NE
nevilsalu5@gmail.com
NextWork Student

Introducing Today's Project!

What is Amazon S3

Amazon S3 (Simple Storage Service) is a scalable object storage service for storing and retrieving data. It's useful for hosting static websites, backups, and archiving due to its high durability, availability, and cost-effectiveness.

How I used Amazon S3 in this project

Intoday'sproject, I used AmazonS3 to hosta static website by creating a bucket, uploading my files including index.html, and enabling static website hosting. After making the files public, I accessed my website through the generated endpoint URL.

One thing I didn't expect in this project was...

Ididn't expectthe need to adjust permissionsfor uploadedfilesto makethem public. The default privacy setting caused a "403 Forbidden" error, requiring me to learn about ACLs to resolve it.

This project took me...

I completed themain website hosting project in about half an hour. I also explored additional topics like presigned URLs, bucket policies, updating the website, versioning, and custom domain hosting, which took me some extra minutes.

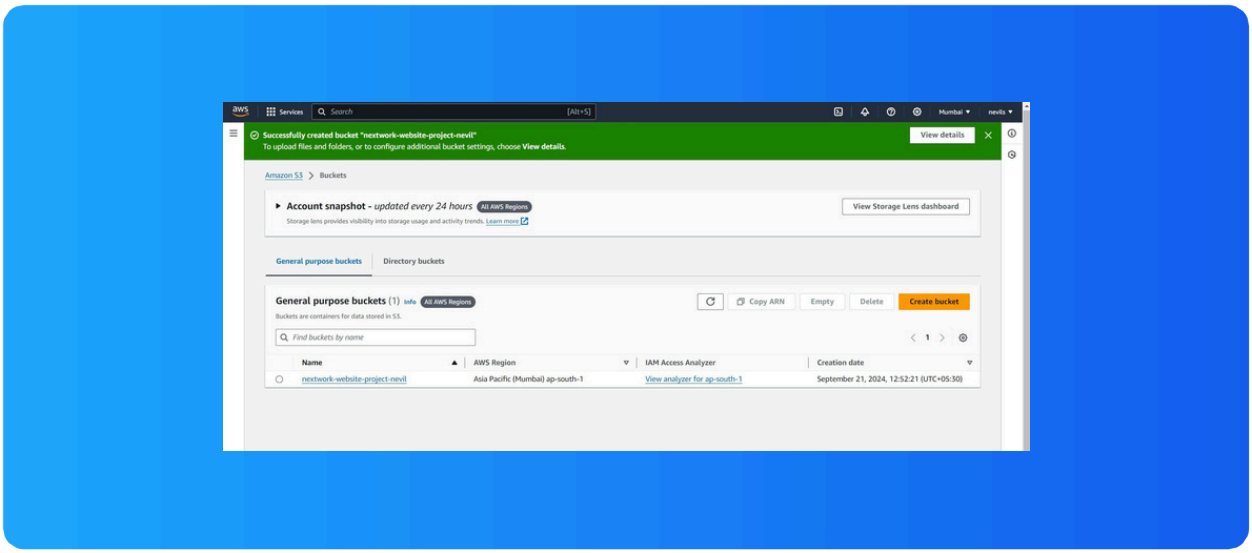
NE
nevilsalu5@gmail.com
NextWork Student

How I Set Up an S3 Bucket

Creating the s3 bucket took me a span of 5 min.

I chose the Hyderabad region for its proximity and compliance with local data residency requirements, ensuring optimal performance for my project.

Understood that bucket names must be unique across all existing buckets in Amazon S3. This uniqueness is crucial for distinguishing my bucket from others in the AWS ecosystem.

