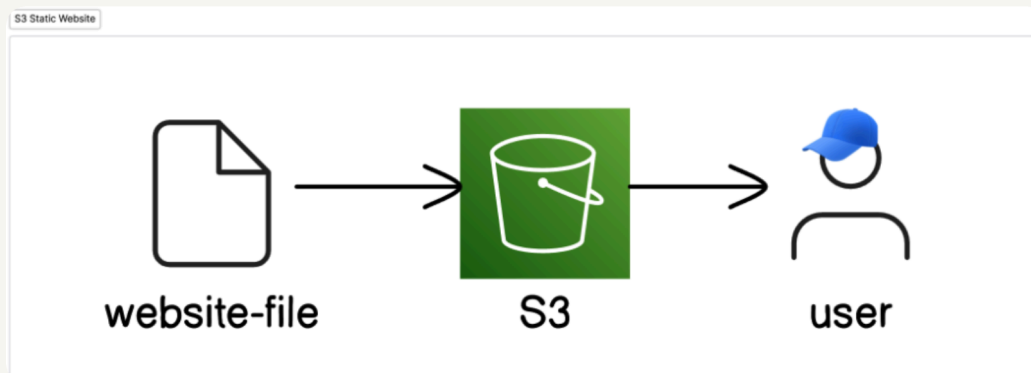


Static Website Hosted on Amazon S3 with IAM-Based Access Control



Introduction

My name is Mohammed Amir. In this project, I hosted a static website on Amazon S3 while enforcing secure access control using IAM policies. The goal was to deploy a publicly accessible website while granting a dedicated IAM user limited permissions to upload objects without the ability to delete them — following the Principle of Least Privilege.



Topics Covered :-

1. Amazon S3 (Simple Storage Service) :- **S3 is an object storage service that allows storing and retrieving any amount of data from anywhere on the web.**

- A Bucket is a container for storing objects (files).
- Each bucket has unique access permissions and configurations like Bucket , Versioning, Lifecycle rules.
- **Bucket Versioning** :- Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

Example of Versioning in S3 Bucket:

You upload a file **report.docx** to an S3 bucket with versioning enabled. If you upload **report.docx** again (same name), S3 saves both versions — you can access or restore any previous version anytime.

- **Bucket Lifecycle** :- Lifecycle policies automatically transition objects to cheaper storage classes or delete them after a set time, helping optimize cost and manage data retention. S3 offers multiple storage classes for different use cases, from frequent access to archival. Below are the main storage classes :-
 1. S3 Standard — For frequently accessed data like active content, websites, and applications.
 2. S3 Standard-IA (Infrequent Access) — For infrequently accessed data that still needs quick retrieval, like backups and disaster recovery.
 3. S3 One Zone-IA — For infrequently accessed data that can be stored in a single AZ, like secondary backups.
 4. S3 Intelligent-Tiering — For data with unknown or changing access patterns, automatically moves objects between s3 storage as per the access.

5. S3 Glacier — For archival data that requires occasional access, like compliance records.
6. S3 Glacier Deep Archive — For long-term archival of rarely accessed data, like regulatory archives.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☐ I acknowledge that the current settings might result in this bucket and the objects within becoming public.


Note :- While creating an S3 bucket if you want the objects to be publicly accessed then you must disable the block all public access.

Bucket Name Rules :-

- Globally Unique
- Bucket names must be between 3 (min) and 63 (max) characters long.
- Bucket names can consist only of lowercase letters, numbers, periods (.), and hyphens (-).
- Bucket names must begin and end with a letter or number.

General purpose buckets (1) [Info](#)


Buckets are containers for data stored in S3.



<

1

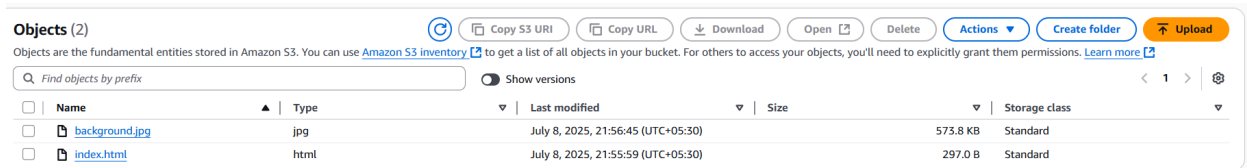
>



	Name ▲	AWS Region ▼	IAM Access Analyzer	Creation date ▼
<input type="radio"/>	amir-static-website-project	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	July 8, 2025, 20:27:45 (UTC+05:30)

2. Objects in S3 :- **An Object in S3 is any file (like HTML, image, video) stored inside a bucket.**

- Each object has metadata and a unique key (filename).
- Objects can be public or private based on bucket policy or ACL.
- You upload all the required files for your website to work in the s3 bucket.



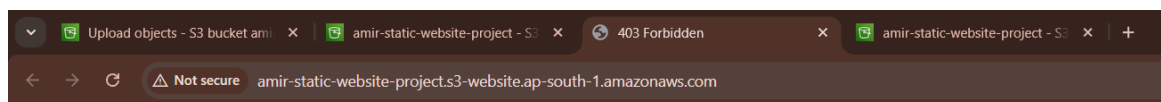
	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	background.jpg	jpg	July 8, 2025, 21:56:45 (UTC+05:30)	573.8 KB	Standard
<input type="checkbox"/>	index.html	html	July 8, 2025, 21:55:59 (UTC+05:30)	297.0 B	Standard

Metadata — Data about the object, such as its content type , size, creation date, encryption details, or custom tags.

Unique Key — The name of the object (file name + path) that uniquely identifies it within the bucket.

3. 403 Access Denied Error

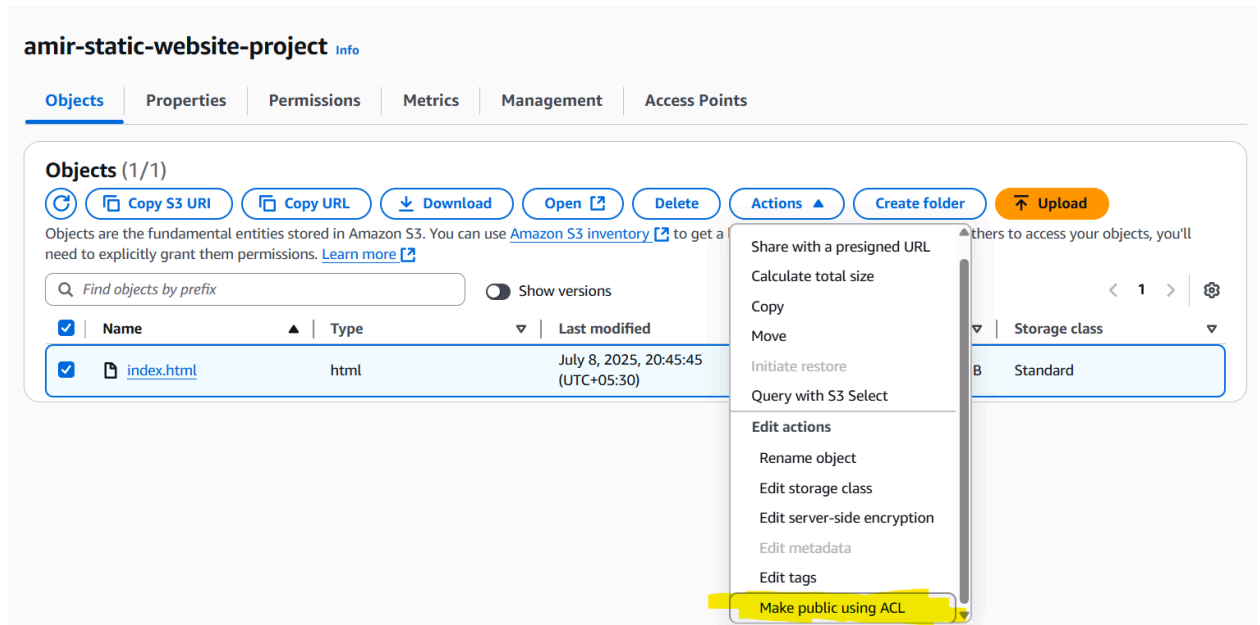
Common error when trying to access an object over the internet without **making the objects public using ACL.**



403 Forbidden

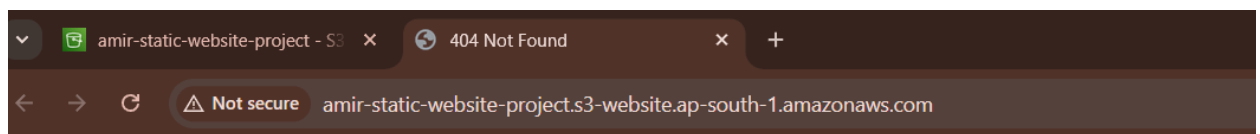
- Code: AccessDenied
- Message: Access Denied
- RequestId: 0BGPD57387KY19ZH
- HostId: eX7z9vW7N1Wv1Xgq37n0g0mxUoKmVmB5qlUkQdUjB6814eWt3dTznDw32X7uAAr+JLcxQzpAfuI=

ACL (Access Control List) in Amazon S3 is a permissions mechanism that defines who can access an object or bucket.



4. 404 Not Found Error

Occurs when the static website hosting is not enabled or index document is not set.



404 Not Found

- Code: NoSuchWebsiteConfiguration
- Message: The specified bucket does not have a website configuration
- BucketName: amir-static-website-project
- RequestId: 607STD5TF0Y0DC0
- HostId: HmMADlpcrNFC4UVgezET9+FXaPcwrMISisli5u28JoGeDZwhH9WR4zxWaQW0YyJEXOeqAZQro4E=

Static Website — A website with fixed content that doesn't change unless edited manually, built using HTML, CSS, and JavaScript. **Dynamic Website** — A website that displays content dynamically based on user interaction or server-side processing, often using databases and backend scripting.

- Solved by enabling static website hosting and specifying **index.html**.(landing page)

Edit static website hosting [Info](#)

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable
☒ Enable

Hosting type

☒ Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Index document
Specify the home or default page of the website.

index.html

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

5. IAM (Identity and Access Management) :- **AWS service to securely manage access to AWS resources by controlling who can do what.**

1 Created a Custom IAM Policy :- A JSON document that defines permissions (allow/deny) for users, groups, or roles to access AWS resources..

Policy Name: S3-StaticWebsite-UploadPolicy :-

Permissions defined in this policy [Info](#)

Permissions defined in this policy document specify which actions are allowed or denied.

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "UploadObject",
6        "Effect": "Allow",
7        "Action": [
8          "s3:PutObject",
9          "s3:GetObject"
10       ],
11       "Resource": "arn:aws:s3:::amir-static-website-project/*"
12     },
13     {
14       "Sid": "ListBucket",
15       "Effect": "Allow",
16       "Action": [
17         "s3:ListAllMyBuckets",
18         "s3:ListBucket"
19       ],
20       "Resource": "*"
21     }
22   ]
23 }
```

2 Created an IAM Group :- A collection of IAM users; permissions assigned to the group apply to all its members.

Group Name: S3-StaticWebsite-Uploaders.

Attached the custom policy S3-StaticWebsite-UploadPolicy to the group.

The screenshot displays the AWS IAM console interface for the 'S3-StaticWebsite-Uploaders' group. The 'Summary' tab is active, showing the group's name, creation time (July 14, 2025, 22:00 UTC+05:30), and ARN. Below this, the 'Permissions' tab is selected, showing a list of attached policies. The 'S3-StaticWebsite-UploadPolicy' is listed as a 'Customer managed' policy. The interface includes buttons for 'Delete', 'Edit', 'Simulate', 'Remove', and 'Add permissions'.

Policy name	Type	Attached entities
S3-StaticWebsite-UploadPolicy	Customer managed	1

✓ Why This Approach?

Easier Management: Future users can be added to the group with inherited permissions.

Policy Consistency: No need to attach policies individually.

Scalable Access Control: Aligns with IAM best practices.

3 Created IAM User :- An identity with credentials used by a person or application to access AWS services.

User Name: amir with **MFA enabled**.

Added to group: S3-StaticWebsite-Uploaders

The screenshot shows the AWS IAM console interface for an IAM user named 'amir'. The 'Summary' tab is active, displaying the user's ARN (arn:aws:iam::322492479923:user/amir), creation date (July 14, 2025, 22:14 UTC+05:30), and console access status (Enabled with MFA). The 'Permissions' tab is also visible, showing a list of permissions policies. One policy, 'S3-StaticWebsite-UploadPolicy', is listed, attached via the 'S3-StaticWebsite-Uploaders' group. The interface includes a search bar, a filter by type dropdown, and a table with columns for policy name, type, and attached via.

Policy name	Type	Attached via
S3-StaticWebsite-UploadPolicy	Customer managed	Group S3-StaticWebsite-Uploaders

✓ What is MFA (Multi-Factor Authentication) ?

MFA = An extra layer of security beyond just username & password.

It requires the user to provide two or more authentication factors to sign in :-

- You sign in with IAM username + password
- ✓ Then, you enter a 6-digit OTP code from an Authenticator app like Google Authenticator or AWS MFA app.

Policy validation using AWS IAM Policy Simulator

To ensure that the IAM permissions granted to the user were correct and functional, we used the AWS IAM Policy Simulator to simulate and test various S3 actions based on the attached policy.

The following actions were tested:

✅ s3:ListAllMyBuckets — Allowed (User can list all buckets in the account)

✅ s3:ListBucket — Allowed (User can list objects inside the allowed bucket)

✅ s3:PutObject — Allowed (User can upload objects to the bucket)

❌ s3:DeleteObject — Denied (As expected, since DeleteObject was not permitted in the policy , **By default, all actions are implicitly denied in AWS unless explicitly allowed**)

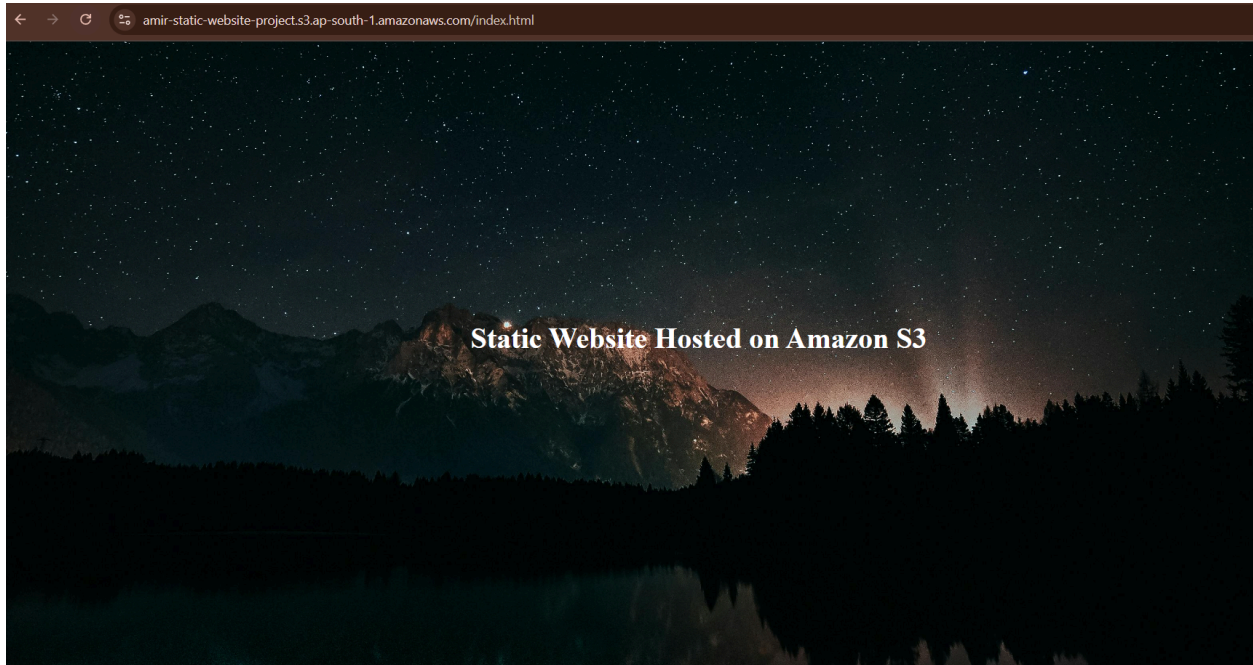
The simulator results confirmed that the IAM policy correctly enforces the intended permissions and restrictions.

The screenshot displays the AWS IAM Policy Simulator interface. On the left, the 'Policies' section shows the 'S3-StaticWebsite-UploadPolicy' being edited. The policy document is visible, showing two statements: one for 's3:PutObject' and 's3:GetObject' actions, and another for 's3:ListAllMyBuckets' and 's3:ListBucket' actions. On the right, the 'Policy Simulator' section shows the simulation results for the selected actions. The results table indicates that 'ListAllMyBuckets', 'ListBucket', and 'PutObject' are allowed, while 'DeleteObject' is denied.

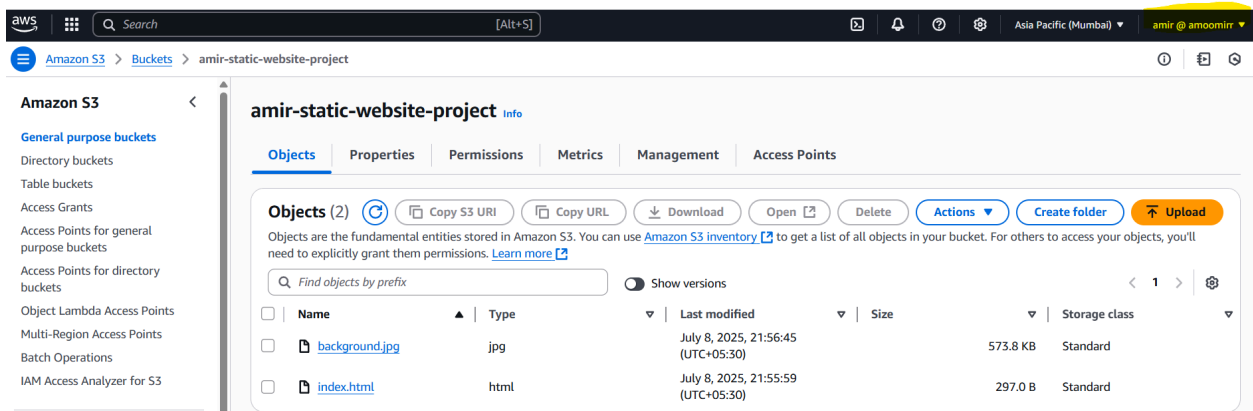
Service	Action	Resource Type	Simulation Resource	Permission
Amazon S3	ListAllMyBuckets	not required	*	allowed 1 matching statements.
Amazon S3	ListBucket	bucket	*	allowed 1 matching statements.
Amazon S3	PutObject	object	object	allowed 1 matching statements.
Amazon S3	DeleteObject	object	object	denied Implicitly denied (no matching...

Website URL :-

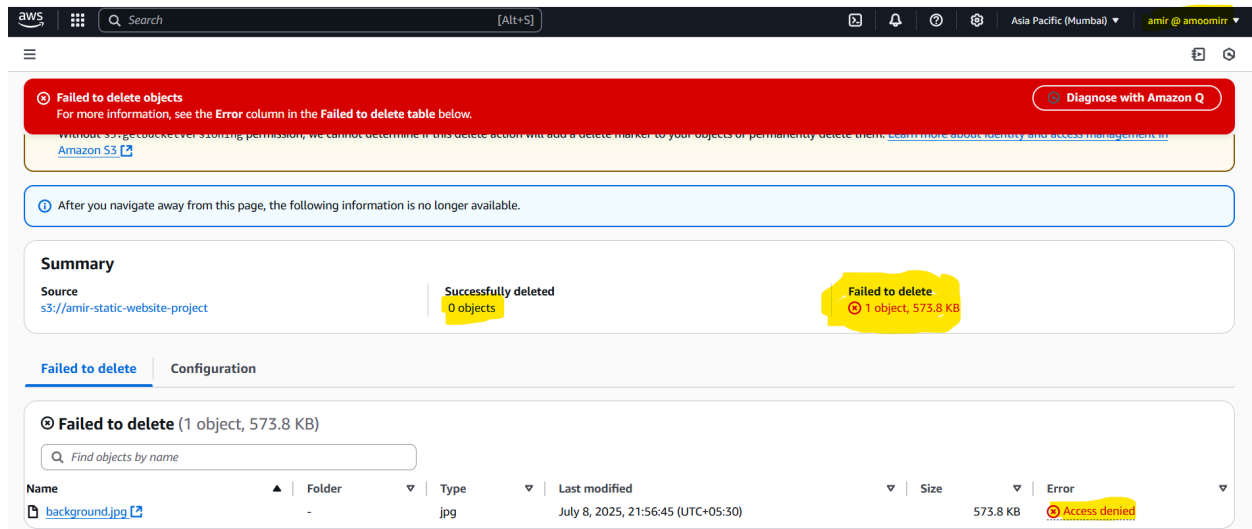
<https://amir-static-website-project.s3.ap-south-1.amazonaws.com/index.html>



User can list the bucket and objects :-



User cannot delete any objects as implicitly denied :-



✓ Conclusion and Project Outcome

In this project, we successfully hosted a static website on Amazon S3 with controlled access using AWS Identity and Access Management (IAM).

We ensured secure and controlled operations by following the principle of least privilege for IAM users.

Through this project, we achieved the following objectives:

- ✓ Created an Amazon S3 bucket and configured it for static website hosting.
- ✓ Uploaded web content and resolved common access errors (403 Forbidden, 404 Not Found).
- ✓ Created an IAM user and assigned permissions using a custom IAM policy
- ✓ Controlled access by allowing only specific actions — listing, uploading, and reading objects.
- ✓ Validated permissions using the AWS IAM Policy Simulator.
- ✓ Ensured sensitive actions like object deletion remained denied.