

HOST A WEBSITE ON AMAZON S3



AUGUST 23

Devamadhav Kattimuttathu

HOST A WEBSITE ON AMAZON S3

Introduction

In this project, we will walk through the process of creating and configuring an Amazon S3 bucket to host a static website. Amazon S3 (Simple Storage Service) is a highly scalable object storage service provided by AWS, ideal for storing and retrieving any amount of data. Hosting a static website on S3 allows you to serve HTML, CSS, and JavaScript files directly from the cloud, making your website accessible to anyone on the internet.

We will cover all the essential steps, from creating the S3 bucket, uploading your website files, and setting the correct permissions using Access Control Lists (ACLs), to configuring the bucket for static website hosting. By the end of this guide, you will have a fully functional static website hosted on Amazon S3, and you will also learn how to manage and delete your resources to avoid unnecessary costs.

Observations and Screenshots

Step 1: Create an S3 Bucket

We started by creating a new S3 bucket. First, we searched for "S3" in the AWS Management Console and selected the option to create a bucket. We chose a unique bucket name and made sure to pick a region closest to us for optimal performance. We enabled ACLs to manage access, ensuring we could control who can interact with the files we'll be uploading. Before proceeding, we carefully reviewed the settings, especially those related to public access, to make sure everything was configured just right. Finally, we clicked "Create bucket," and just like that, our S3 bucket was ready to use.