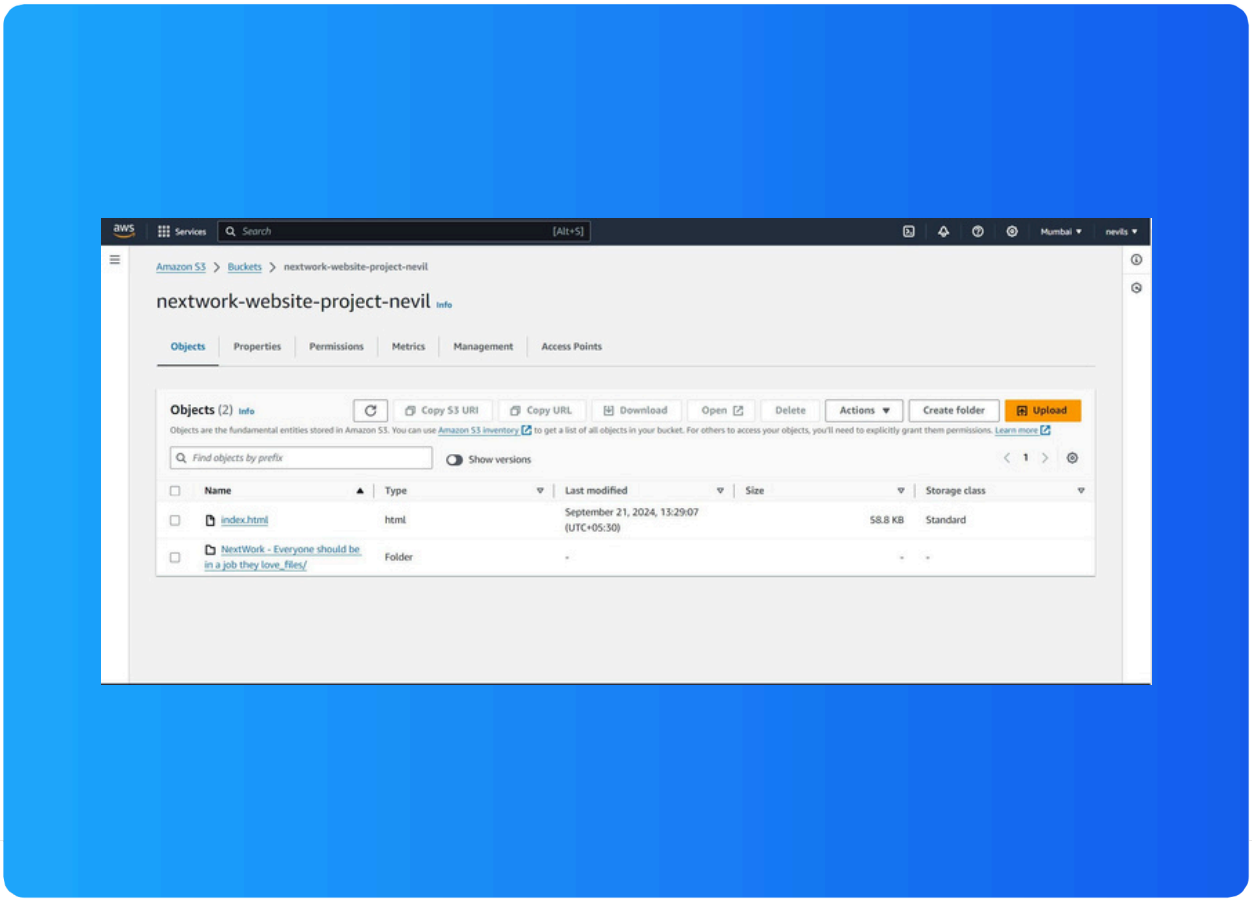


NE
nevilsalu5@gmail.com
NextWork Student

Upload Website Files to S3

I uploaded two files to my S3 bucket - they were an HTML file named index.html, which directed users to the website, and a folder containing various files, including photos and other documents.

Both files were necessary for this project as the index.html file served as the main entry point for the website, while the unzipped folder contained essential resources like images and documents that index.html referenced. Without the contents of the folder, the HTML file would not display correctly.



NE
nevilsalu5@gmail.com
NextWork Student
NextWork.org

Static Website Hosting on S3

Website hosting means storing website files on a server so they can be accessed on the internet. In my project, I used an S3 bucket to host my static website, enabling users to access my index.html file and other resources via a unique URL.

To enable website hosting with my S3 bucket, I selected the bucket in the S3 console, navigated to the Properties tab, and scrolled down to the Static website hosting section. I enabled static website hosting, entered "index.html" as the Index doc.

