**AWS S3 bucket**

An **S3 bucket** in **Amazon Web Services (AWS)** is a storage container in **Amazon S3 (Simple Storage Service)**. It’s designed to store and manage **any amount of data (massive unstructured data)** — from files like documents, images, videos, audios and backups to large-scale big data or machine learning datasets.

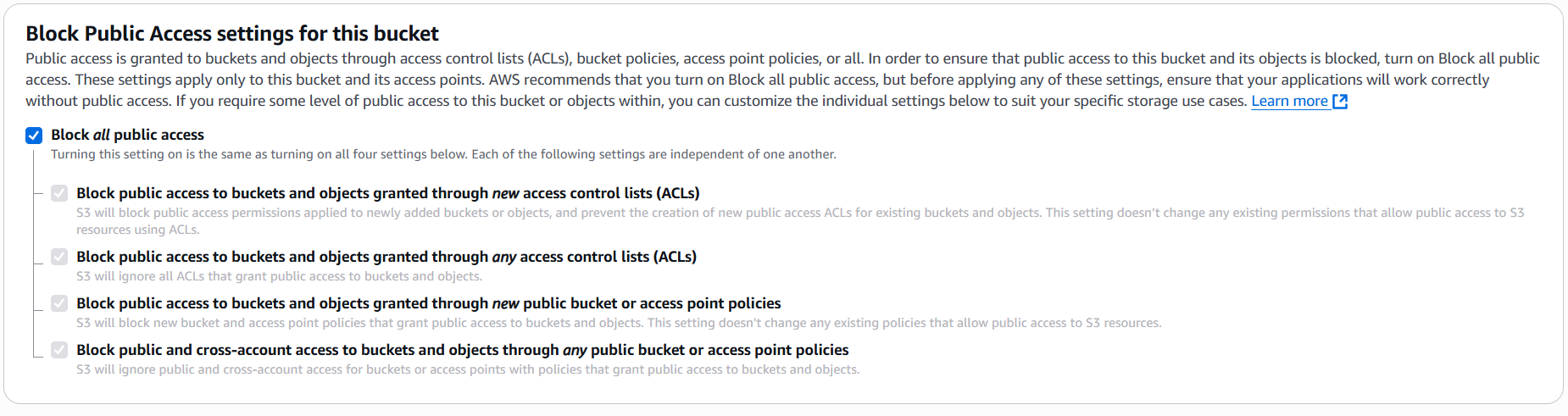
**Amazon S3 Bucket Features:**

1. **Unlimited Storage Capacity**
   * Stores any amount of data (from kilobytes to petabytes).
2. **High Durability**
   * 99.999999999% (11 9s) durability by storing data across multiple Availability Zones.
3. **High Availability**
   * Designed for 99.99% availability with automatic redundancy.
4. **Scalability**
   * Automatically scales storage as your data grows.
5. **Multiple Storage Classes**
   * Choose from S3 Standard, S3 Intelligent-Tiering, S3 Infrequent Access, S3 Glacier, etc., based on access needs and cost.
6. **Security and Access Control**
   * Supports IAM policies, bucket policies, ACLs, and encryption (SSE, KMS).
7. **Data Encryption**
   * Encrypt data at rest (SSE-S3, SSE-KMS, SSE-C) and in transit (HTTPS/SSL).
8. **Versioning**
   * Maintain multiple versions of an object to protect against accidental deletion/overwrite.
9. **Lifecycle Management**
   * Automatically transition objects between storage classes or delete them after a specified time.
10. **Static Website Hosting**
    * Host static websites directly from S3 with custom domains and routing.
11. **Cross-Region Replication (CRR)**
    * Automatically replicates data to another AWS region for disaster recovery or latency reduction.
12. **Same-Region Replication (SRR)**
    * Replicates data within the same region for compliance or backup purposes.
13. **Object Lock**
    * Protects objects from deletion/modification for compliance (WORM - Write Once, Read Many).
14. **Logging and Monitoring**
    * Integrates with AWS CloudTrail and CloudWatch for access logs and activity monitoring.
15. **Storage capacity**
    * We can store unlimited data, but the maximum storage capacity of a single file/object is 5TB.
16. **Tagging and Metadata**
    * Add custom tags and metadata to manage and organize objects.
17. **Unique naming**
    * You have to give the storage name as Global unique.
18. **Data Consistency**
    * Strong read-after-write and list consistency across all S3 storage classes.

Let’s work with the AWS S3 Bucket practically:

**Step1:** Create an AWS S3 bucker.

**Note:** while creating an AWS S3 bucket the public access is disabled.



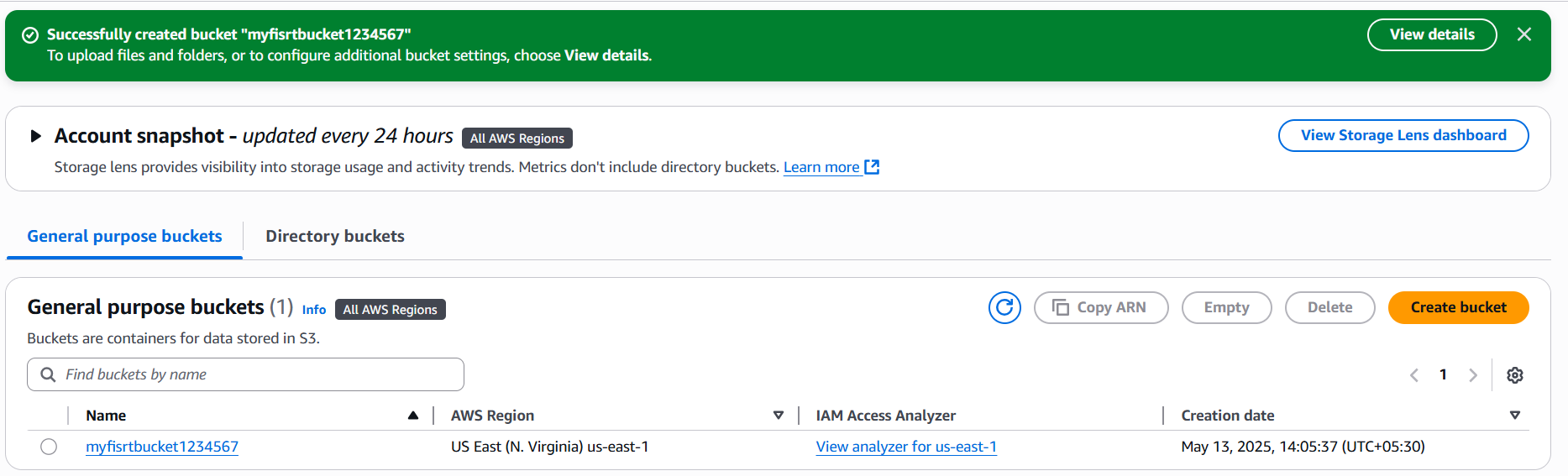


Fig: AWS S3 bucket (myfirstbucket1234567).

