**Project 2**

**Deploy a Static Website on AWS**

In this, we will learn how to create a static website and deploy it using AWS services. A static website is a site that consists of HTML, CSS, and JavaScript files, and it doesn't require server-side processing or a database Deploy a Static Website on AWS.

📄 Table of Contents

* Introduction
* Why use AWS to host a static website?
* Common use cases
* Key AWS services used
* Summery
* Pictorial representation of deploying a static website on AWS
* Lab steps
* Optional Enhancement

**🌐 Introduction: Deploying a Static Website on AWS**

Deploying a **static website on AWS** is a powerful, scalable, and cost-effective way to host content such as personal blogs, portfolios, documentation, product landing pages, or informational pages.

Unlike dynamic websites, **static websites** consist of fixed files — like .html, .css, .js, and images — that don't require server-side processing. AWS provides services like **Amazon S3 (Simple Storage Service)** to host these static files and deliver them to users around the world reliably.

**🚀 Why Use AWS to Host a Static Website?**

**✅ Benefits:**

* **Low-cost**: You only pay for storage and bandwidth used.
* **Highly available & durable**: S3 offers 99.999999999% (11 9's) durability.
* **Scalable**: No servers to manage; S3 handles any traffic scale.
* **Secure**: Fine-grained access control and encryption.
* **Global reach**: Integrates easily with CloudFront CDN for fast worldwide delivery.
* **Custom domain support**: Easily connect your own domain name.
* **HTTPS support**: Through AWS CloudFront.

**🔧 Common Use Cases:**

* Personal or business **portfolio websites**
* **Documentation sites**
* **Landing pages**
* **Project demos**
* **Knowledge bases** with Table of Contents

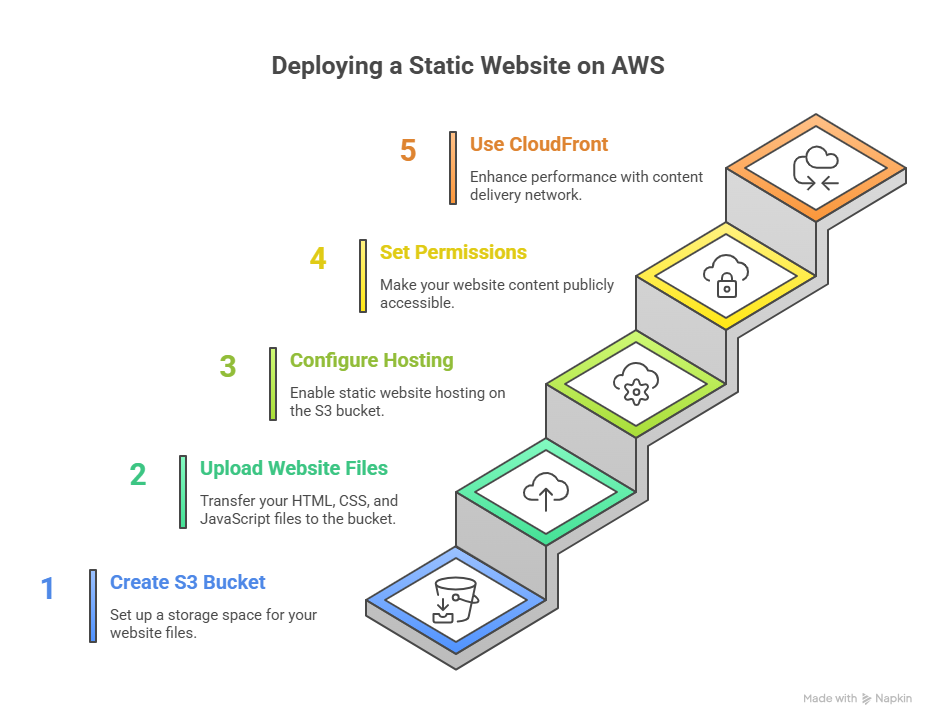
**🧭 Key AWS Services Used:**

| **Service** | **Role** |
| --- | --- |
| **Amazon S3** | Stores and serves your static files |
| **AWS IAM** | Manages access and permissions |
| **Amazon Route 53** *(optional)* | Connects custom domain names |
| **Amazon CloudFront** *(optional)* | Adds HTTPS and accelerates delivery |

**📌 Summary**

Deploying a static website on AWS lets you host professional-grade web content without managing web servers. By using S3, optionally CloudFront, and Route 53, you can deliver content quickly, securely, and at low cost.