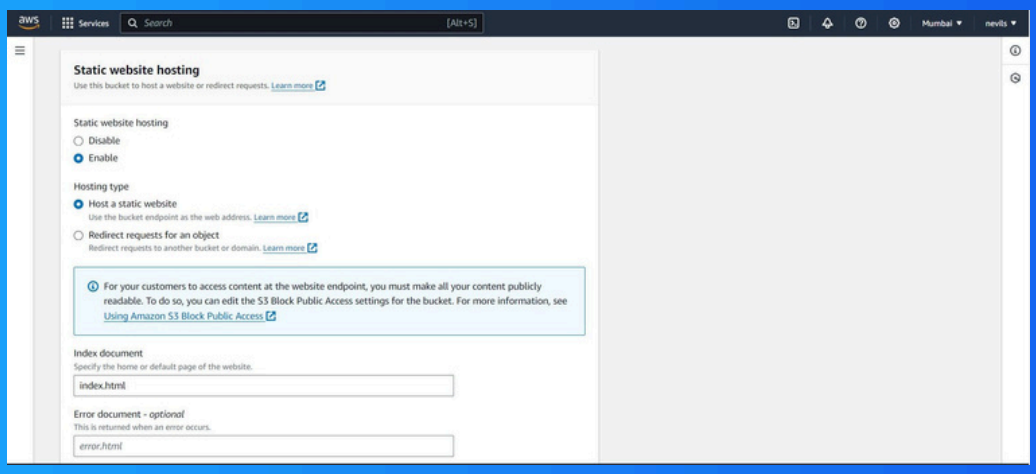
Access Control Lists ACL

I explored Access Control Lists (ACLs) to manage access to the bucket. I understood that enabling ACLs would allow broader access to anyone visiting the website hosted in the bucket.

NE
nevilsalu5@gmail.com
NextWork Student

Bucket Endpoints

Once static website hosting is enabled, S3 produces a bucket website endpoint URL, which is a special URL used to access the hosted website content directly from the S3 bucket.