**Step2:** Now upload any file in the S3 bucket.

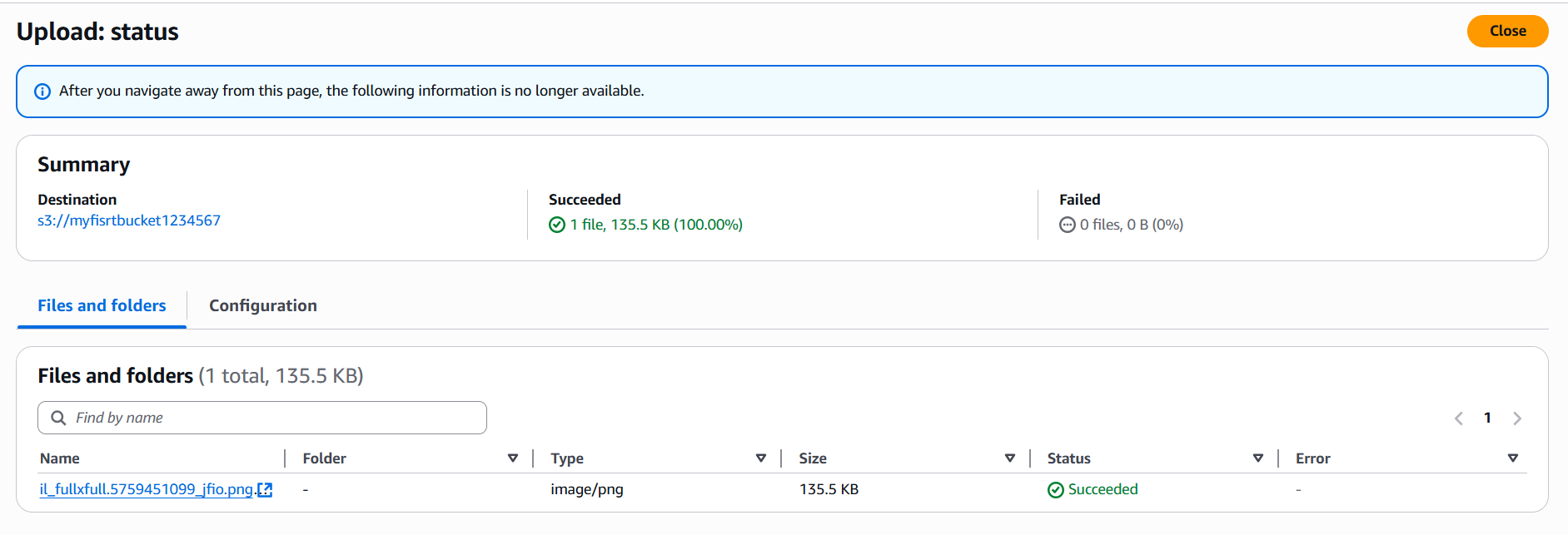


Fig: An image file is uploaded.

**Step3:** Copy the image URL and browse it.

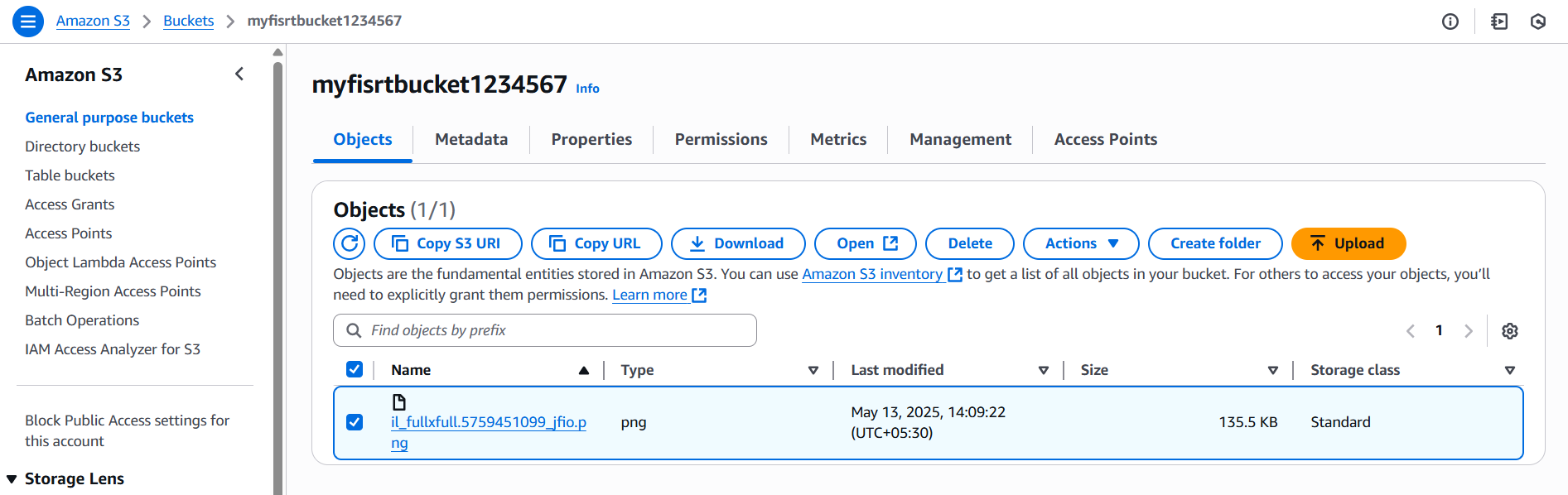


Fig: Copy file (image) URL.

**URL:** <https://myfisrtbucket1234567.s3.us-east-1.amazonaws.com/il_fullxfull.5759451099_jfio.png>

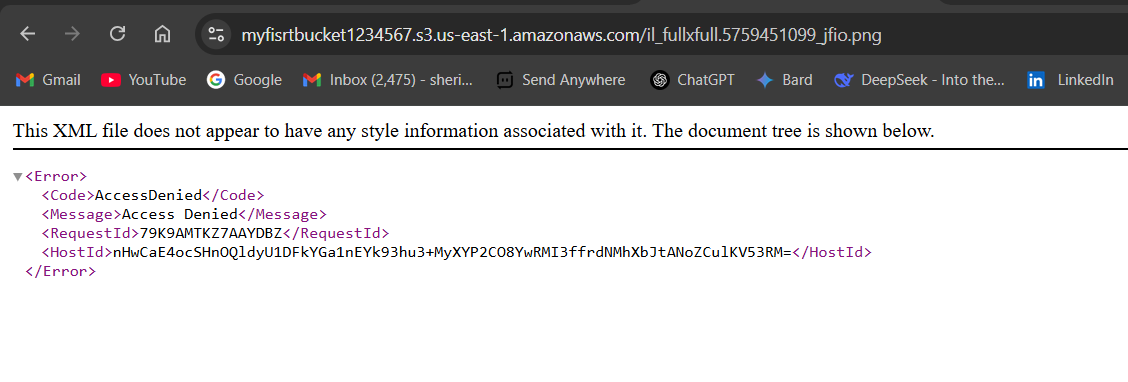


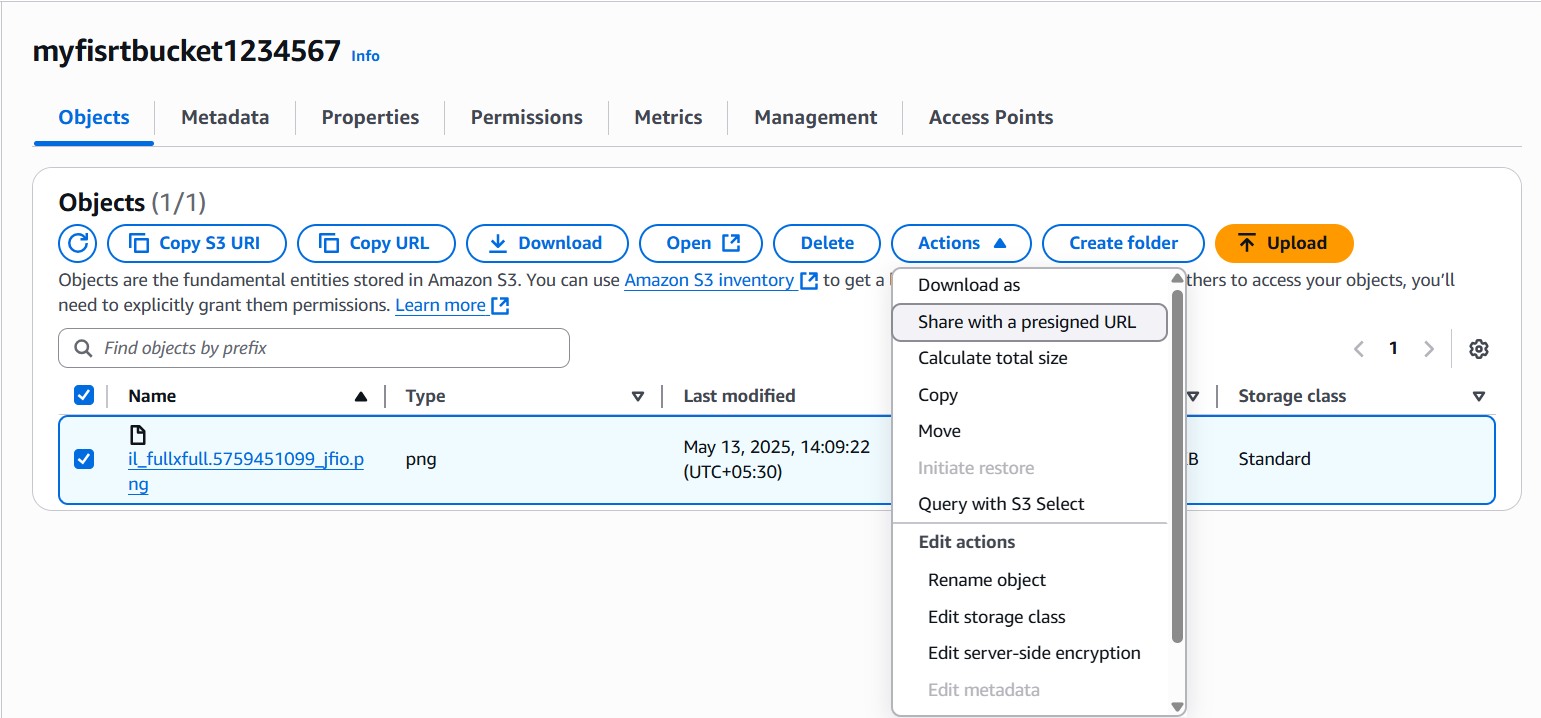
Fig: File does not appeared.

**Viewing/Browsing of Image/File from Private S3 bucket:**

The image is does not appeared, it is because our S3 bucket is not in public access, it is in private access.

Even though we can access (View) the image which is uploaded in a private S3 bucket, by using the “**presigned URL”**

By using **presigned URL** we can give the access to view the file to a **specific time spam**.



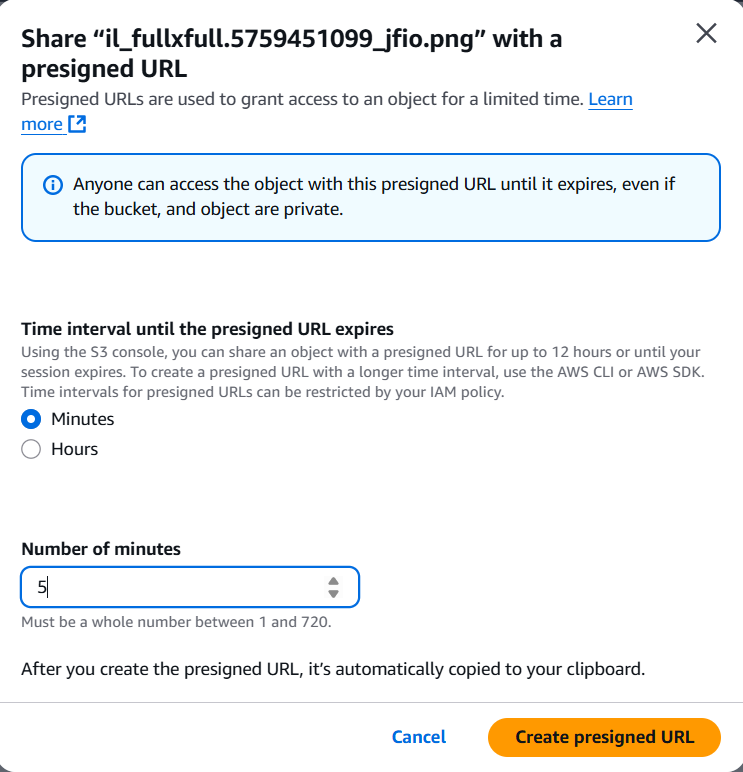


Fig: Share with Presigned URL.

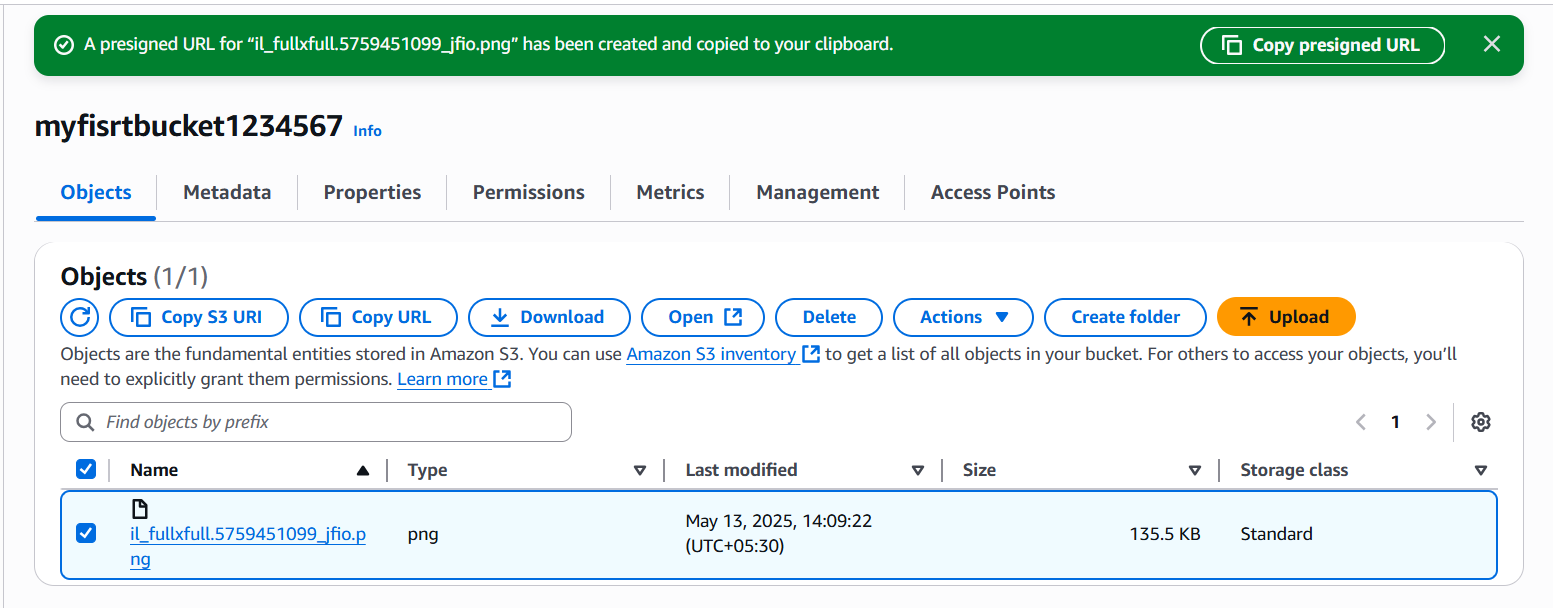


Fig: Copy Presigned URL.