* **Pictorial representation of deploying a static website on AWS**



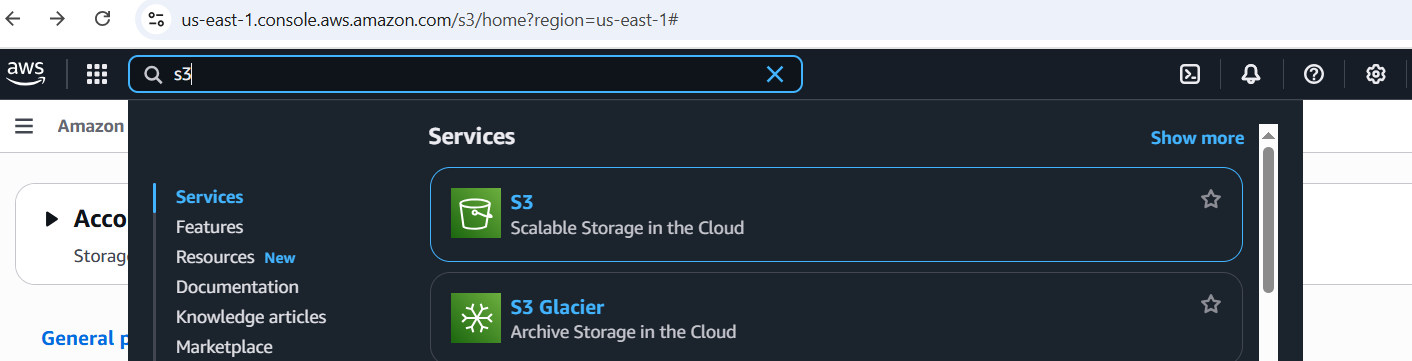
* Labs Steps:

**🚀 Step 1: Prepare Your Website Files**

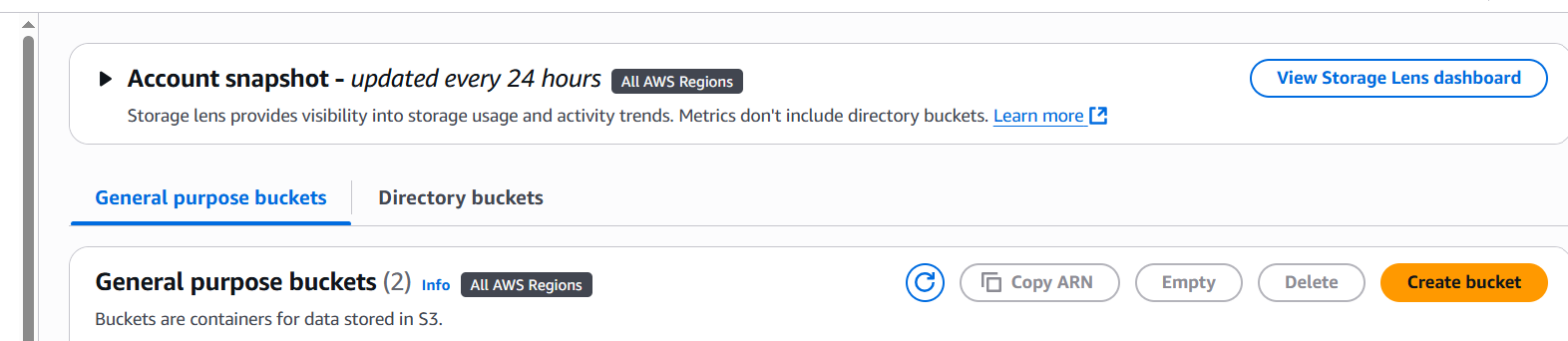
* Your website should consist of static files: index.html, styles.css, script.js, images, etc.
* Ensure your entry point is index.html.

**🟡 Step 2: Create an S3 Bucket**

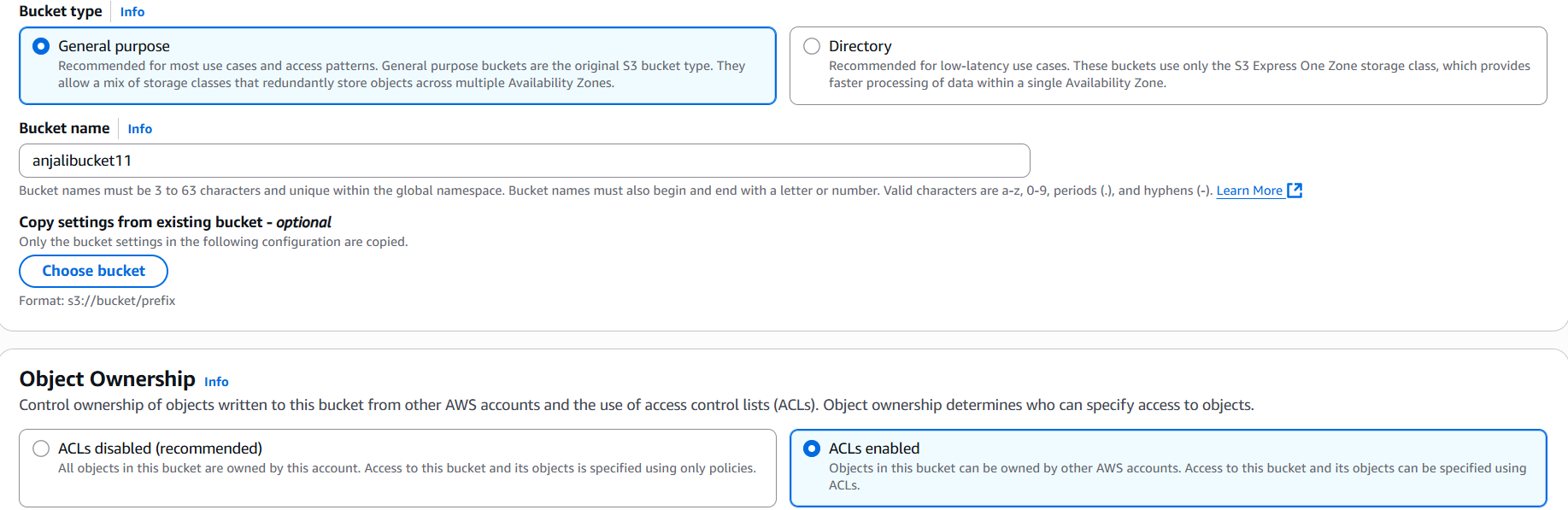
1. Go to the [AWS Management Console](https://console.aws.amazon.com/)
2. Navigate to **S3**



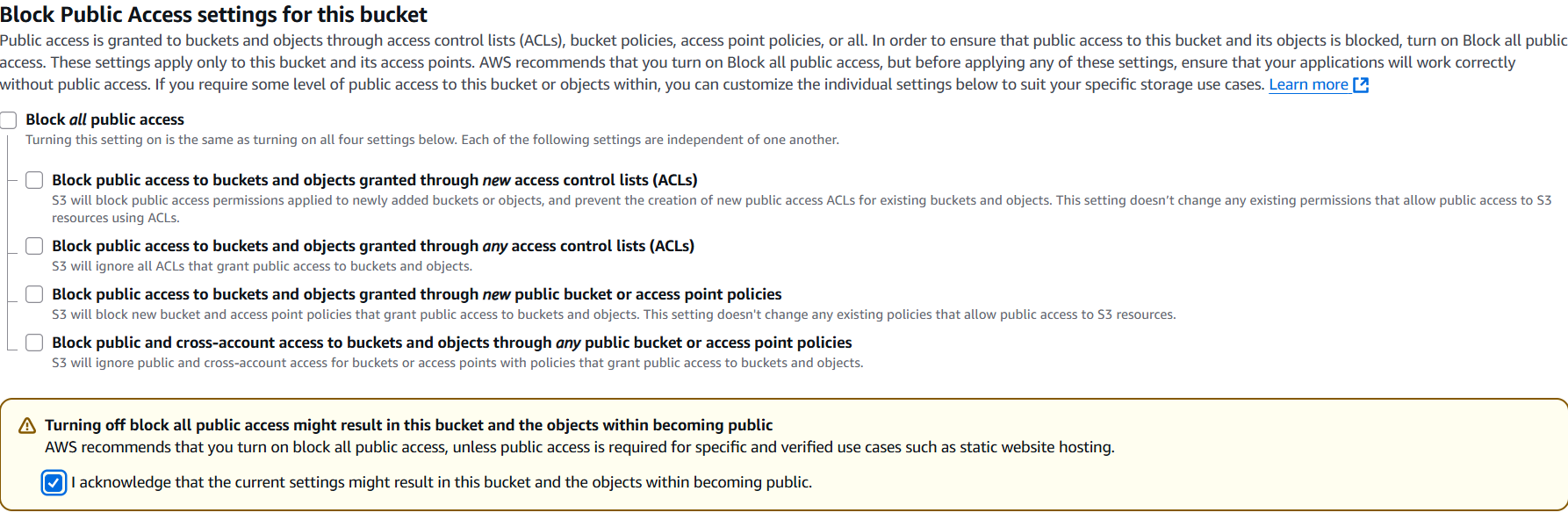
1. Click **Create bucket**



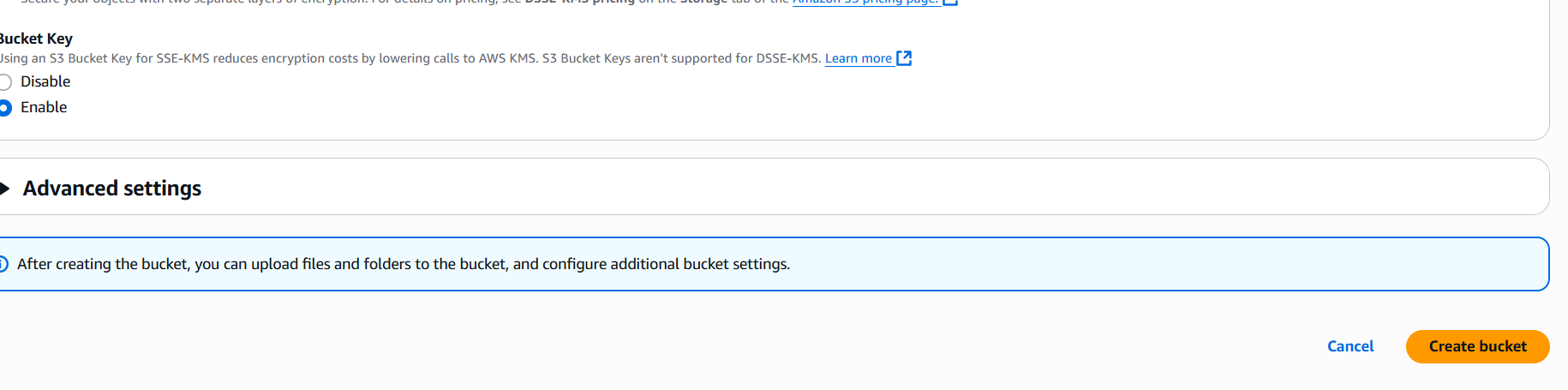
* + **Bucket name**: e.g., my-static-site
  + **Region**: Choose your preferred region



* + **Uncheck**: "Block all public access" (you will make it public)