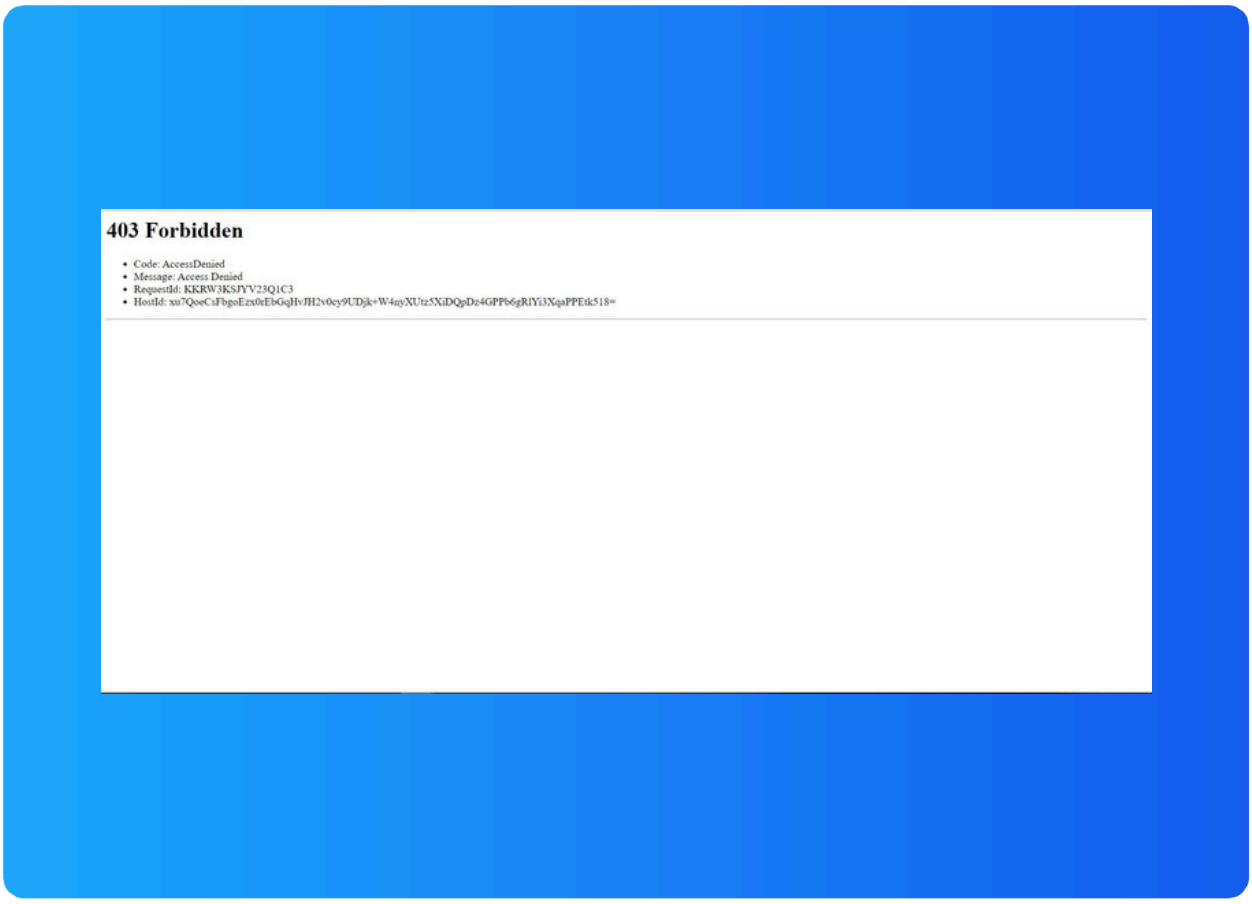


NE
nevilsalu5@gmail.com
NextWork Student

An error!

When I first visited the bucket endpoint URL, I saw a "403 Forbidden" error message. This indicated that I did not have the necessary permissions to access the content in the S3 bucket.

The "403 Forbidden" error occurred because all objects in the S3 bucket are private by default, blocking public access to the uploaded files. To resolve this, I needed to adjust the object permissions to allow public viewing of the website content.