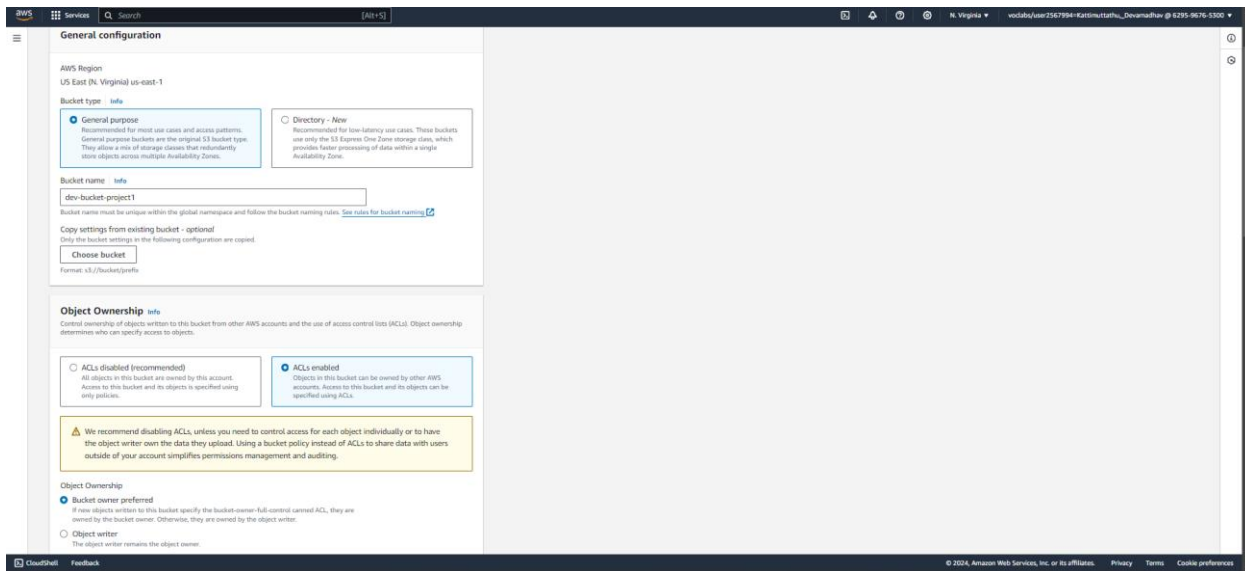


Figure - Creating a new bucket

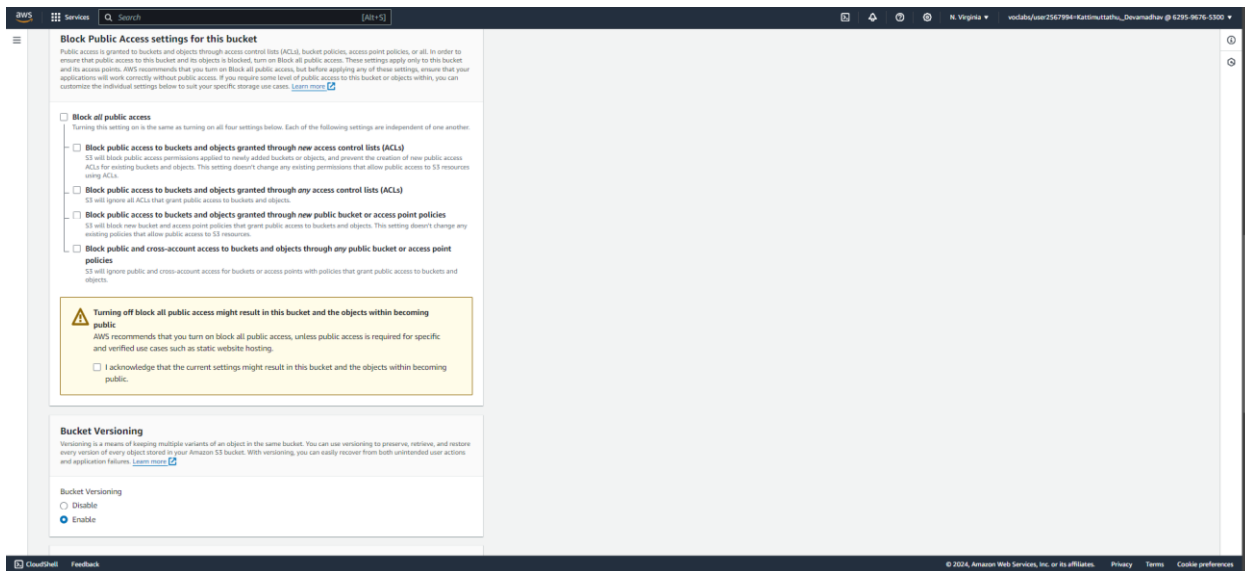


Figure - Making the bucket public

