enable Versioning:**
   * **In Source bucket and Destination bucket: Go to Properties → Versioning → Enable.**
2. **Go to Source Bucket → Management → Replication → Add rule:**
   * **Destination: Another bucket in different region.**
3. **Create or choose IAM role for replication.**
4. **Save.**









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

**Go to your s3 and create permissions bucket policy,  
Assume Admin IAM user ARN:  
arn:aws:iam::123456789012:user/Admin**

**Bucket Policy:**

{

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

"Statement": [

{

"Sid": "DenyAllUsersExceptAdmin",

"Effect": "Deny",

"Principal": "\*",

"Action": "s3:\*",

"Resource": [

"arn:aws:s3:::s3-tk1",

"arn:aws:s3:::s3-tk1/\*"

],

"Condition": {

"StringNotEquals": {

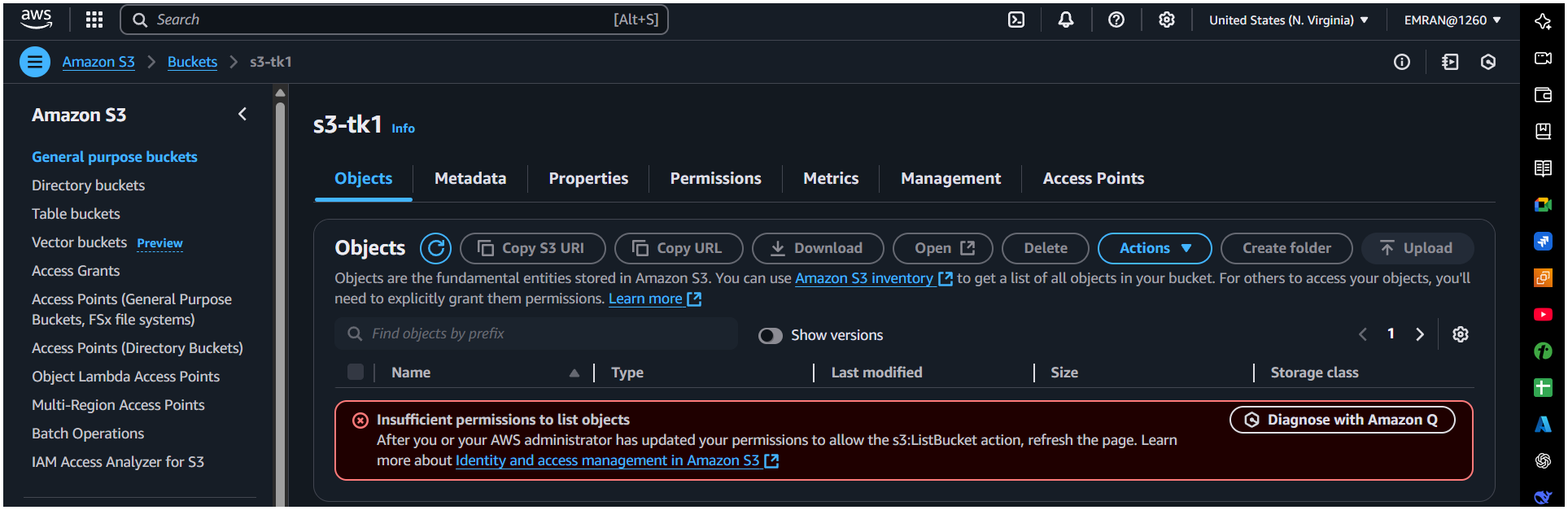
"aws:PrincipalArn": "arn:aws:iam::520506377811:user/Admin"

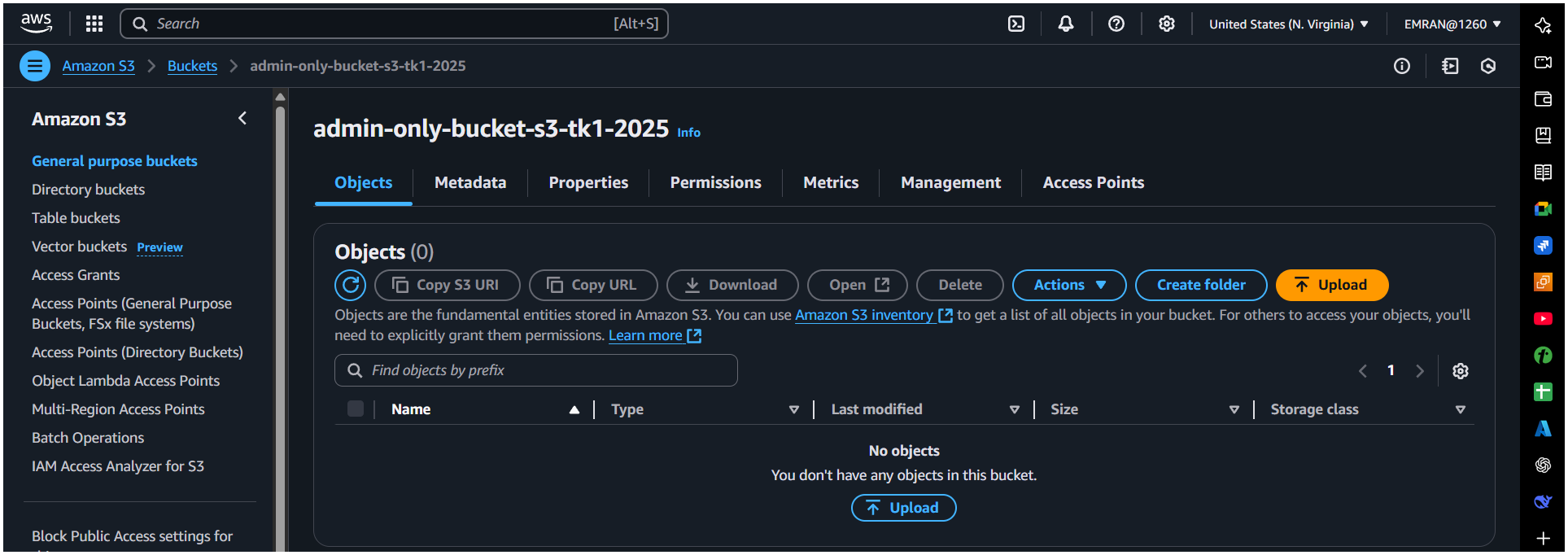
}

}

}

]