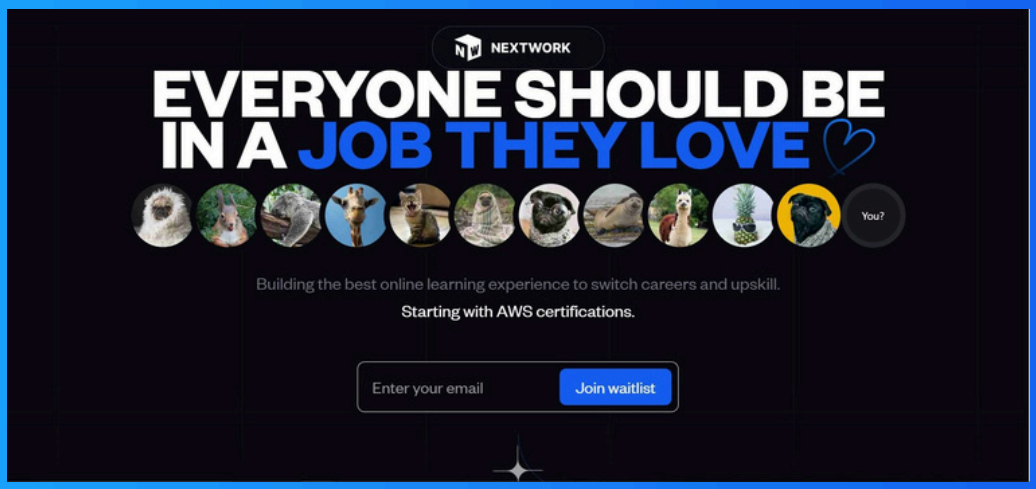


NE
nevilsalu5@gmail.com
NextWork Student

NextWork.org

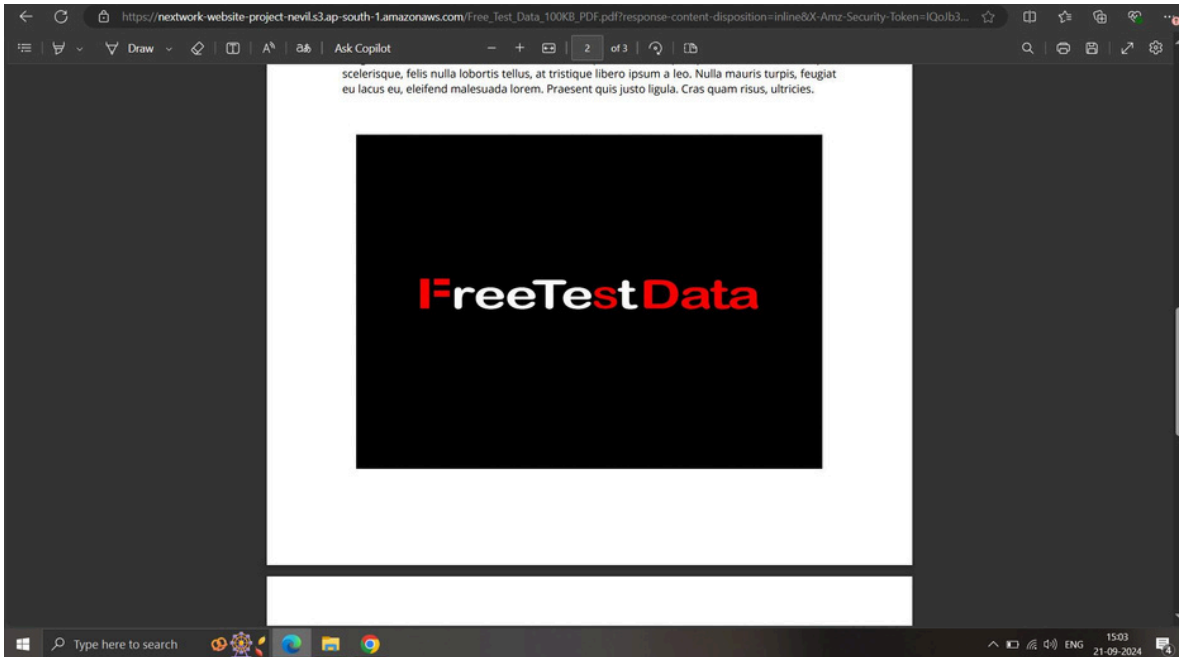
Success!

To resolve this connection error, I opened the Objects tab in my S3 bucket, selected the HTML file and folder, and used the Actions menu to make them public by adjusting the ACLs. This allowed public access and fixed the "403 Forbidden" error.