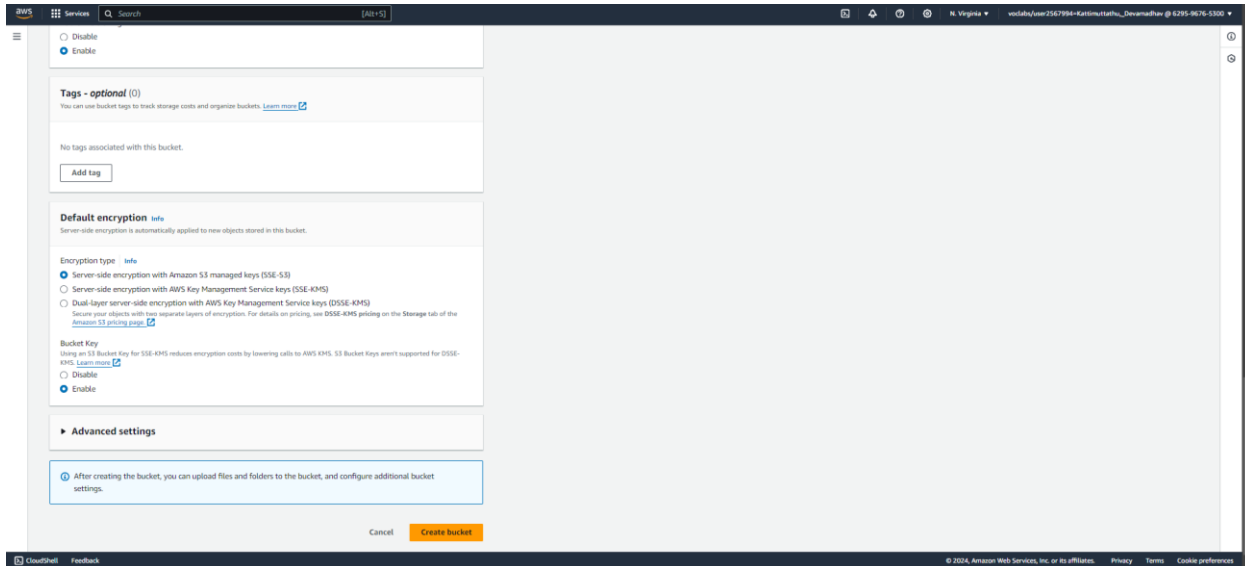


Figure - Kept the rest of the settings as default

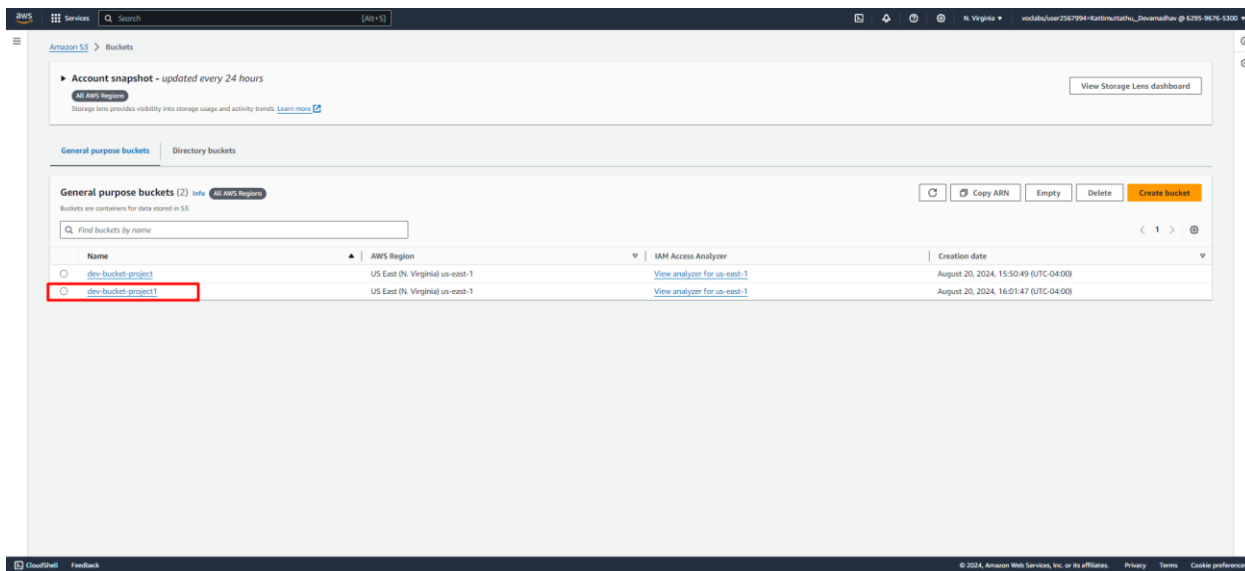
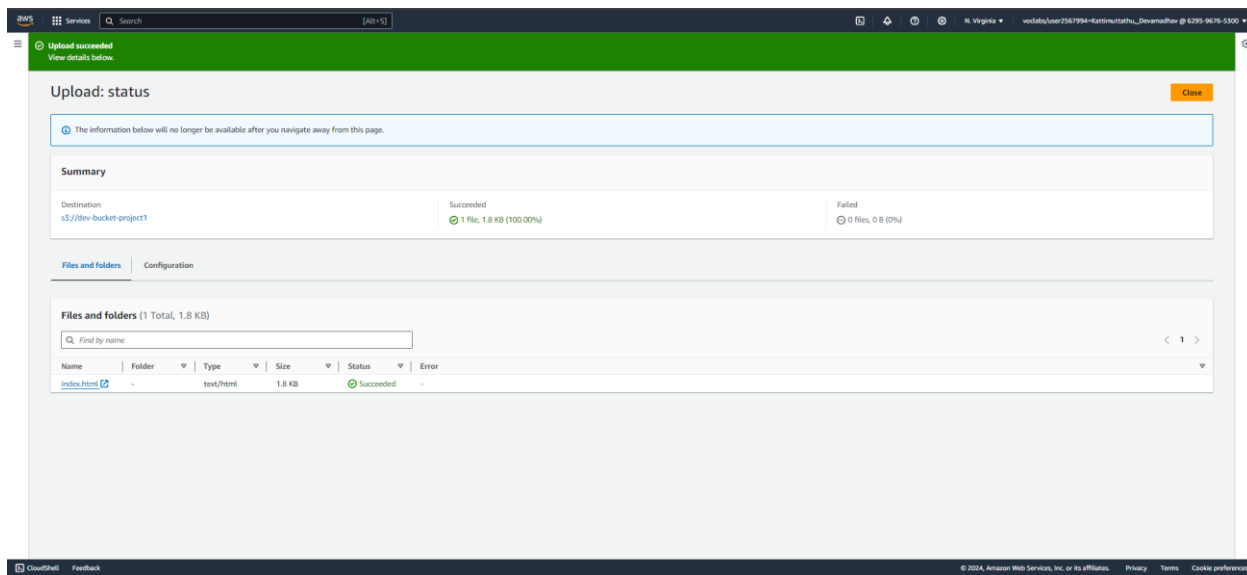


Figure - Successfully created a new bucket

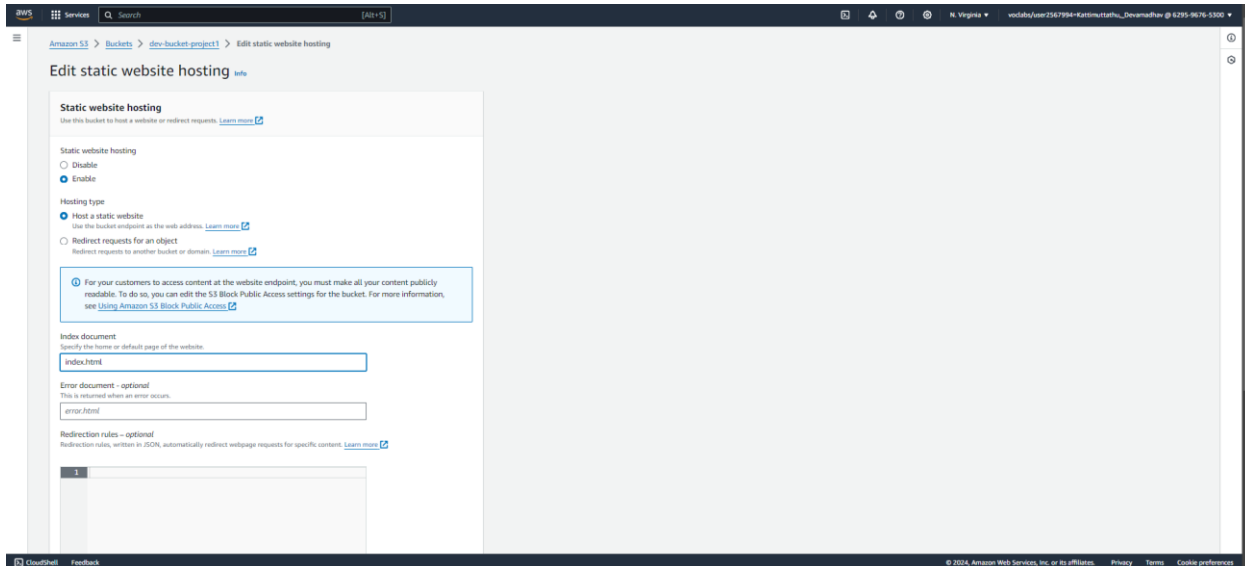
Step 2: Upload Website Content

Next, we moved on to uploading our website content. We navigated to our newly created bucket and selected "Upload" to start adding our files. We made sure to select the correct files, including index.html. After confirming the selection, we initiated the upload process. Watching the progress, we could see our files being securely transferred to the cloud, and in just a few moments, everything was uploaded and ready for the next step.



Step 3: Configure Static Website Hosting

With our files uploaded, we then configured the bucket for static website hosting. We went to the "Properties" tab of our bucket and scrolled down to find the static website hosting section. Here, we enabled the static website hosting feature and set the hosting type to "Host a static website." We specified index.html as the index document, which would be the entry point for our website. After saving the settings, we were excited to see the bucket's endpoint URL, knowing that our website was almost ready to go live.



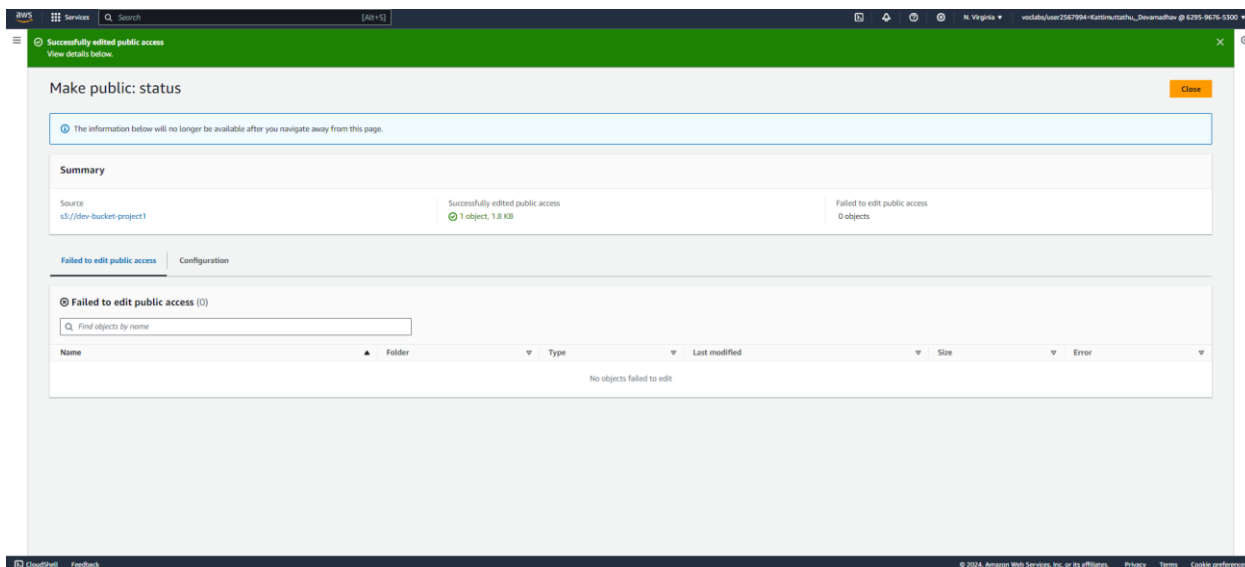
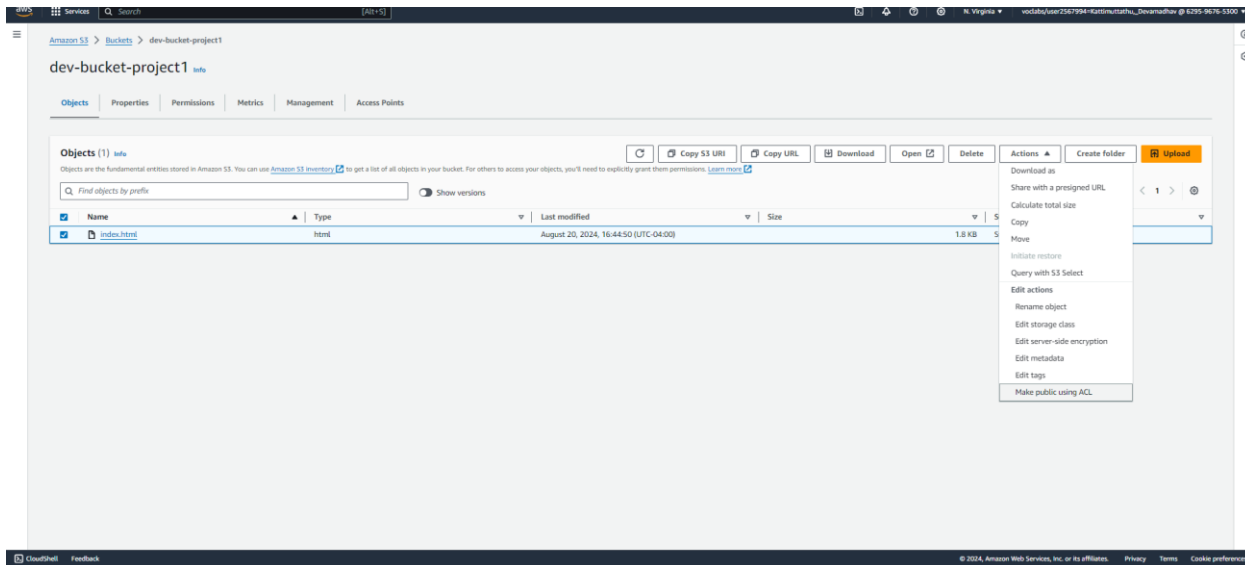
Step 4: Make Objects Public Using ACLs