Copy the Presigned URL and browse it, then we can able to view the image or file or object, which is uploaded in a private S3 bucket.

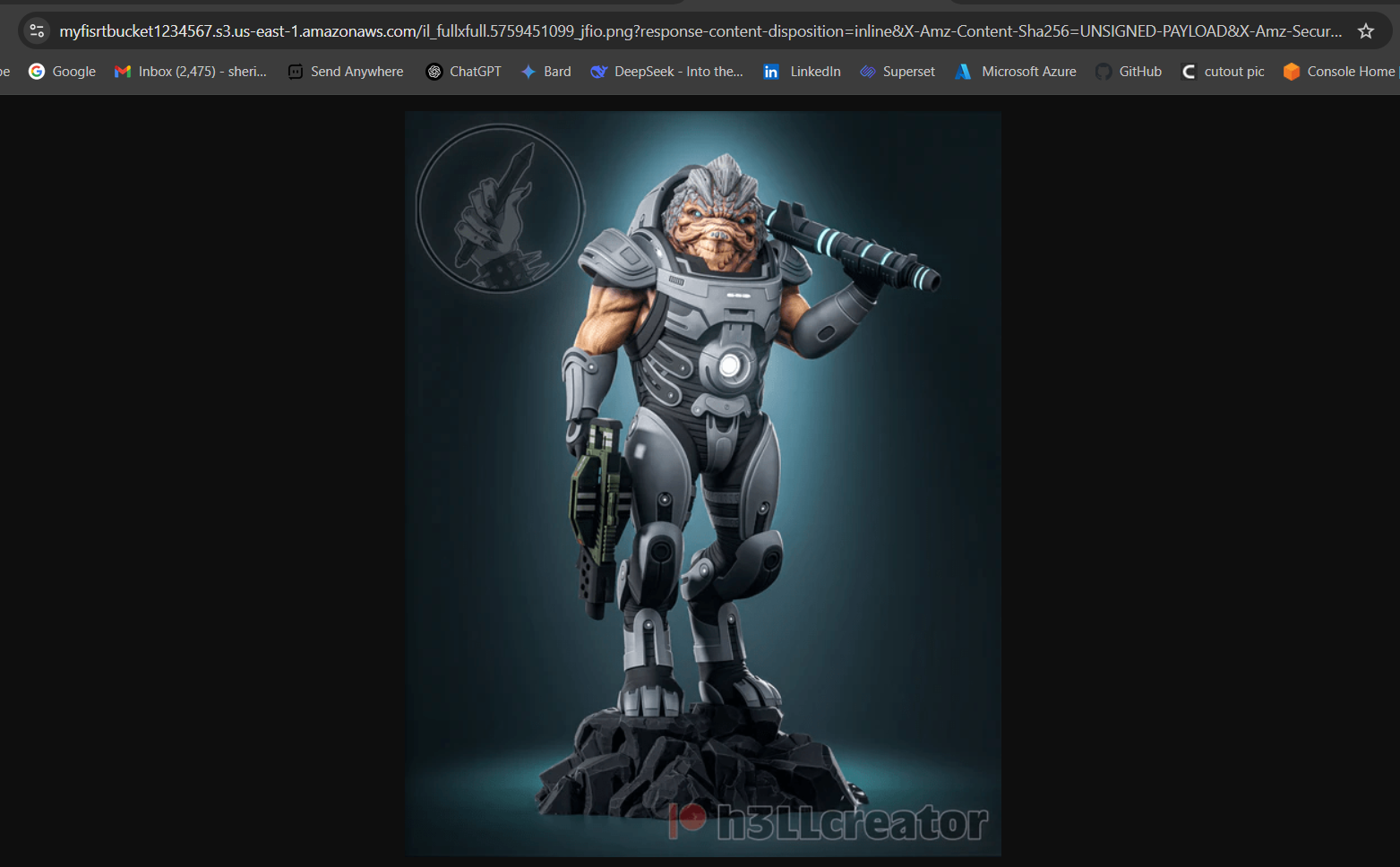


Fig: Able to View the image.

**Downloading of files/image form the Private S3 bucket using AWS CLI:**

The other way to view the image/file is by downloading of image/file using AWS CLI.

For that we have to install the AWS CLI from any browser.

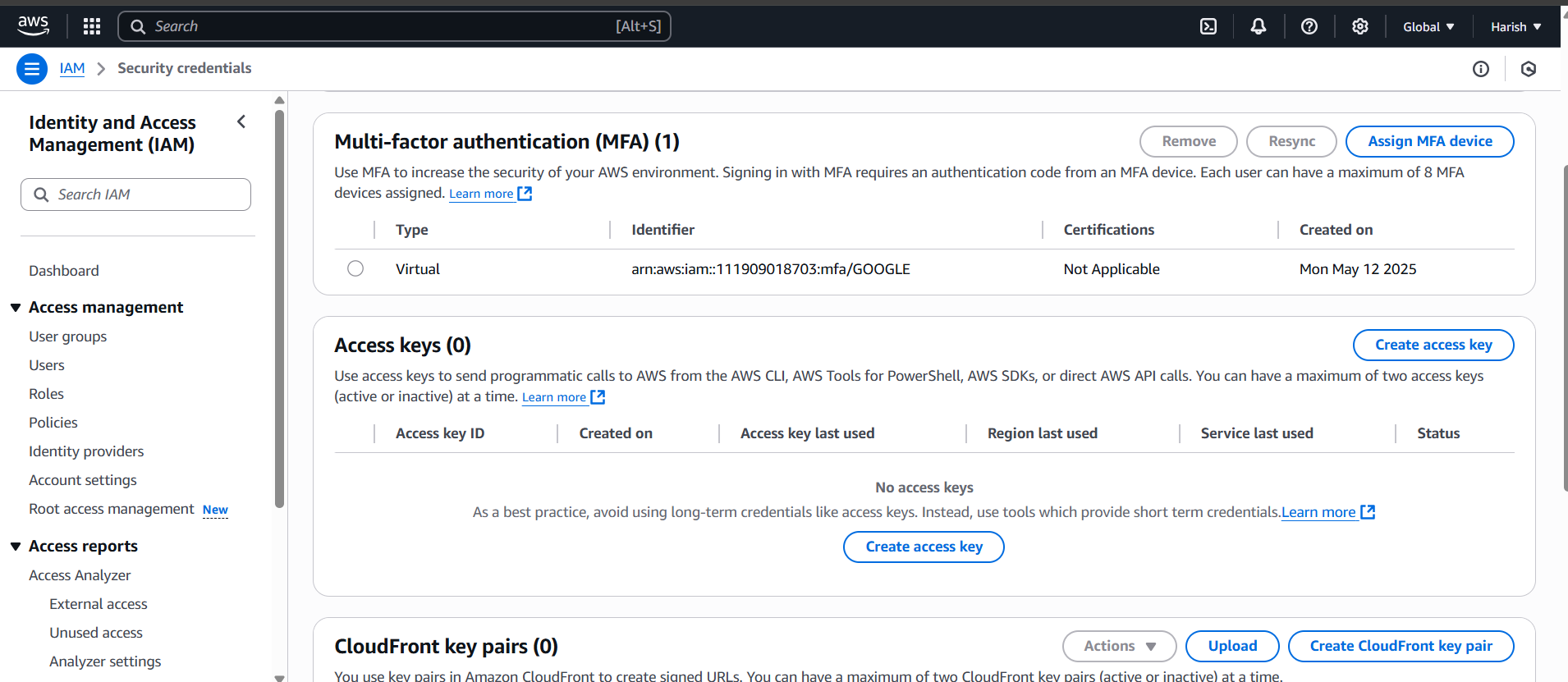
**Link:** https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

To connect the AWS CLI with the AWS portal we use the command

**Command:** aws configure.

Now we need Access key and Secrete access key for authentication with the AWS portal.

Go To 🡪User (Harish) 🡪Security credential 🡪create access key.



**Fig:** Create access keys.

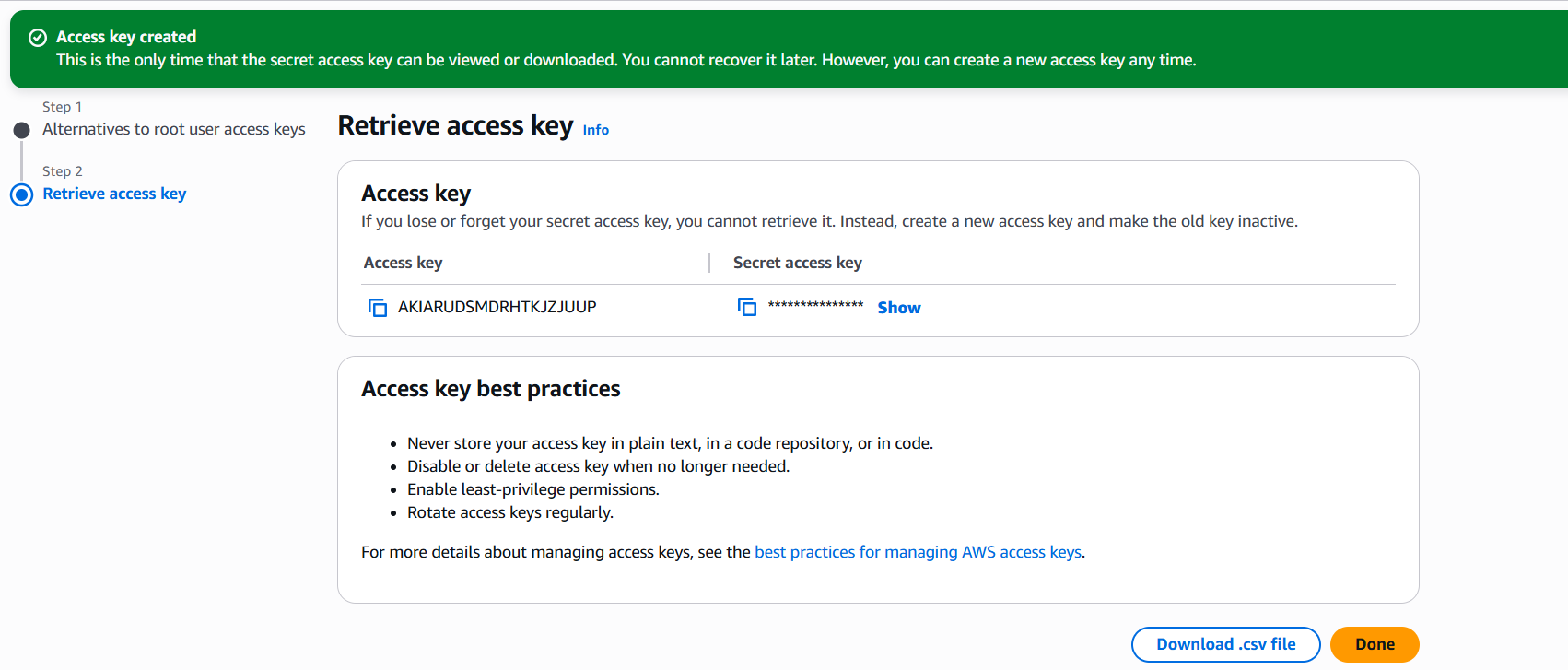


Fig: Access and secrete access keys.

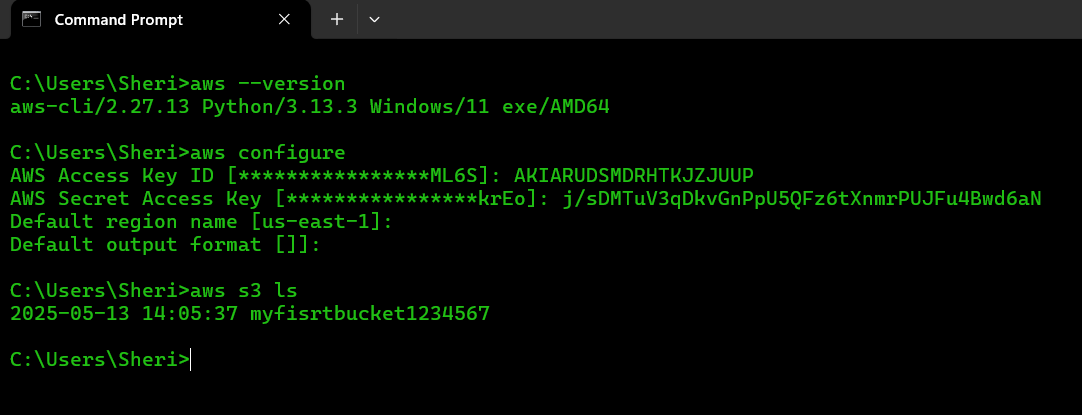


Fig: Successfully connect to the AWS portal.

**Note:** No need to enter the Default region & Default output format while connecting to the AWS portal.

The command to download the image from the private S3 bucket is

**Command:** aws s3 cp s3://your-bucket-name/path/to/image.jpg local/path/to/save/image.jpg

**EX:** aws s3 cp s3://my-photo-bucket/images/dog.jpg **.**

**Bucket name:** my-photo-bucket

**Object key**: images/dog.jpg

This downloads dog.jpg to your current working directory (.)

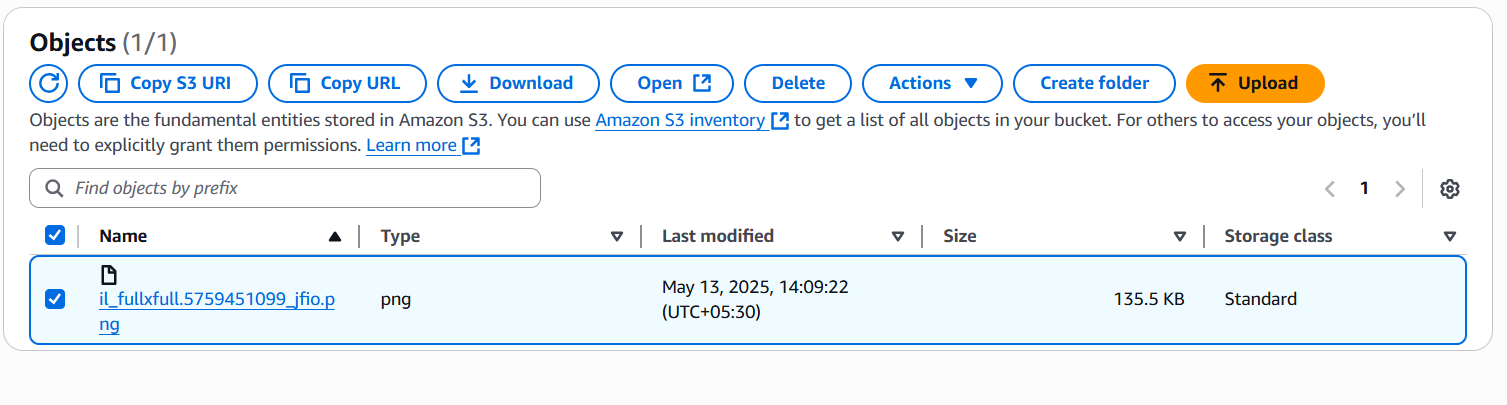


Fig: Copy S3 URL.

**Copy S3 URL:** s3://myfisrtbucket1234567/il\_fullxfull.5759451099\_jfio.png

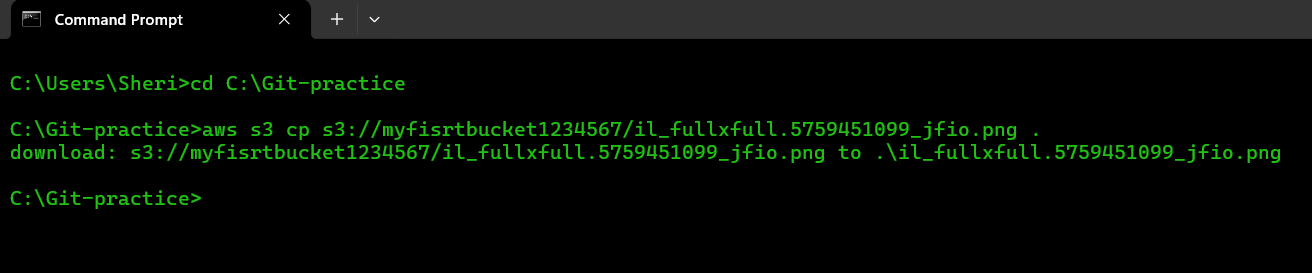


Fig: Image file is downloaded at current path (C:\Git-practice).

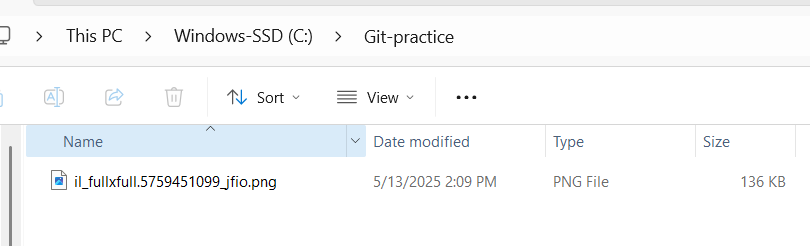


Fig: Image file is downloaded in our local machine.

From this we can say that we can access the Objects of the private S3 bucket in two ways, using **Presigned URL** and by **downloading image using AWS CLI.**

**Making S3 bucker public and accessing the objects within it:**

Now try to access the Objects of the S3 bucket by keeping it in Public. For this

**Go To 🡪 S3 bucket 🡪permissions 🡪 Block Public access.**

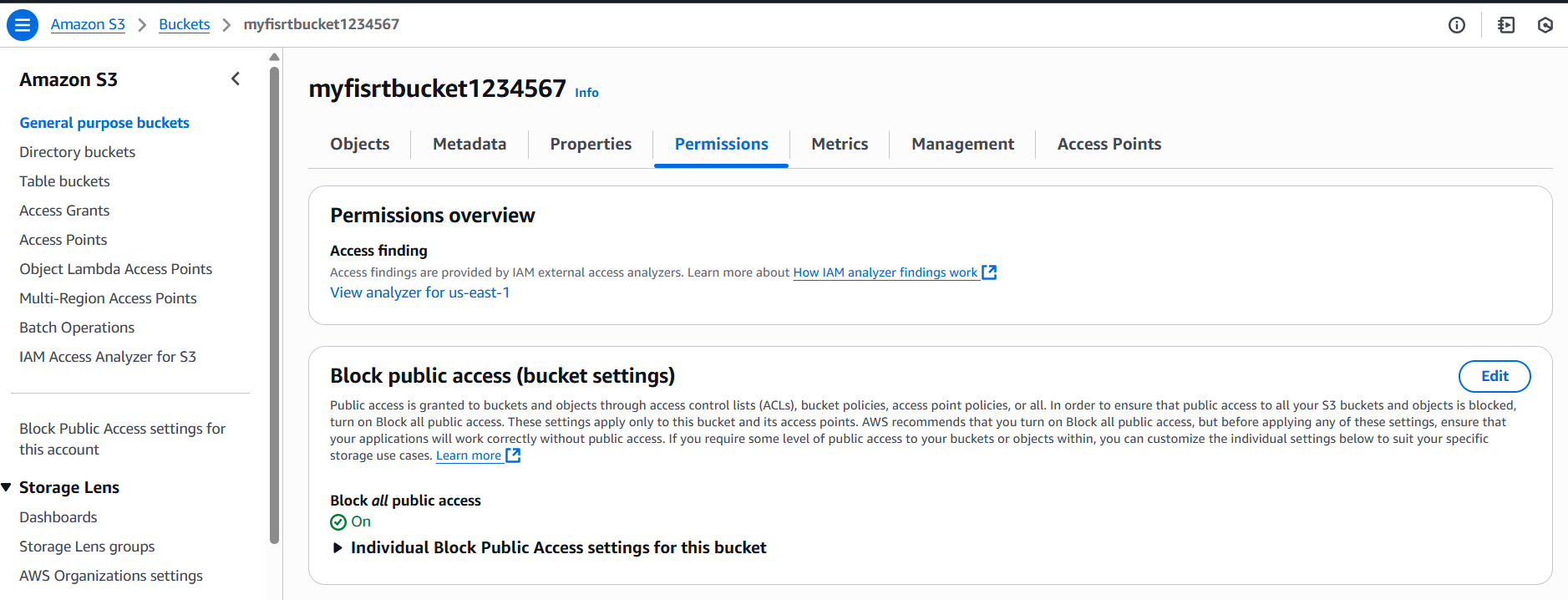


Fig: Edit Block public access.

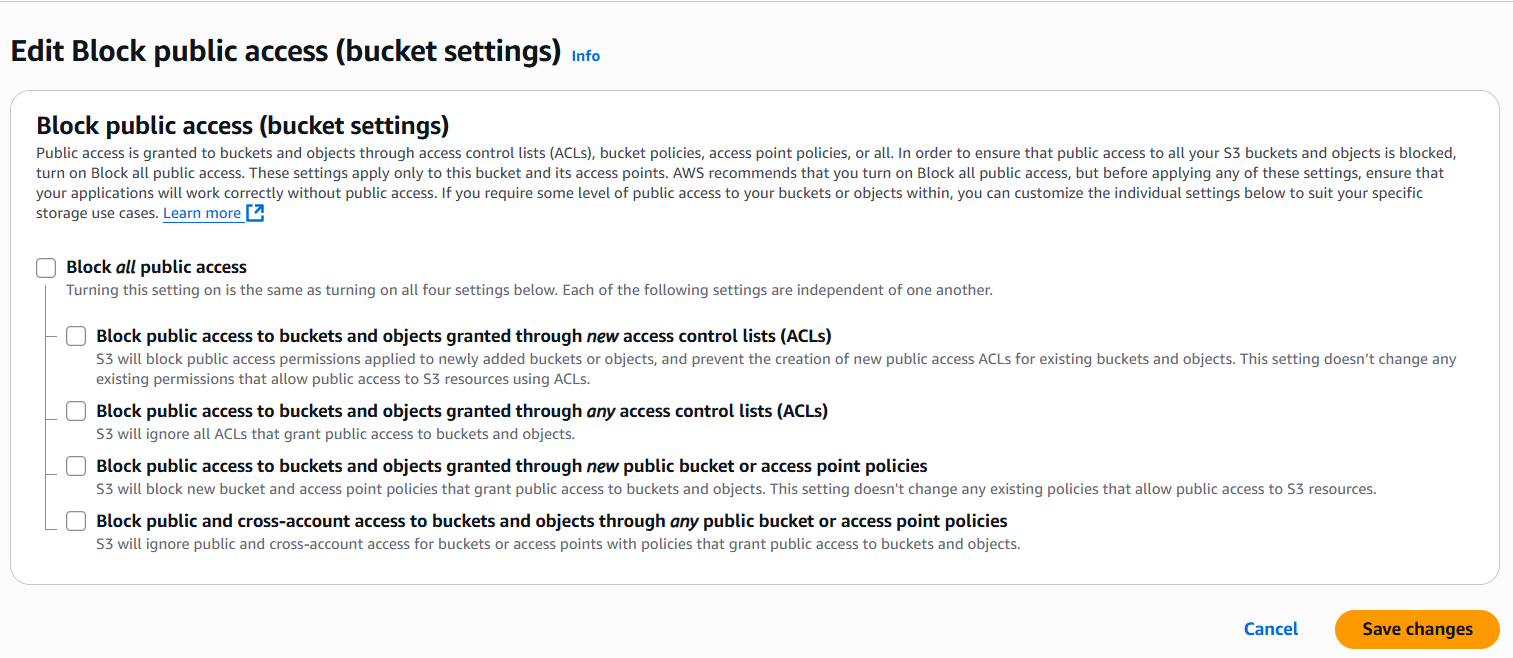


Fig: Enable public access.

Now again browse the object by using the Object URL in any browser.

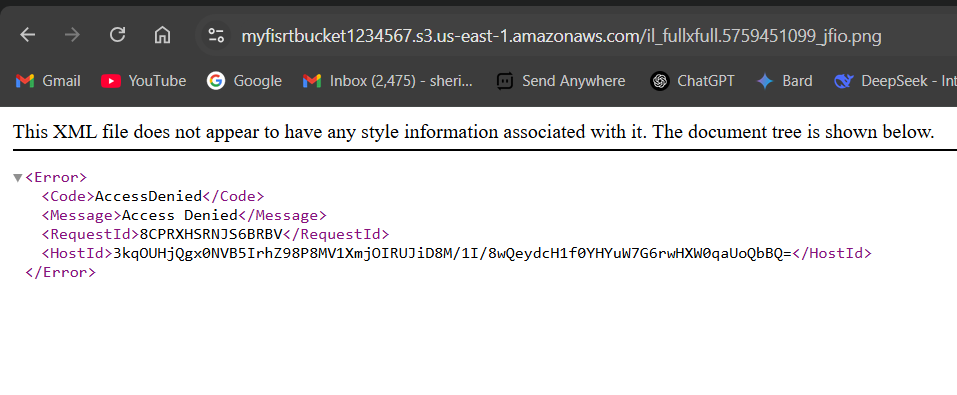


Fig: Still we are unable to access the object/image.

To access objects from a public S3 bucket, you need to set the right **permissions** using something called a **bucket policy**.

**Note1:** Even if the **S3** **bucket is public**, you still need to **allow public read access to each object** you want to make accessible. Without this, users won't be able to open or download the file.

**Note2:** These bucket policies are written in **JSON** file.

{

"Id": "BucketPolicy",

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

"Statement": [

{

"Sid": "AllAccess",

"Action": "s3:\*",

"Effect": "Allow",

"Resource": [

"arn:aws:s3:::myfisrtbucket1234567",

"arn:aws:s3:::myfisrtbucket1234567/\*"

],

"Principal": "\*"

}

]

Fig: Bucket policies to allow all access.

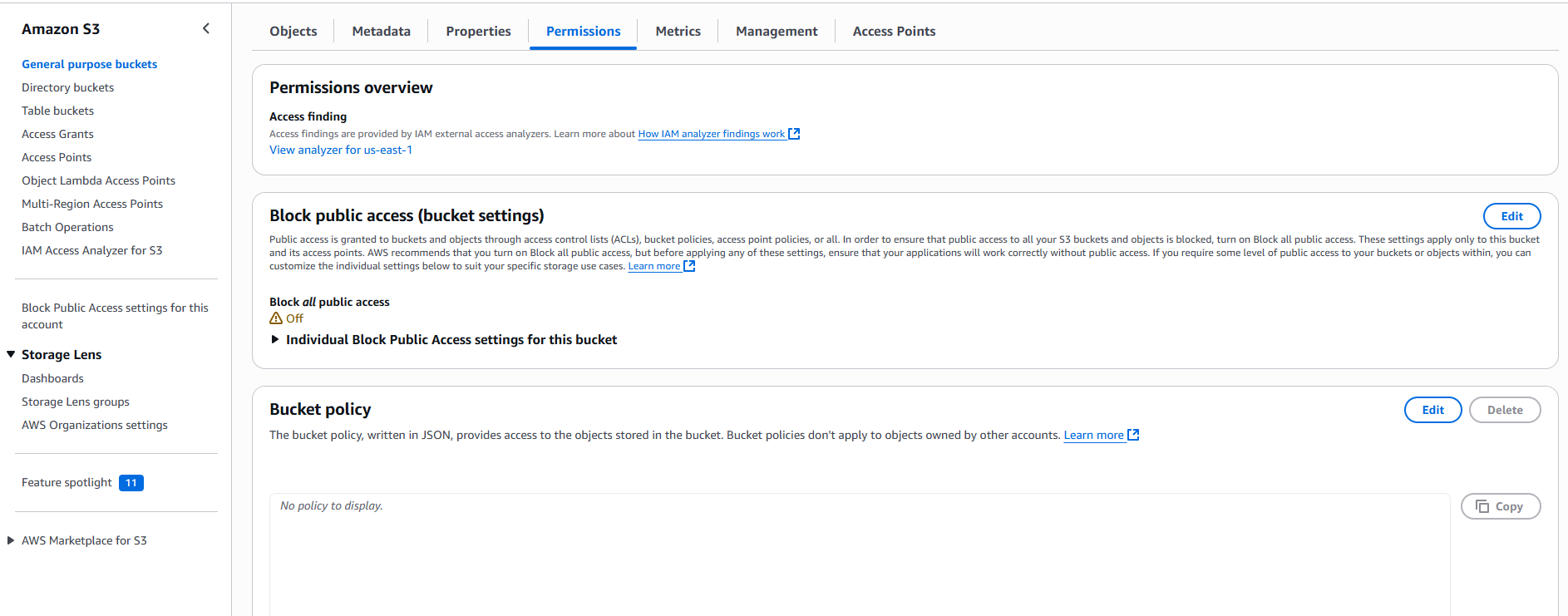


Fig: Bucket policy

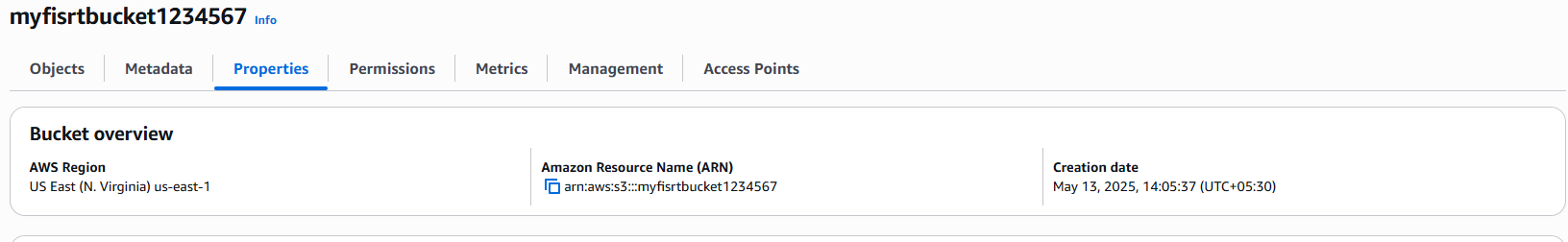


Fig: Amazon resource name (ARN).

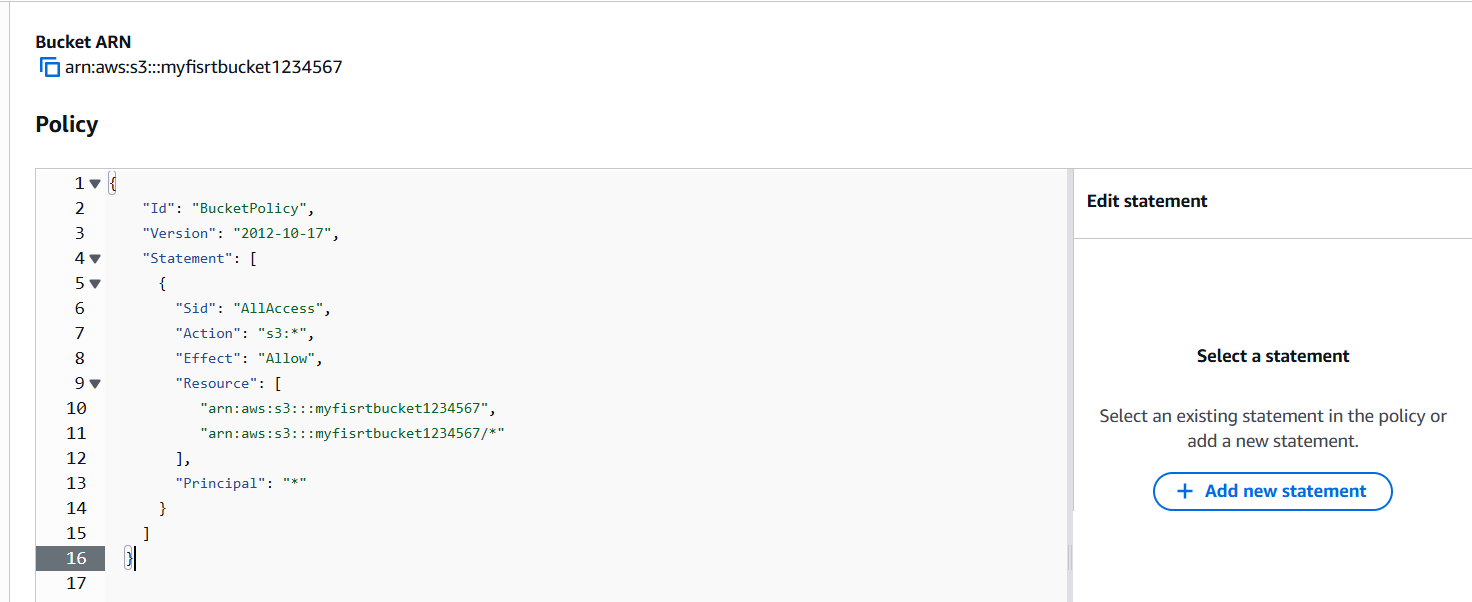


Fig: Bucket Policy is uploaded.

Save the Bucket policy and try once again using the image URL in any browser.

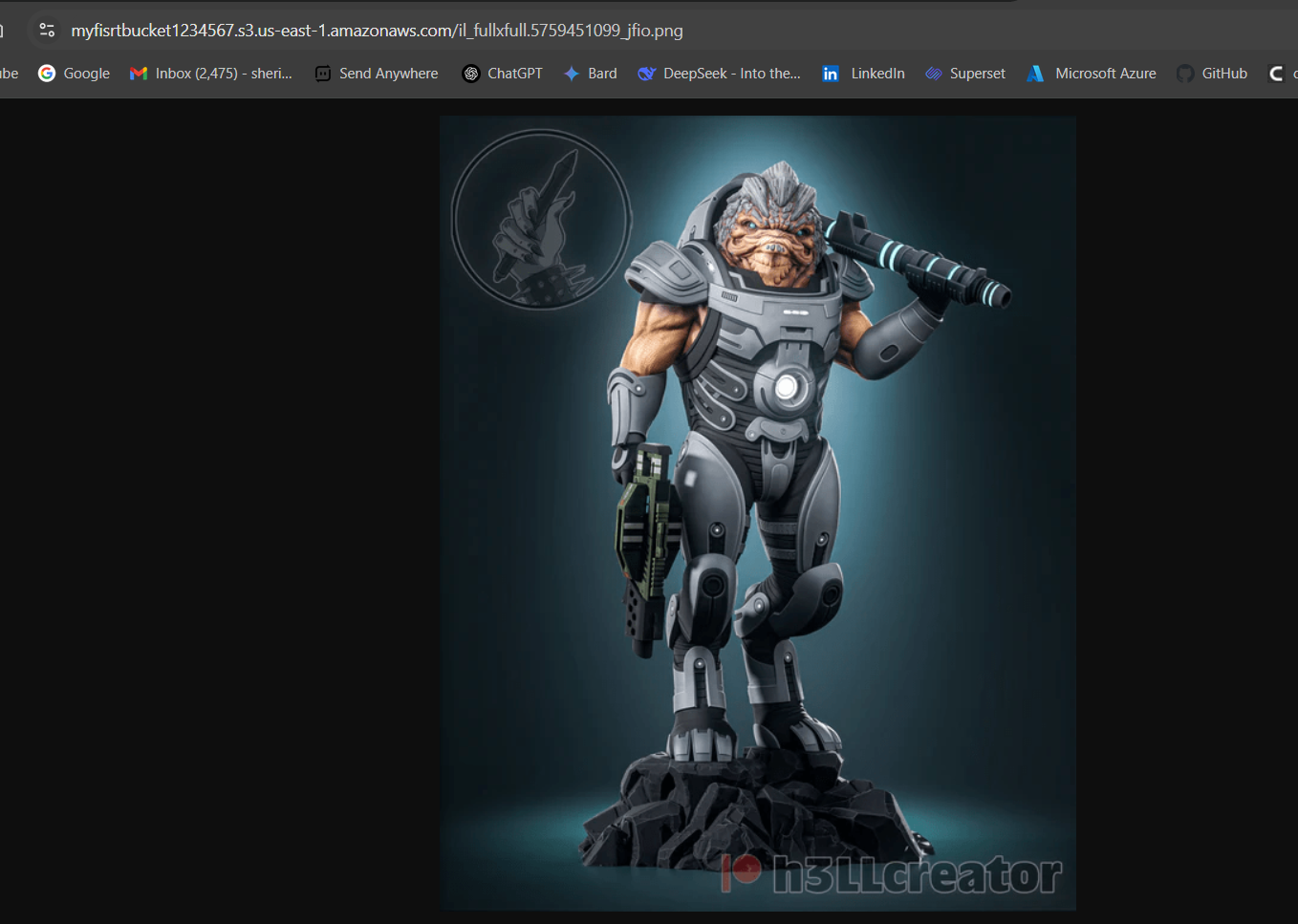


Fig: Able to access the image.

So we are able to access the image using the URL after writing the bucket policy for the specific image (object).

Now again write one more bucket policy to allow access to the **specific IP address** over a specific S3 bucket object.

{

"Version": "2008-10-17",

"Id": "S3PolicyId1",

"Statement": [

{

"Sid": "IPAllow",

"Effect": "Allow",

"Principal": {

"AWS": "\*"

},

"Action": "s3:\*",

"Resource": "arn:aws:s3:::myfisrtbucket1234567/\*",

"Condition": {

"IpAddress": {

"aws:SourceIp": "115.98.47.41/32"

}

}

}

]

}

Fig: Bucket policy to allow access to specific IP address.

IP Address: 115.98.47.41 is my laptop public IP.

So the policy say that the specific IP Address which is provided in the policy can only access the S3 bucket object using the object URL.