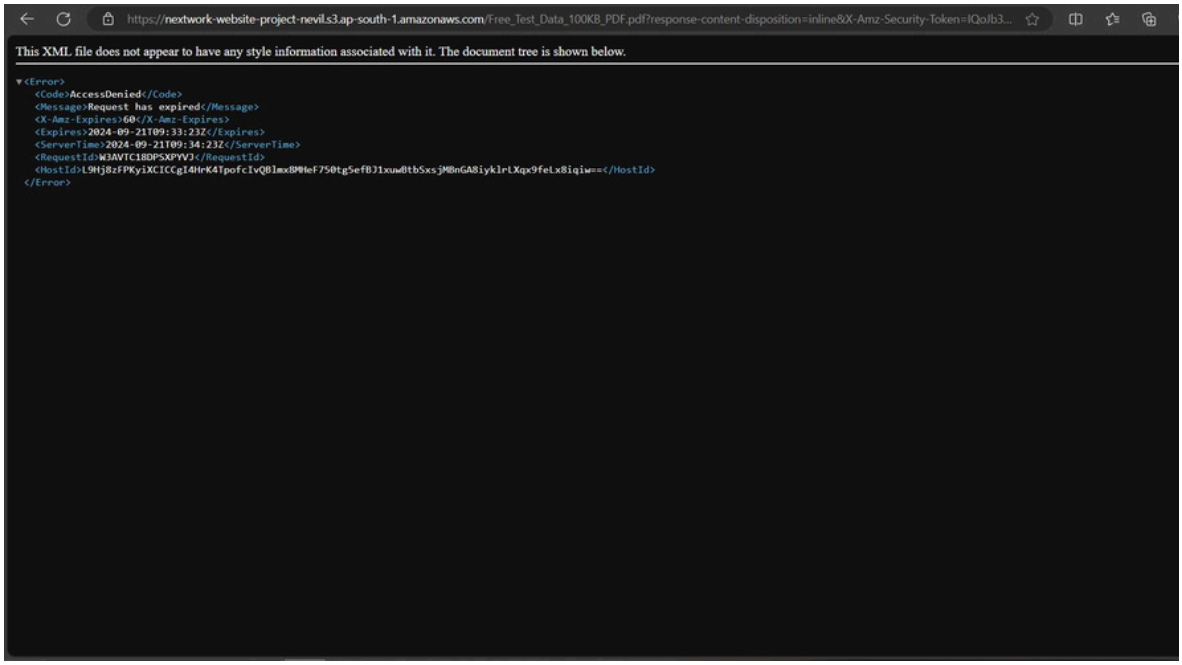


Adding a random pdf file to Explore Presigned URL

I uploaded a random PDF file to the S3 bucket and then selected the option for generating a presigned URL from the Actions menu. I configured the presigned URL to expire in one minute and successfully created the URL, which was copied directly to my clipboard.



Later on after one minute, the webpage showed error.

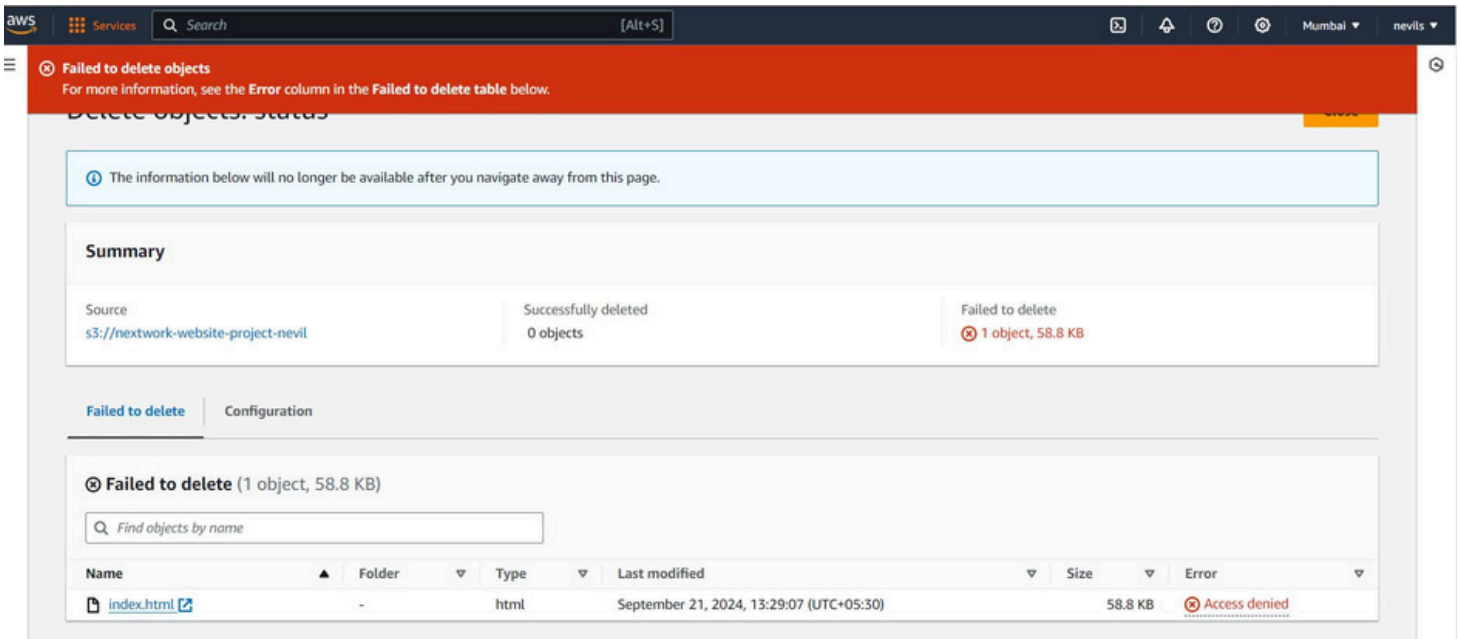


Using bucket policy to secure your bucket

To secure my S3 bucket and prevent the deletion of the HTML file, I implemented a bucket policy. I navigated to the Permissions tab, selected Bucket Policy, and edited it to include the following JSON:json

```
{
  "Version": "2012-10-17",
  "Id": "MyBucketPolicy",
  "Statement": [{
  "Sid": "BucketPutDelete",
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:DeleteObject",
  "Resource": "arn:aws:s3:::<bucket-name>/<object-name>" }]
}
```

