Project: Hosting a Static Website on AWS S3

Introduction

This document outlines the steps to host a static website on Amazon Web Services (AWS) using the S3 (Simple Storage Service) service. S3 is a highly scalable, durable, and available object storage service that is perfect for hosting static websites.

Key Features

I) Cost-effective: Only pay for the storage and data transfer you use.

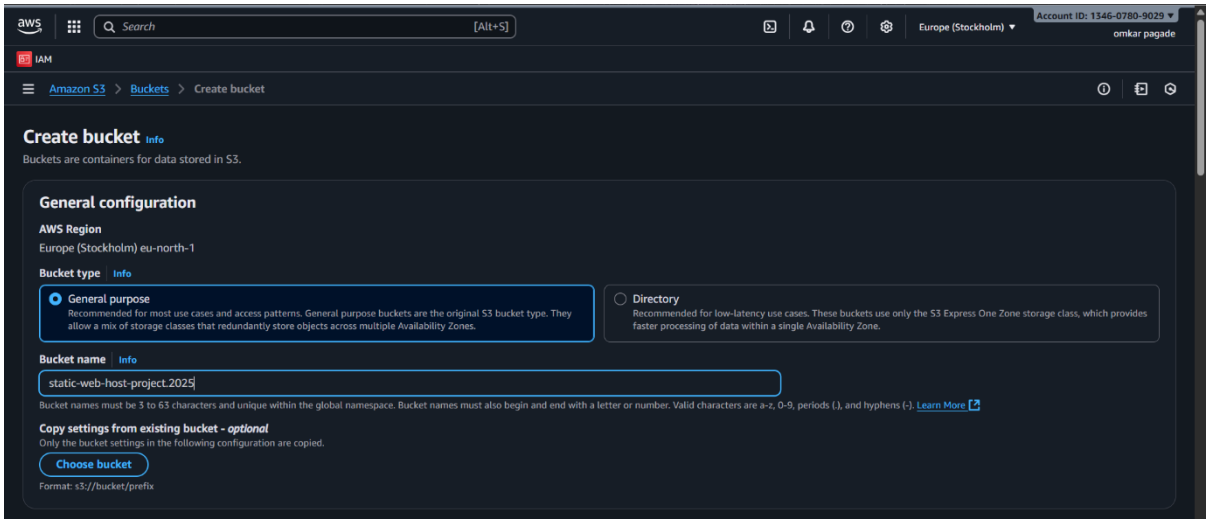
II) High availability: Your website is automatically replicated across multiple servers.

III) Scalability: Handles a large amount of traffic without manual intervention.

Step 1: Creating an S3 Bucket

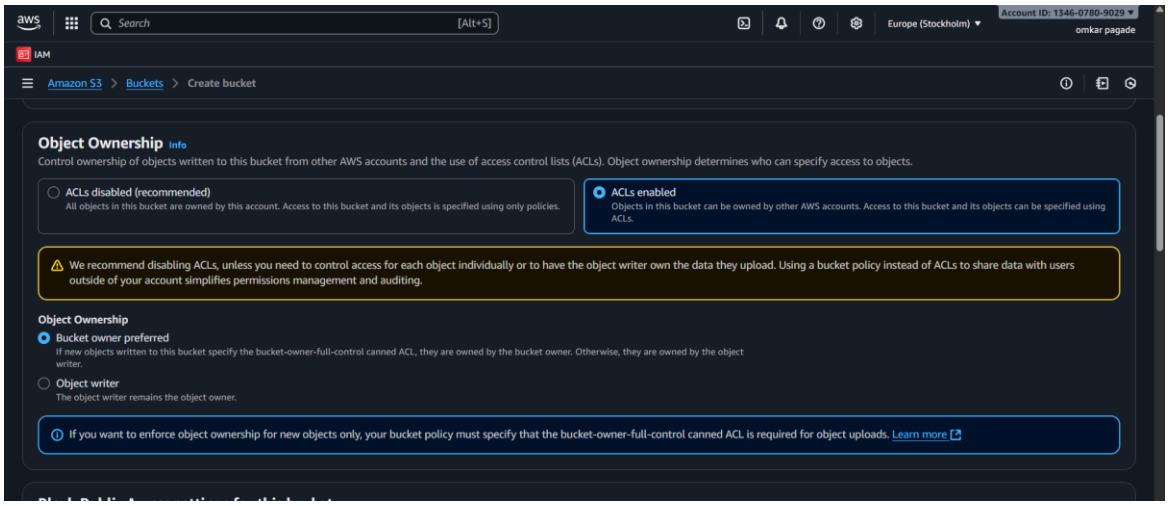
The first step is to create an S3 bucket. The bucket will store all the files for your static website.