At this point, we needed to make our uploaded objects public so they could be accessed by anyone visiting our website. We returned to the "Objects" tab of our bucket, selected the checkboxes next to index.html and the folder of assets, and used the ACL settings to make them public. This step was crucial because, without it, our website would remain hidden from the world. After confirming the changes, we felt confident that our website was now ready to be accessed online.

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: S2TQ6DK6G9HC82ZA
- HostId: xdYdjAALTCI69KM2oXSuHUtuCJrYwmdzT5DkA2sknTmi2eZotaPyUJE8Y6eiMTNbONMKsTQq9M=

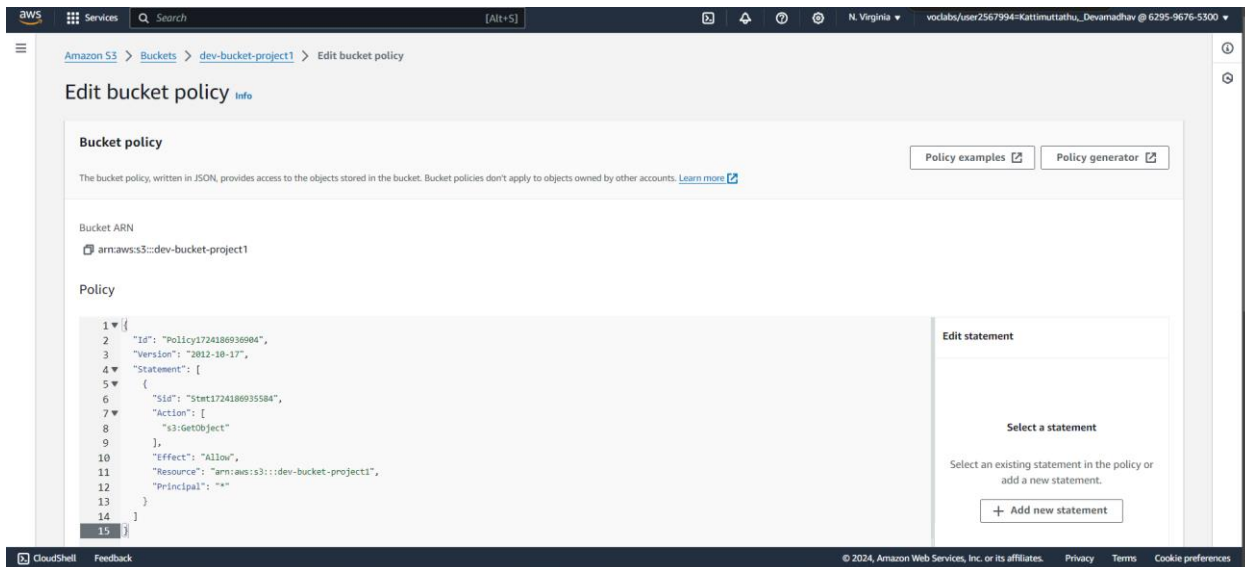
Figure - Error accessing the webpage due to permission issues



Alternate Step: Make the Bucket Public by Setting a Policy in Permissions

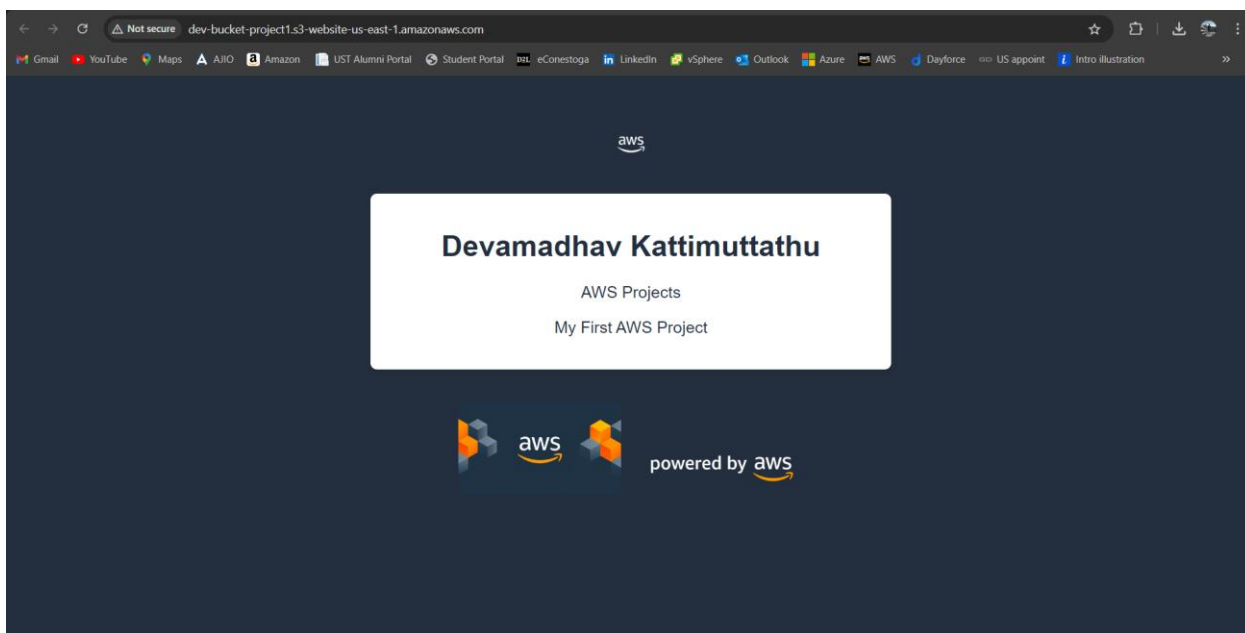
Alternatively, we can make our S3 bucket public instead of using ACLs by setting a bucket policy. This method allows us to manage access to the entire bucket with a single policy, which can be simpler if all the contents need to be public.

To do this, we navigate to the Permissions tab of our bucket and select Bucket policy. Here, we can add a JSON policy that grants public read access to the bucket.



Step 5: Verify Website Hosting

To verify everything was working, we clicked on the URL provided in the static website hosting section. Initially, we encountered an error because our objects were still private. However, after making them public in the previous step, refreshing the page brought our website to life. Thereby, we had successfully set up our static website.



Step 6: Delete Resources

Finally, after confirming that our website was live, we moved on to cleaning up our resources. We returned to the S3 console, selected all the objects in our bucket, and deleted them. We also deleted the bucket itself to ensure we wouldn't incur any unnecessary charges.