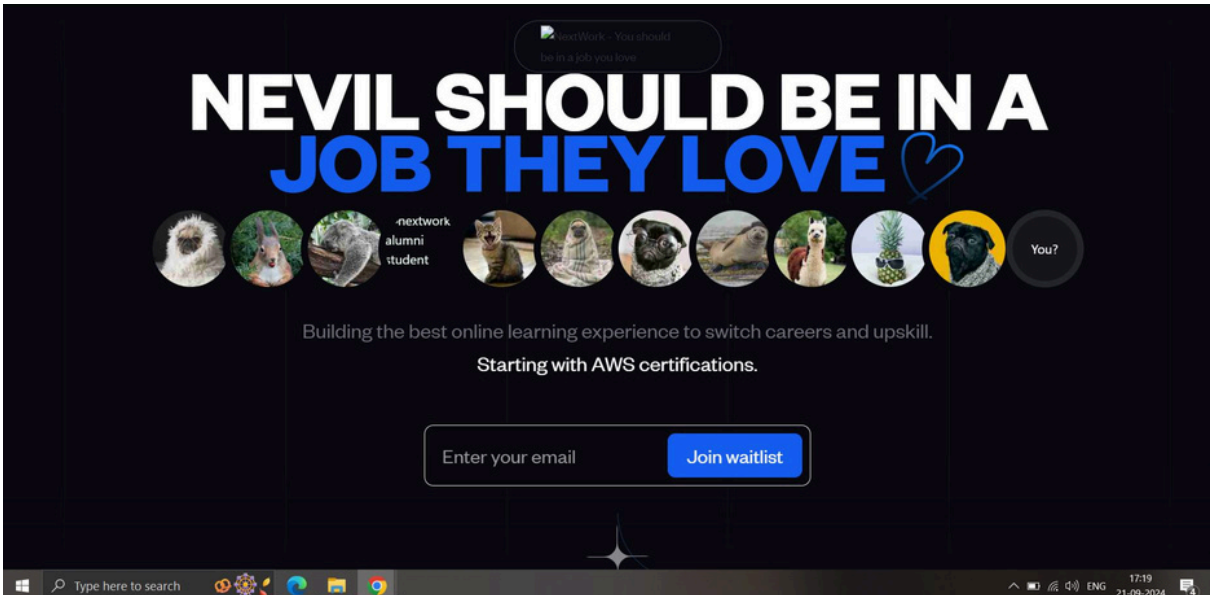
After saving these changes, I confirmed that it was impossible to delete the HTML file due to the updated bucket policy.