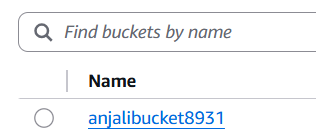


* + Confirm the warning and click **Create bucket**

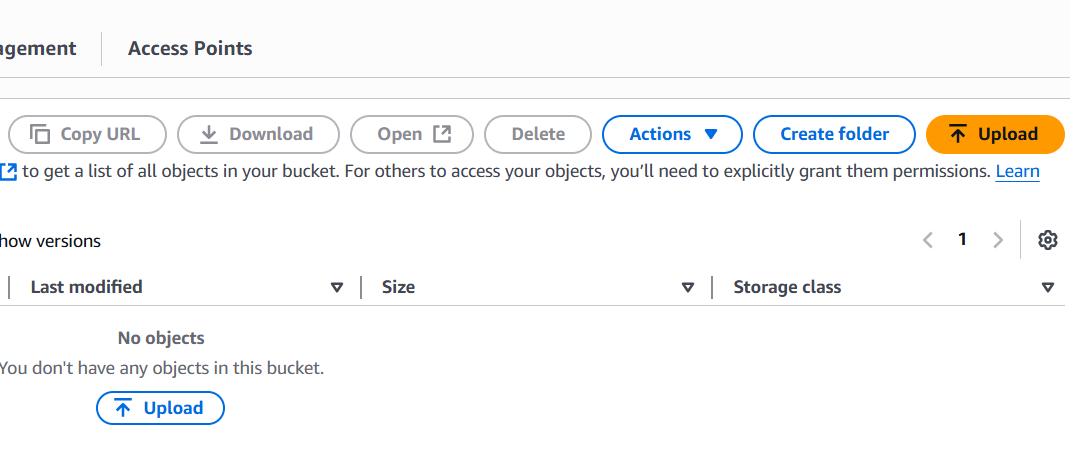
****

**📁 Step 3: Upload Website Files**

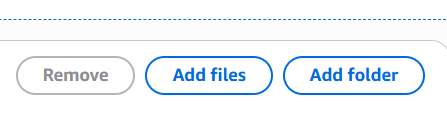
1. Click on the bucket name



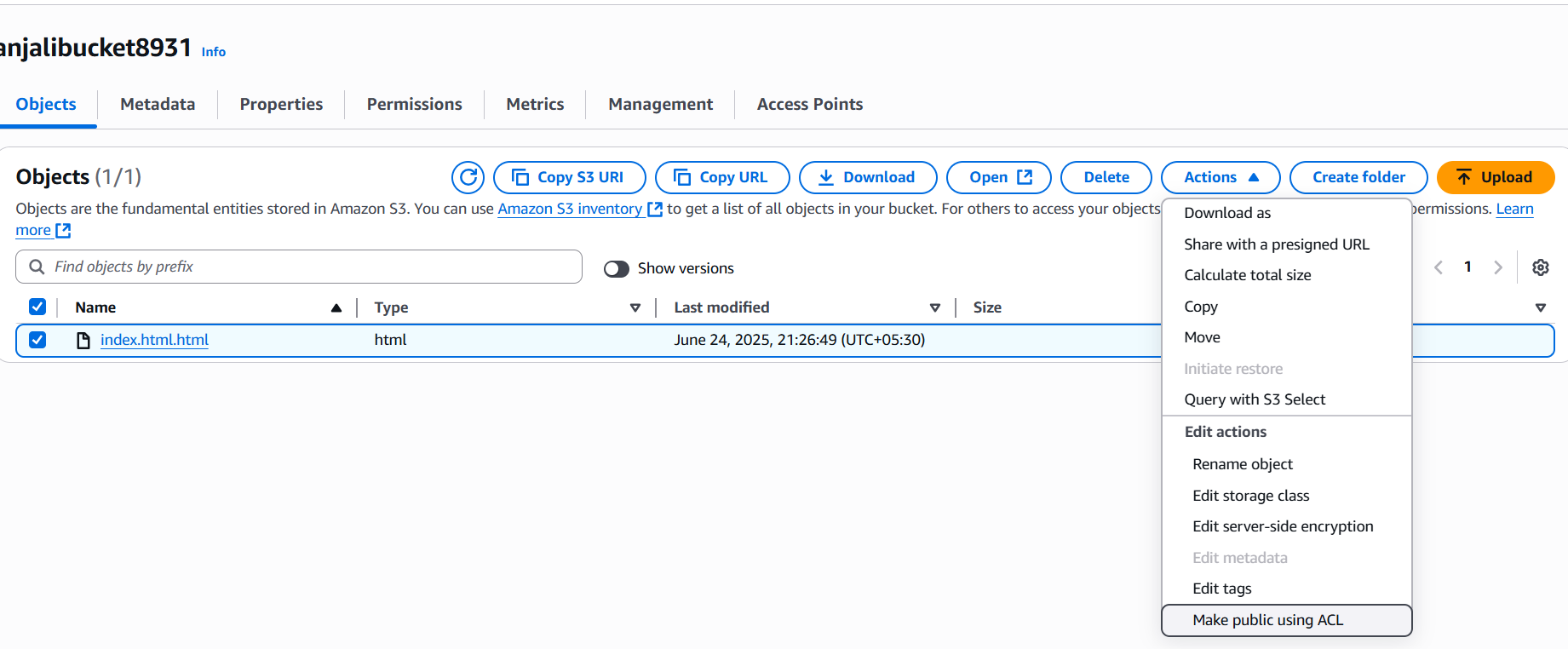
1. Click **Upload**



1. Upload your website files (e.g., index.html)



1. After uploading, go to each file > **Permissions** tab > **Make public** (or configure a bucket policy as shown below)

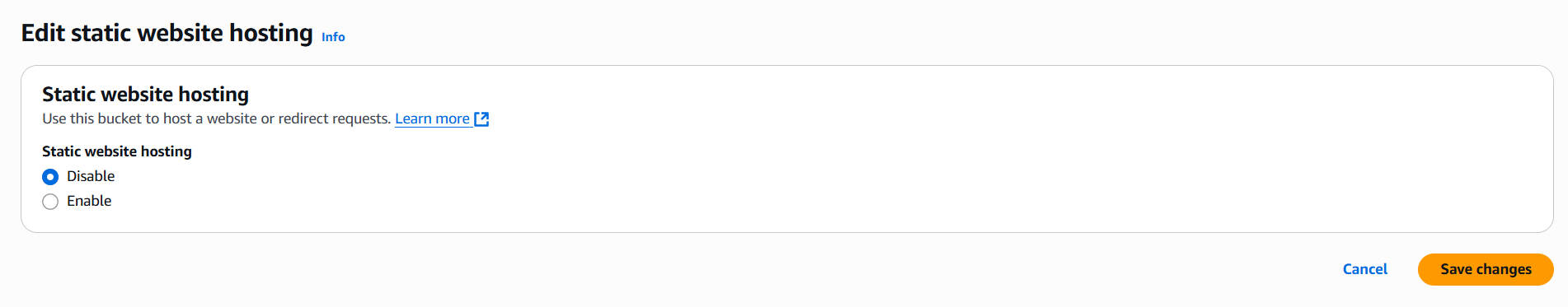


**⚙️ Step 4: Enable Static Website Hosting**

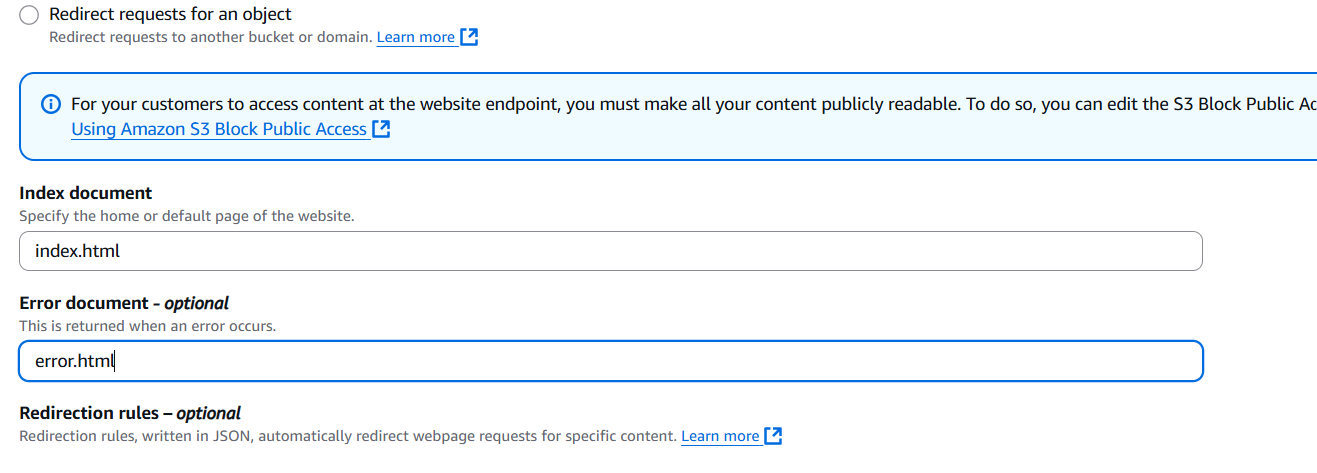
1. Go to **Properties** of the bucket



1. Scroll to **Static website hosting**
2. Choose **Enable**

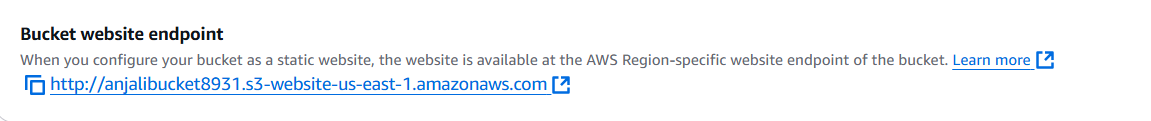


1. Set:
   * **Index document**: index.html
   * (Optional) **Error document**: error.html



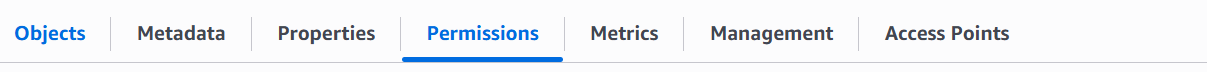
1. Click **Save**

A URL will be generated, e.g.,  
<http://my-static-site.s3-website-us-east-1.amazonaws.com/>



**🔐 Step 5: Set Bucket Policy for Public Access**

Go to **Permissions → Bucket Policy**, and paste this (replace your-bucket-name):



json

CopyEdit

{

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

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

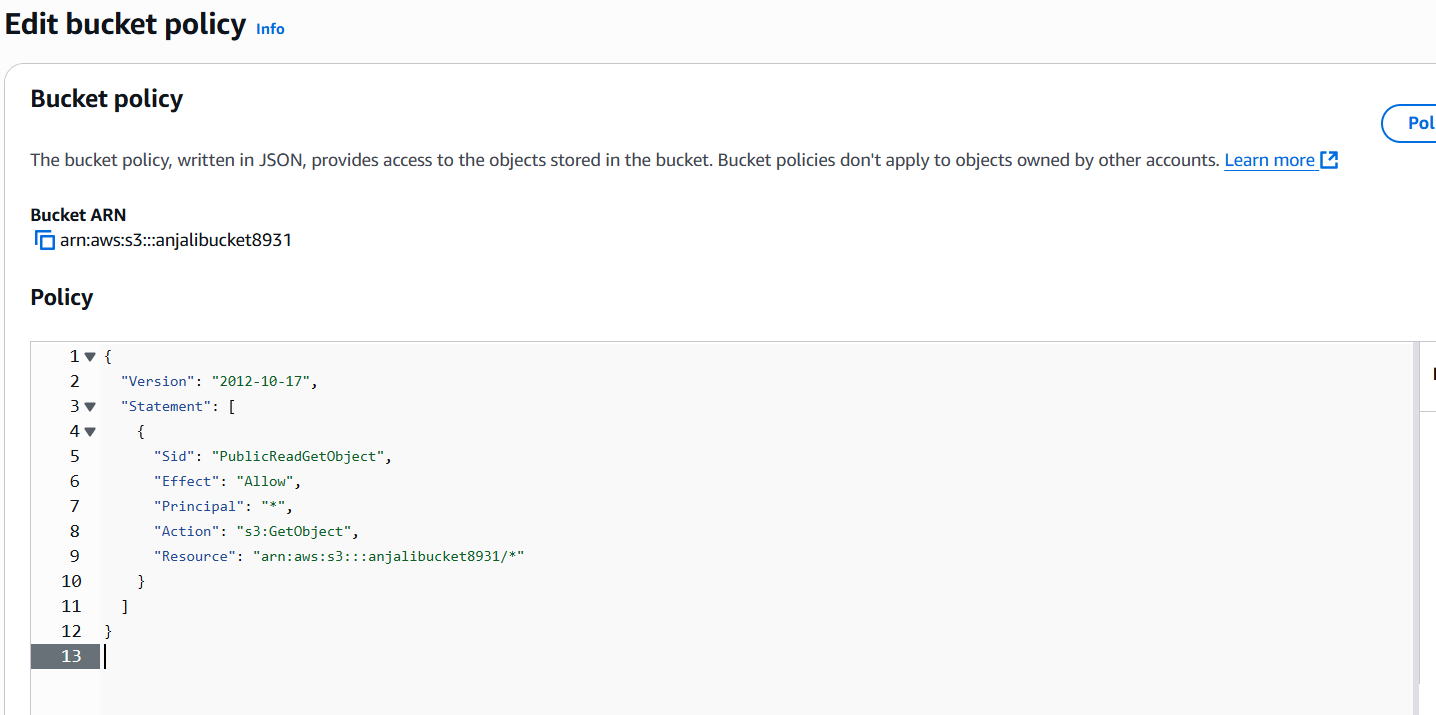
"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::your-bucket-name/\*"

}

]

}



Then click **Save**.

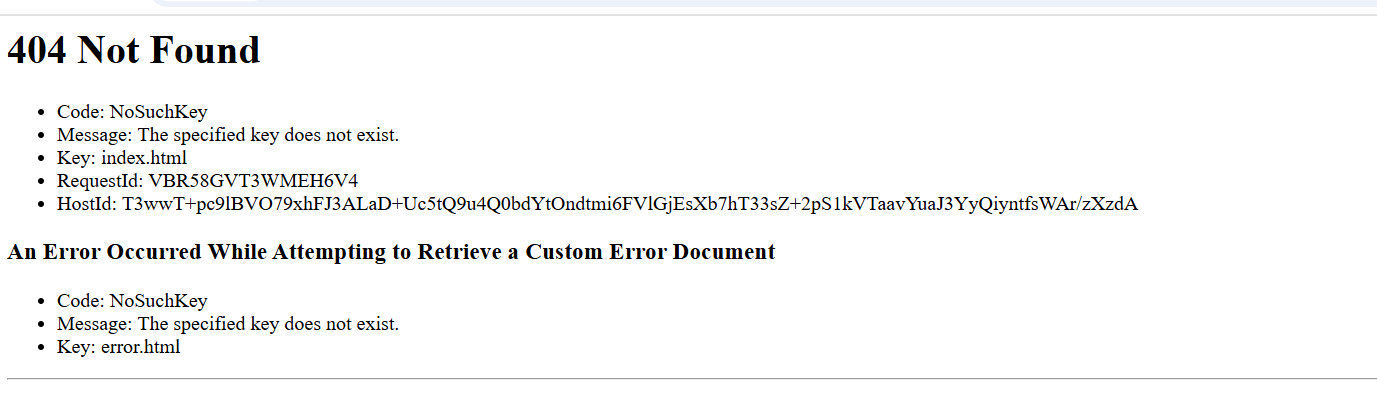
**🌐 Step 6: Access Your Website**

Visit the **Endpoint URL** provided under Static Website Hosting settings:  
http://<your-bucket-name>.s3-website-<region>.amazonaws.com



✅ Your static site is now live on AWS!

* If you put some unknown character on your URL so then you will see the error.



**🛡️ Optional Enhancements**

* **Use AWS CloudFront**: To enable HTTPS and CDN caching.
* **Connect a Custom Domain**: Use Route 53 or another DNS provider to point your domain to the CloudFront distribution.
* **Enable Logging**: To monitor access.

Thankyou….