Screenshot 1: Creating a new S3 bucket.



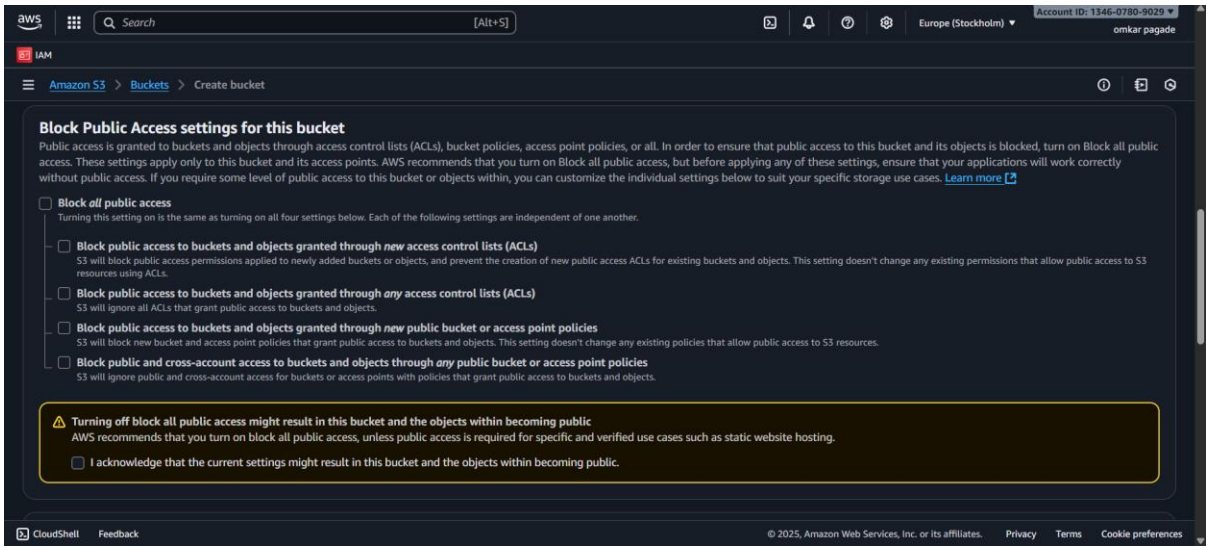
This screenshot shows the bucket creation page in the AWS Management Console. We are naming the bucket "static-web-host-project-2025" and selecting the "Europe (Stockholm)" region. The bucket name must be globally unique.

Screenshot 2: Configuring bucket permissions.



This image shows the "Object Ownership" and "Block Public Access" settings during bucket creation. We have chosen "ACLS enabled" but it is recommended to disable ACLs and use bucket policies for permission management. We will need to disable "Block all public access" later to make our website public.

Screenshot 3: Disabling public access block.