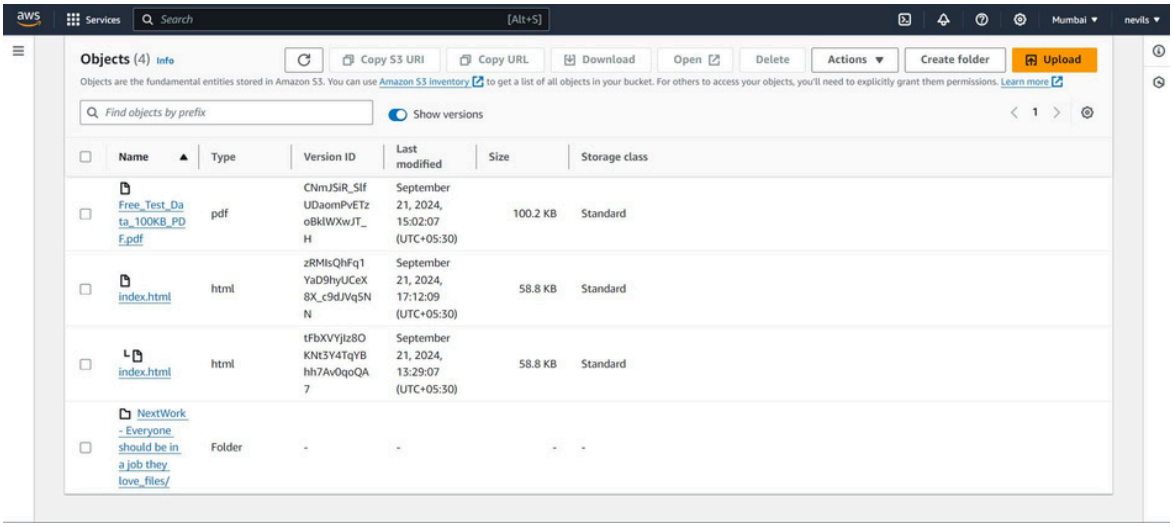
Updating the website itself

To update the HTML file, I opened it in a text editor on my PC to make the necessary changes. To reflect these updates on the website hosted in the S3 bucket, I simply uploaded the modified file without deleting the previous version. However, after uploading the updated HTML file, I encountered a "403 Forbidden" error when visiting the website. This indicates that the permissions need to be updated again to allow public access to the newly uploaded file.

Refreshing after making it public using ACLs;

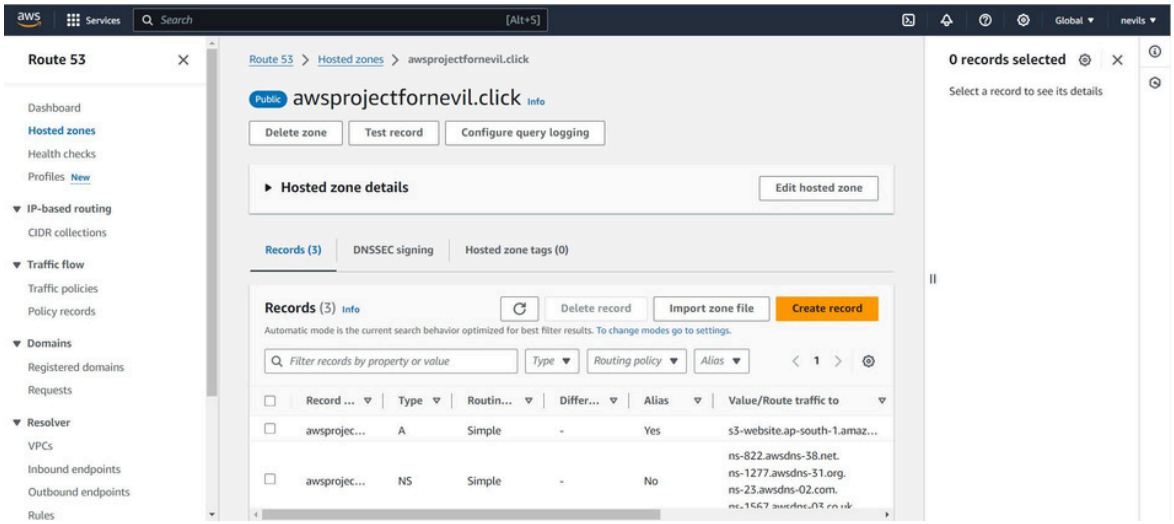


It was able to see all versions of files, after bucket versioning .



Hosting my own website on my own special domain.

- Accessed Route 53: I opened the AWS Management Console and navigated to Amazon Route 53.
- Registered Domain: I chose "Domain registration" and registered a new domain, selecting a cost-effective option.
- Created Hosted Zone: After the domain registration, Route 53 automatically created a hosted zone for the domain.
- Configured DNS Records: I selected the hosted zone for my domain.
    - I created a new record set with the following details: Record Type: A (for IPv4)
    - Alias: Enabled and selected "Alias to S3 website endpoint."
    - Region: Set to Mumbai (ap-south-1).
    - Bucket Name: Ensured it matched my custom domain name.
- Finalized Setup: I saved the record set and waited for the DNS changes to propagate.
- Tested Domain: I accessed the custom domain in a web browser to confirm that it correctly routed to the S3 bucket, successfully running my website.



Domain working: