

# **Report: Hosting a Static Website on Amazon S3**

## **1. Introduction**

Amazon Simple Storage Service (Amazon S3) is a scalable object storage service widely used for hosting static websites, storing data, and building cloud-native applications. This report documents the step-by-step procedure of hosting a static website using Amazon S3, including bucket creation, uploading files, configuring public access, enabling static hosting, and validating the live website endpoint.

The objective of this experiment is to understand how S3 can be used as a low-cost and reliable hosting platform for static web content such as HTML, CSS, JavaScript, and media files.

## **2. Aim**

To host a static website on Amazon S3 by creating a uniquely named bucket, uploading website files, enabling public access, configuring static website hosting, and verifying the website through the S3 endpoint.

## **3. Requirements**

- AWS Account (Free Tier is sufficient)
- HTML webpage (index.html)
- Optional: images, CSS files, JavaScript files
- Stable internet connection

## **4. Procedure**

### **Step 1: Login to AWS Console**

- Open AWS Management Console.
- Sign in using your AWS credentials.
- Navigate to the S3 service from the Services menu.

## Step 2: Create an S3 Bucket

1. Navigate to **S3 → Create bucket**
2. Enter a unique bucket name (e.g., mythragsai)
3. Choose the region **ap-south-1 (Mumbai)**
4. Under **Block Public Access**, uncheck **Block all public access**
5. Acknowledge the warning
6. (Optional) Enable **Bucket Versioning**
7. Click **Create bucket**

### Observation:

- Bucket name: *mythragsai*
- Region: *Asia Pacific (Mumbai)*
- Public access: *Enabled*

Name	AWS Region	Creation date
mythragsai	Asia Pacific (Mumbai) ap-south-1	September 18, 2025, 09:13:03 (UTC+05:30)

## Step 3: Upload Website Files

1. Open the created bucket
2. Click **Upload → Add files**
3. Upload:
  - o index.html
  - o profile.jpg
  - o Any other assets
4. Click **Upload**

### Observation:

Files successfully uploaded to the S3 bucket.

The screenshot shows the Amazon S3 'Objects' page with two items listed:

Name	Type	Last modified	Size	Storage class
index.html	html	September 18, 2025, 09:16:53 (UTC+05:30)	4.7 KB	Standard
profile photo.jpg	jpg	September 18, 2025, 09:16:53 (UTC+05:30)	9.4 KB	Standard

## Step 4: Set Permissions Using ACL

By default, all uploaded objects are private. To allow public access:

1. Select index.html
2. Click **Actions → Make public using ACL**
3. Repeat for all required assets (images, CSS, scripts)

### Observation:

Public read permissions applied to website files.

The screenshot shows the 'Access control list (ACL)' page with the following details:

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: 34188f590c6b72d5491805503038784819810802317b9e34b4d93d5f3a4b128	List, Write	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	-	-
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-
S3 log delivery group Group: http://acs.amazonaws.com/groups/s3/LogDelivery	-	-

## Step 5: Enable Static Website Hosting

1. Go to **Bucket → Properties**
2. Scroll to **Static website hosting**
3. Select **Enable**
4. Choose **Host a static website**
5. Enter:
  - o **Index document:** index.html
6. Save changes
7. Copy the generated **Endpoint URL**

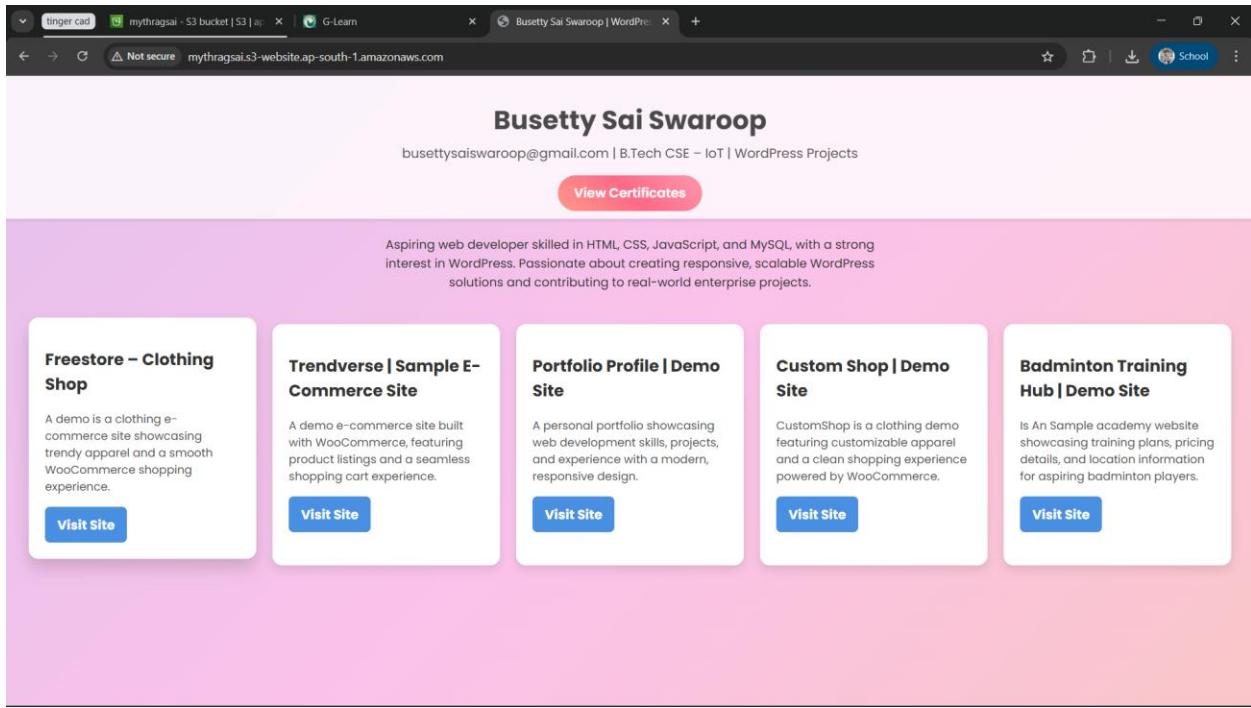
The screenshot shows the 'Static website hosting' configuration for an S3 bucket. At the top, there's a note about using AWS Amplify Hosting for static website hosting, with a 'Create Amplify app' button. Below that, it says 'S3 static website hosting' is Enabled. Under 'Hosting type', it's set to 'Bucket hosting'. In the 'Bucket website endpoint' section, it shows the configured URL: <http://mythragsai.s3-website.ap-south-1.amazonaws.com>.

**Endpoint URL:**

👉 <http://mythragsai.s3-website.ap-south-1.amazonaws.com>

## 6. Output

When visiting the endpoint URL, the hosted HTML webpage loads successfully with all images and assets. The website becomes publicly accessible from any browser over the internet.



## 7. Results

- A static website was successfully hosted on Amazon S3.
- The bucket was configured with proper public access settings and ACLs.
- Static website hosting was enabled and tested.
- The website is publicly available at the endpoint URL.

## 8. Conclusion

The experiment demonstrates how Amazon S3 can be used as an efficient, scalable, and cost-effective platform for hosting static websites. By configuring a bucket for public access and enabling static hosting, developers can deploy simple websites without managing servers or backend infrastructure. This approach is ideal for portfolios, documentation pages, and lightweight websites.

## 9. Endpoint URL

🔗 <http://mythragsai.s3-website.ap-south-1.amazonaws.com>