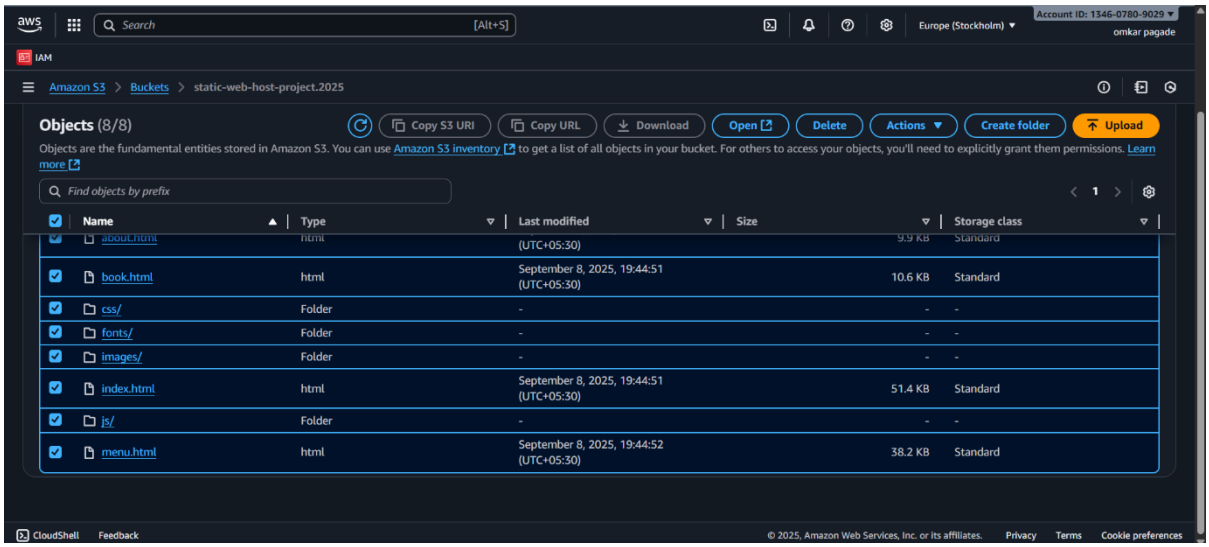


This screenshot confirms that we are disabling the "Block all public access" setting. This is a critical step because a static website needs to be publicly accessible for visitors to view it. The warning indicates that making the bucket public can expose its contents, which is the desired behaviour for a public website.

Step 2: Uploading Website Files

Once the bucket is created, you can upload your website's files and folders.

Screenshot 4: Uploading files to the bucket.

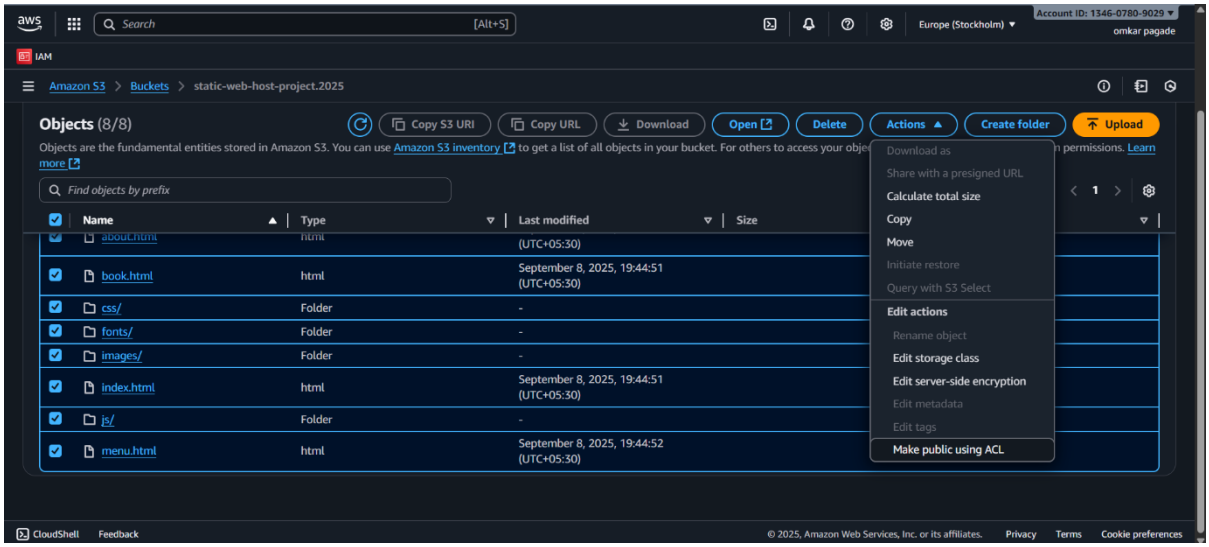


This screenshot shows the upload interface. Here, you would drag and drop your index.html, css/, js/, and other website files into the bucket.

Step 3: Making Files Public

By default, the uploaded files are not publicly accessible. You must grant public read access to each file.

Screenshot 6: Making objects public.

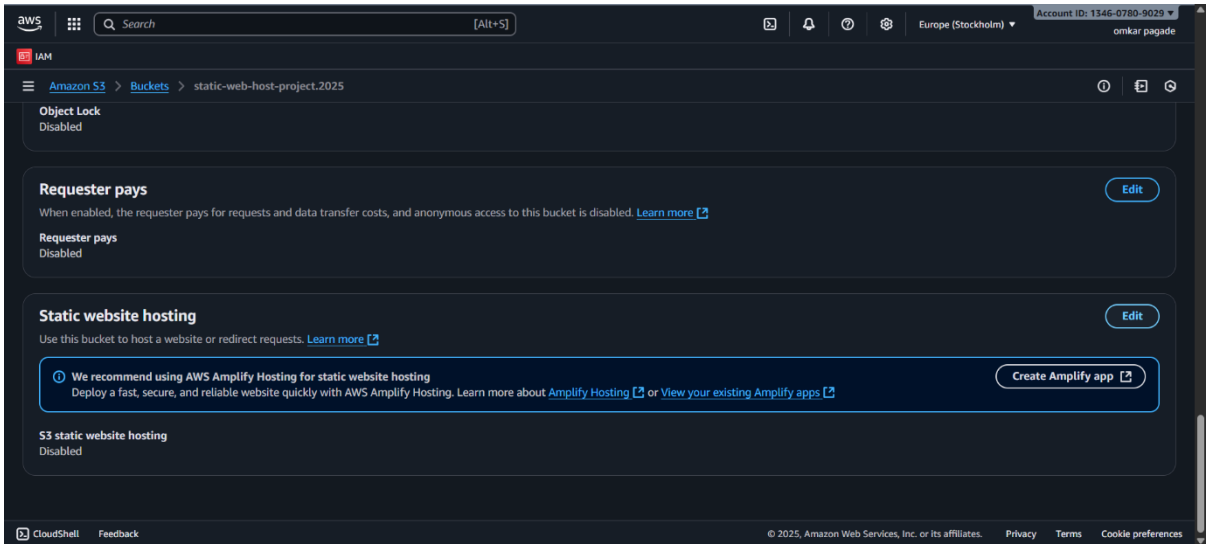


This screenshot shows the context menu after selecting the files. We choose "Make public using ACL" to grant public read permissions. This allows anyone on the internet to view the website content.

Step 4: Enabling Static Website Hosting

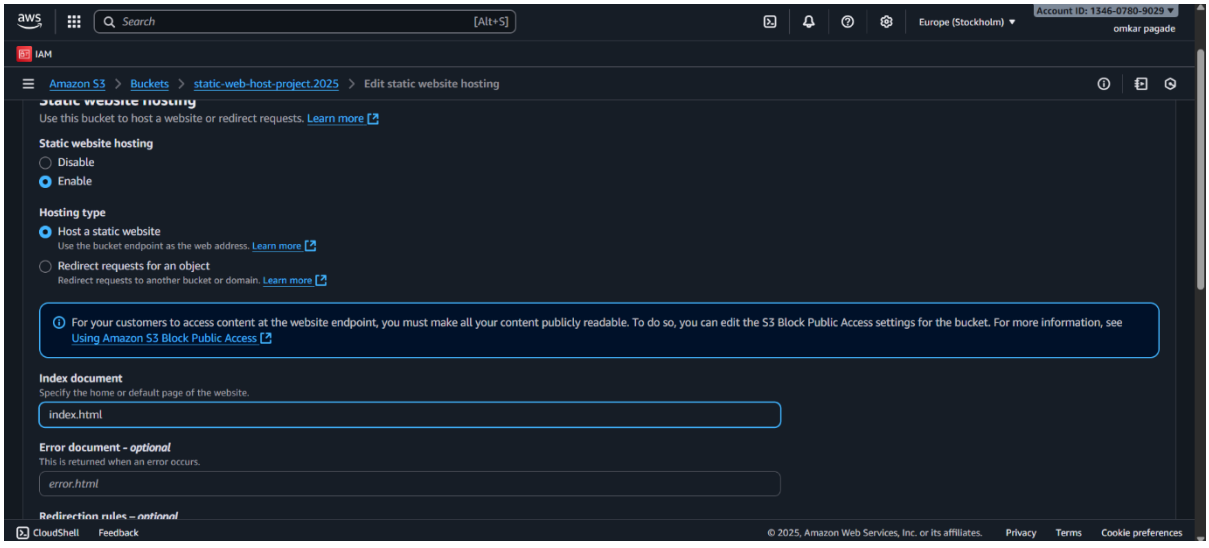
This is the final configuration step. You must enable the static website hosting feature for your S3 bucket.

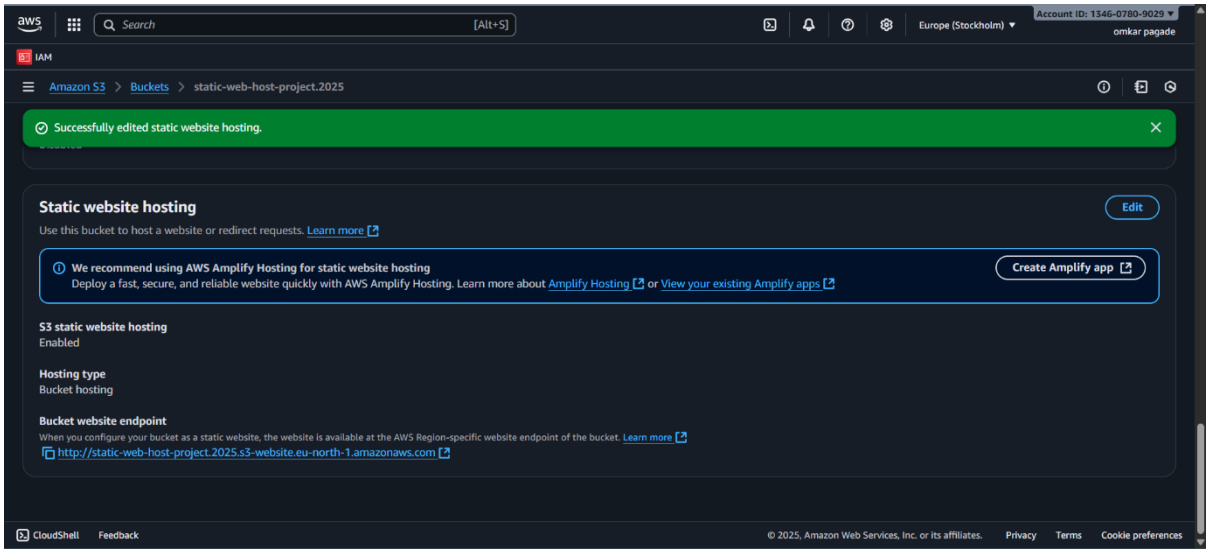
Screenshot 7: Enabling static website hosting.



This screenshot shows the "Edit static website hosting" page. We enable the feature, set the hosting type to "Host a static website," and specify the index.html file as the "Index document." This tells S3 which file to load when a user visits your website's root URL.

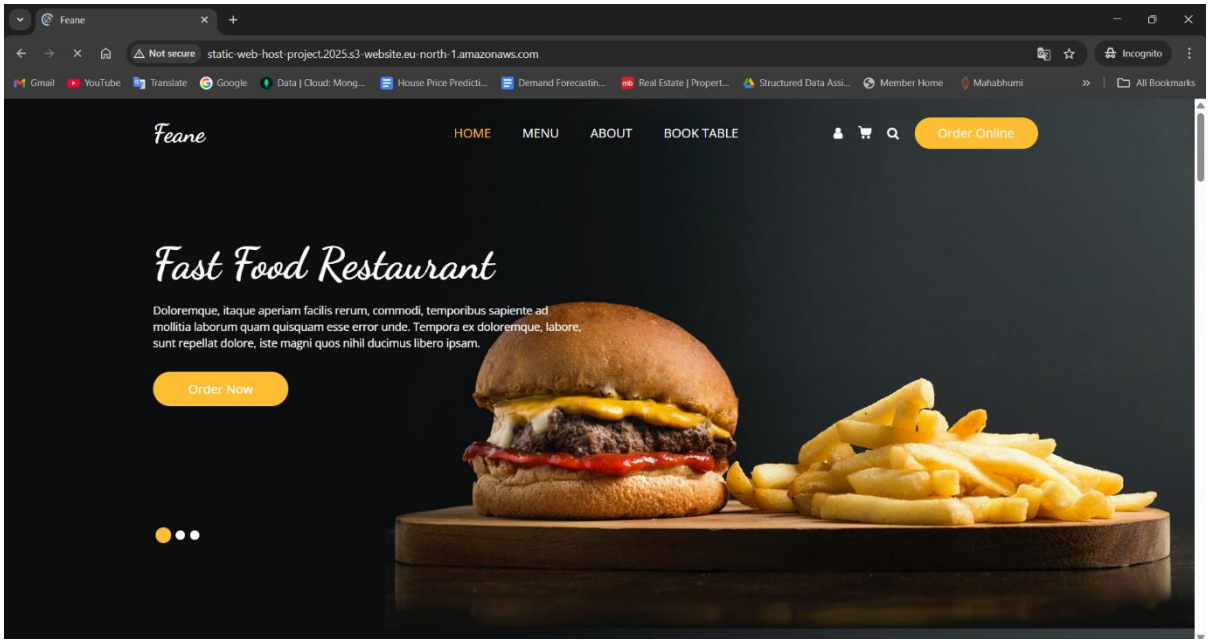
Screenshot 8: Static website hosting enabled.





This screenshot shows the final configuration. The success message confirms that static website hosting has been enabled. The "Bucket website endpoint" URL is provided, which is the public URL for your hosted website. You can share this URL with others to view your website.

Output web was hosted



Conclusion :-

In this project, we successfully demonstrated how to leverage AWS S3 to host a fully functional static website. By following the key steps of creating a bucket, configuring public access, uploading all necessary files, and enabling the static website hosting feature, we created a scalable and cost-effective hosting solution.