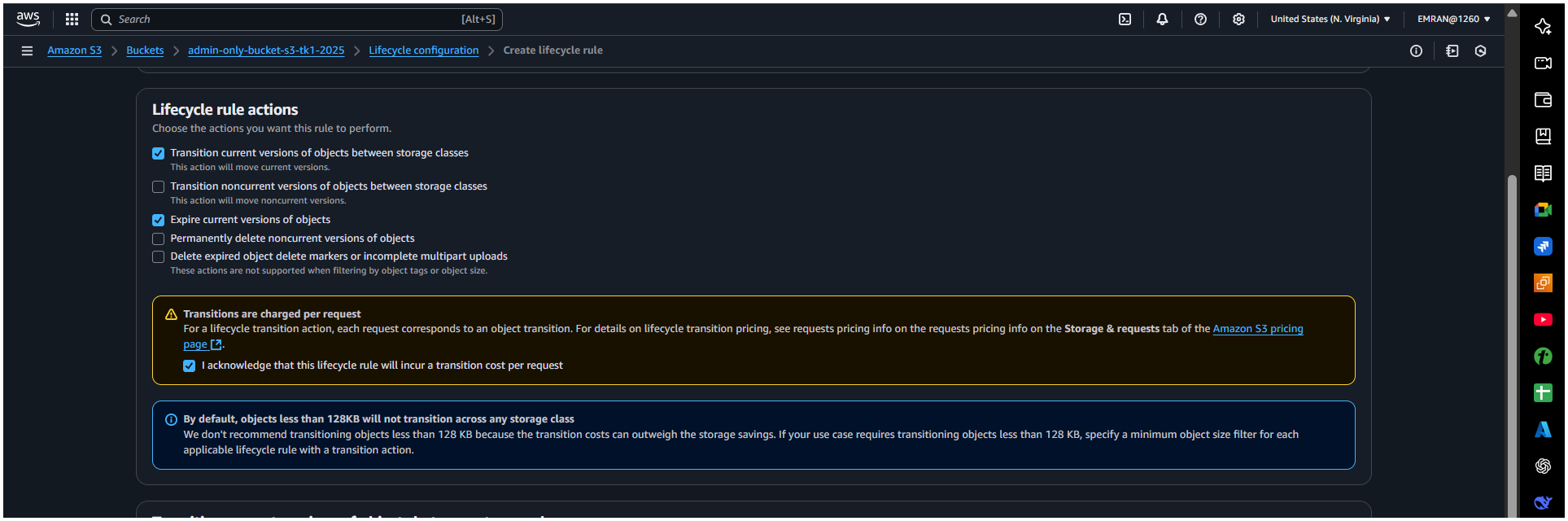
}

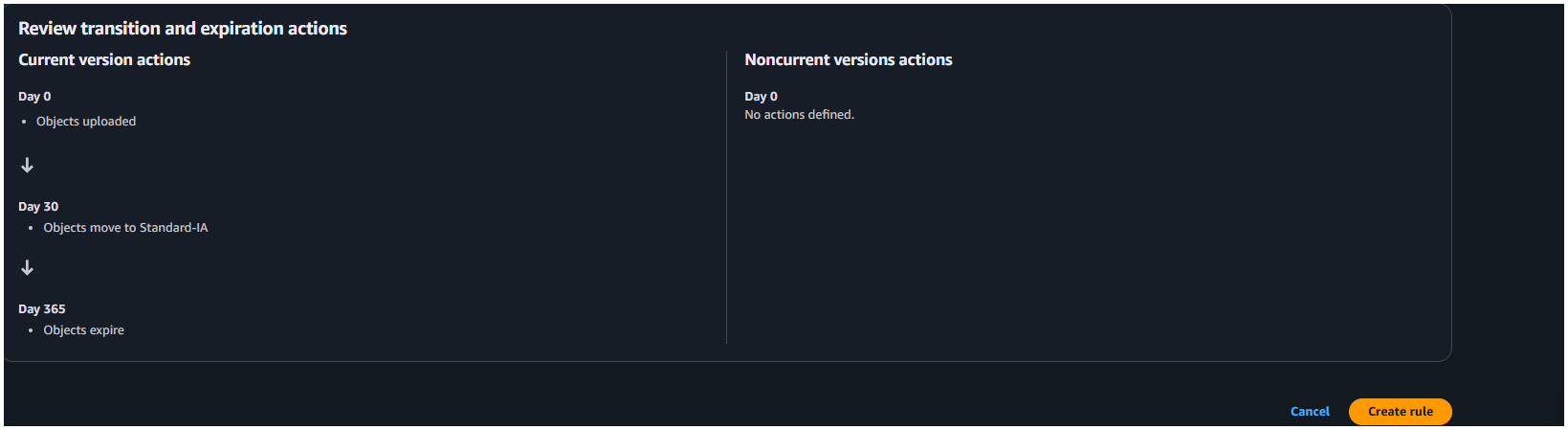


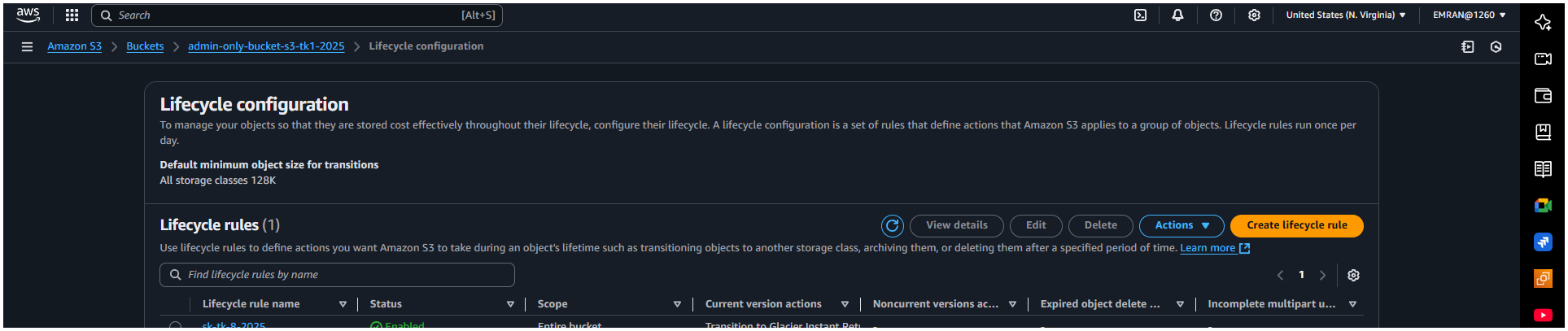
1. Setup lifecycle policies to automatically transition or delete objects based on specific criteria.

**Steps:**

1. Go to bucket → **Management → Lifecycle Rules → Create rule**.
2. Set **Transition**:
   * After **30 days**, move to **Glacier**.
3. Set **Expiration**:
   * After **365 days**, delete objects.



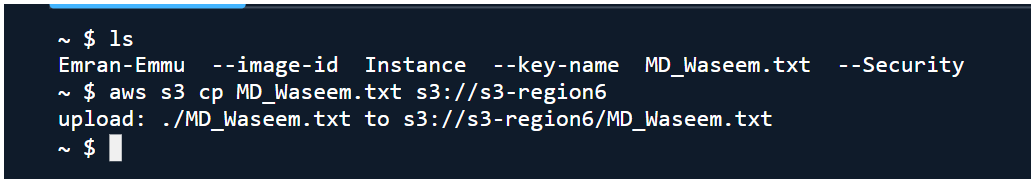


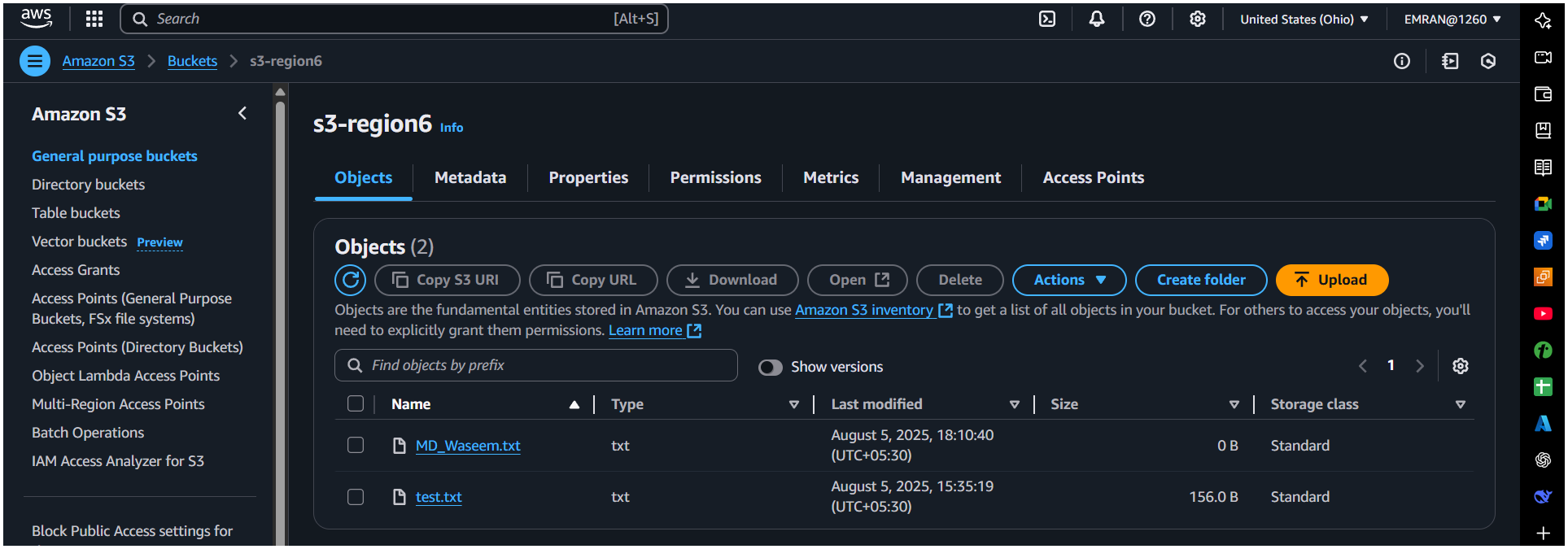


1. Push some objects in s3 using AWS CLI.

Upload multiple files:

aws s3 cp /path/to/local/dir s3://my-static-website-bucket/ --recursive





7) Write a bash script to create s3 bucket.  
 #!/bin/bash

read -p "Enter bucket name: " bucket

read -p "Enter region: " region

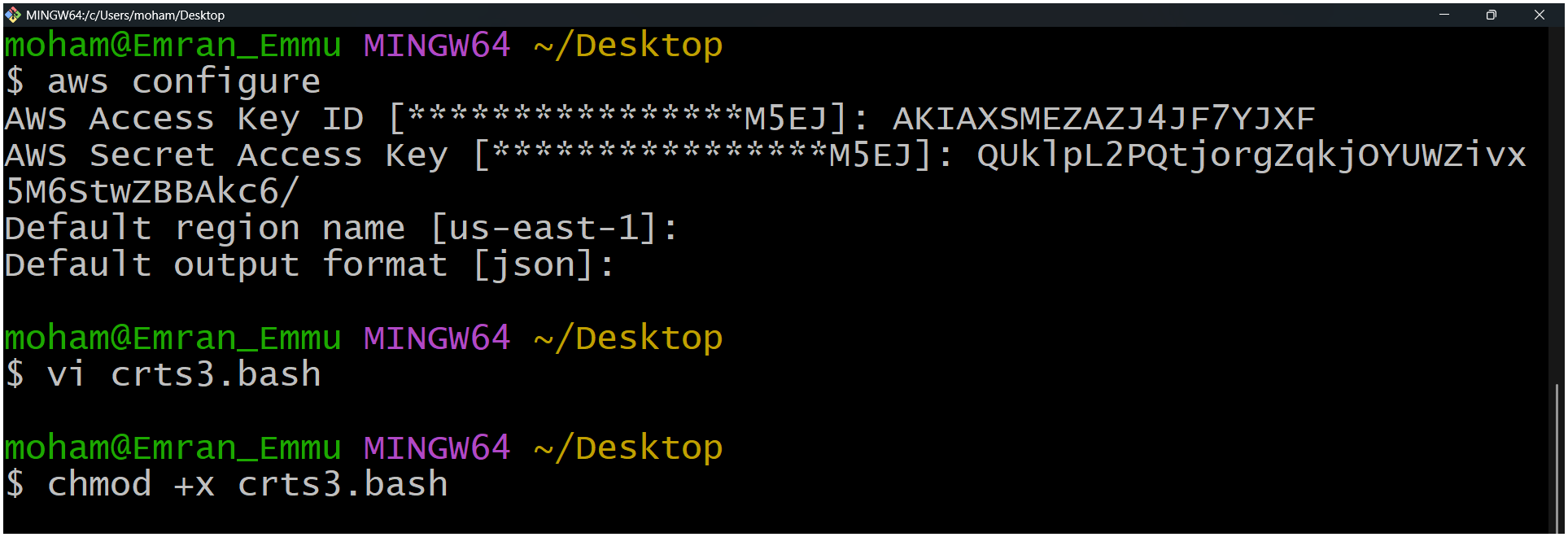
aws s3 mb s3://$bucket --region $region

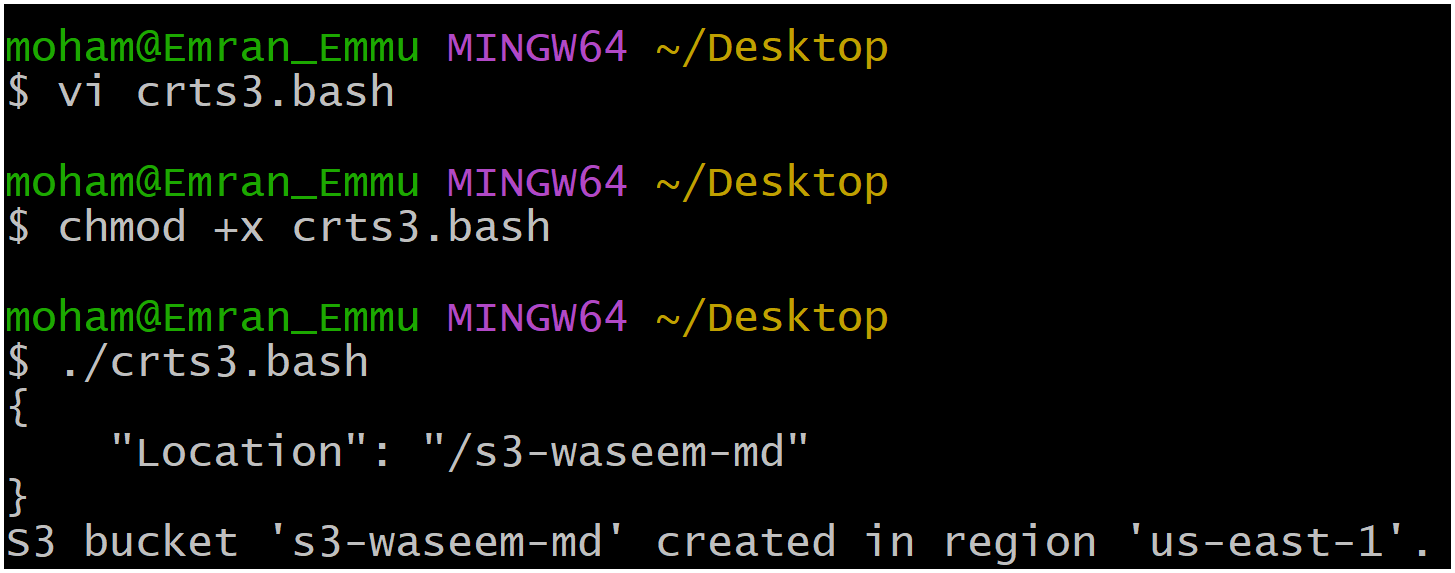
if [ $? -eq 0 ]; then

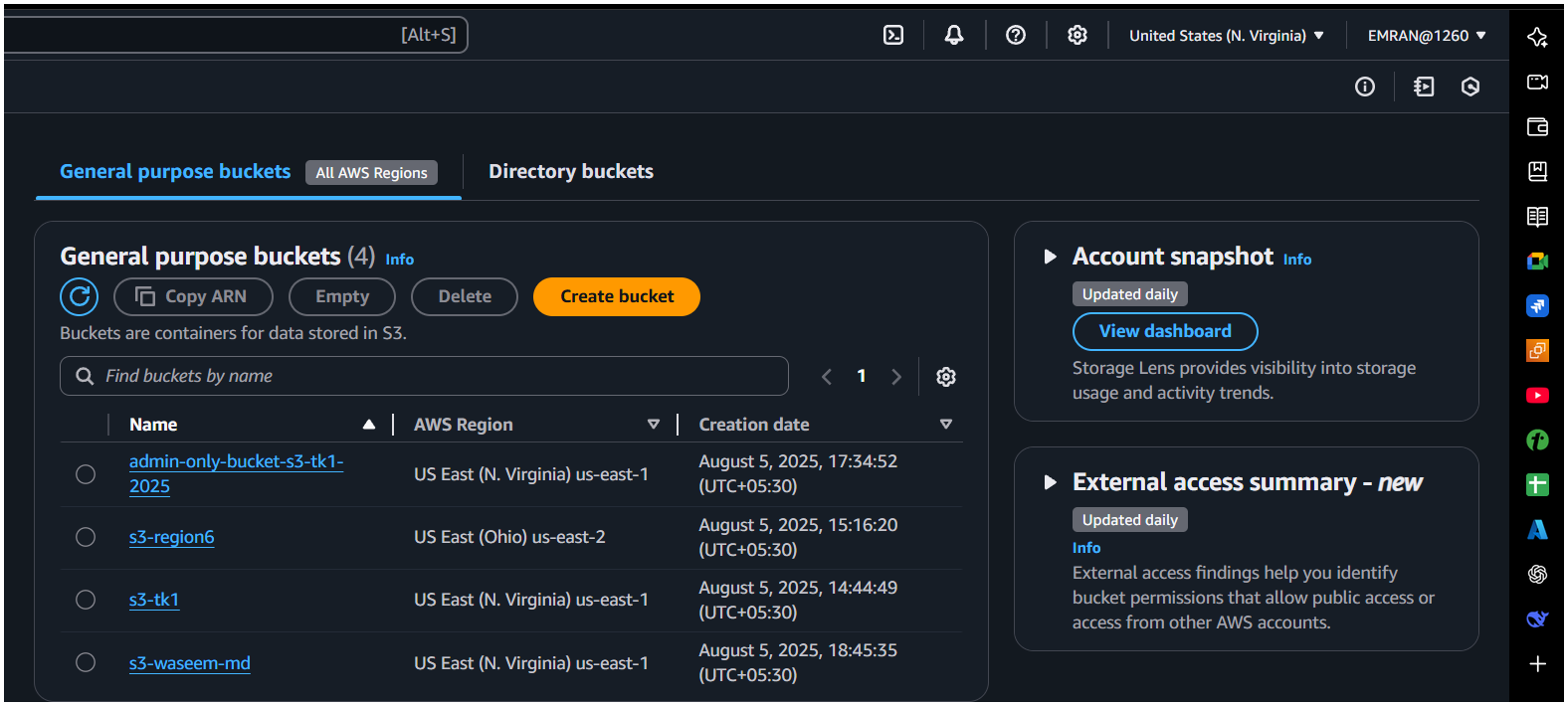
echo "Bucket $bucket created successfully in $region."

else

echo "Failed to create bucket."

fi





Make executable:

chmod +x create\_bucket.sh

./create\_bucket.sh

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

1. Create a **1GB file**:

fallocate -l 1G bigfile.txt # Linux  
OR

1. dd if=/dev/zero of=bigfile.txt bs=1M count=1024  
   Upload to S3:

aws s3 cp bigfile.txt s3://my-static-website-bucket/

