

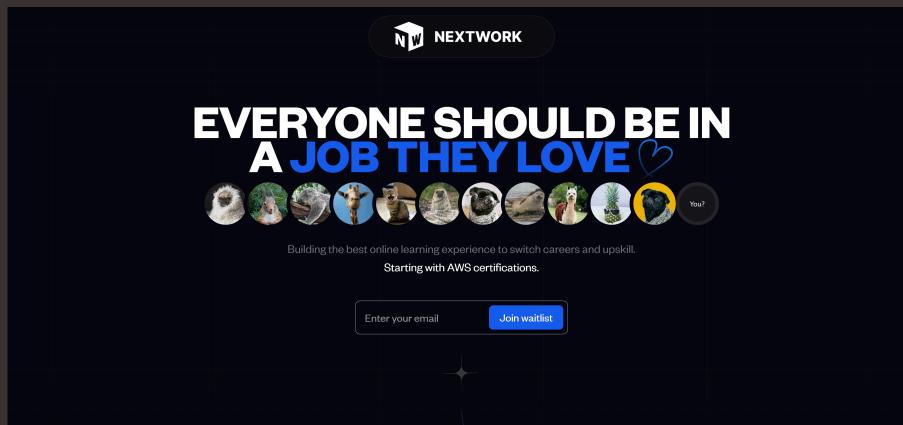


[nextwork.org](http://nextwork.org)

# Host a Website on Amazon S3

M

Muhammad Abbas



# Introducing Today's Project!

In this project, I will demonstrate how to use S3 to host a static website. I'm doing this project to learn about AWS and cloud services and how they can be used to store objects in the cloud AND even host websites (how does that work?!)

## Tools and concepts

Services I used were Amazon S3. Key concepts I learnt include bucket policies, uploading static website files, index.html, bucket endpoint URLs and ACLs and how they control access to the bucket's objects.

## Project reflection

This project took me approximately 45 minutes including quiz time and secret mission time. The most challenging part was resolving the 403 Forbidden error. It was most rewarding to see my webpage load live and be public to the world!



# How I Set Up an S3 Bucket

Creating an S3 bucket took me less than 5 minutes - I needed to learn a few new concepts like block public access and ACLs, but once the learning is done, I can create buckets in even shorter time in the future.

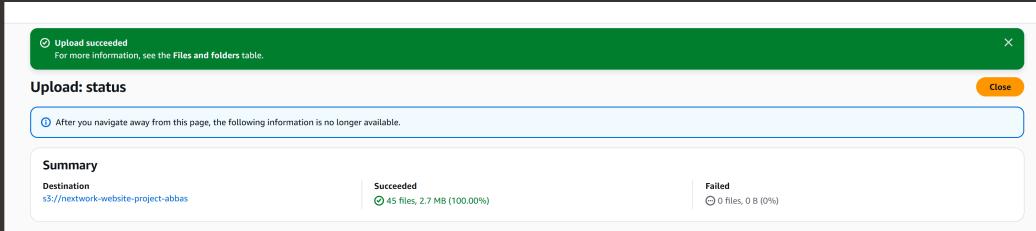
The Region I picked for my S3 bucket was Singapore, because it's the region that's closest to me. It's best practice to pick the region closest to you because it lowers time to retrieve your things (aka latency), and costs (although this is not important for this project)

S3 bucket names are globally unique! This means no two buckets Amazon S3 buckets in the entire world can have same name. They have to completely unique, regardless of the region or the account ID.

M

**Muhammad Abbas**  
NextWork Student

[nextwork.org](http://nextwork.org)



# Upload Website Files to S3

## index.html and image assets

I uploaded two files to my S3 bucket - they were an index.html file (this determines the structure i.e. what goes inside the website) and a folder of images and assets (this will fill in the website with images and things to look at)

Both files are necessary for this project as index.html determines the structure, but the structure alone does not illustrate the contents of the website, i.e. if index.html says "insert image here", it might not have the actual image to display.

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected. Below the navigation bar, there's a toolbar with buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. A prominent orange 'Upload' button is also visible. A search bar and a 'Show versions' checkbox are located below the toolbar. The main area displays a table of objects. The columns in the table are: Name, Type, Last modified, Size, and Storage class. There are two entries:

| Name  | Type   | Last modified                          | Size    | Storage class |
|---|--------|--|---------|---------------|
| index.html  | html   | July 31, 2025,<br>20:04:39 (UTC+08:00) | 58.8 KB | Standard      |
| NextWork -<br>Everyone should be in<br>a job they love_files/ | Folder | -                                      | -       | -             |



# Static Website Hosting on S3

Website hosting means putting our website files on a web server, which is a special computer designed to turn the files into a website page that people can visit.

To enable website hosting with my S3 bucket, I went into the Properties tab of my bucket, enabled static website hosting and I also labelled "index.html" as my index document i.e. this is the document that I am trying to host.

An ACL is (aka is an Access Control List) is a way to configure permission settings inside a bucket. We enrolled ACLs so that we can control access to our website files later. There was a popup mentioning that AWS recommends disabling ACLs, but keep it enabled to learn how ACLs work and compare it with bucket policies later.



**Edit static website hosting** Info

**Static website hosting**

Use this bucket to host a website or redirect requests. [Learn more](#)

**Static website hosting**

- Disable  
 Enable

**Hosting type**

- Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

- Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

**For your customers to access content at the website endpoint, you must make all your content publicly readable.** To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

**Index document**

Specify the home or default page of the website.

index.html



# Bucket Endpoints

Once static website is enabled, S3 produces a bucket endpoint URL, which is a URL that takes you (and anyone on the internet) to the website that you're hosting!

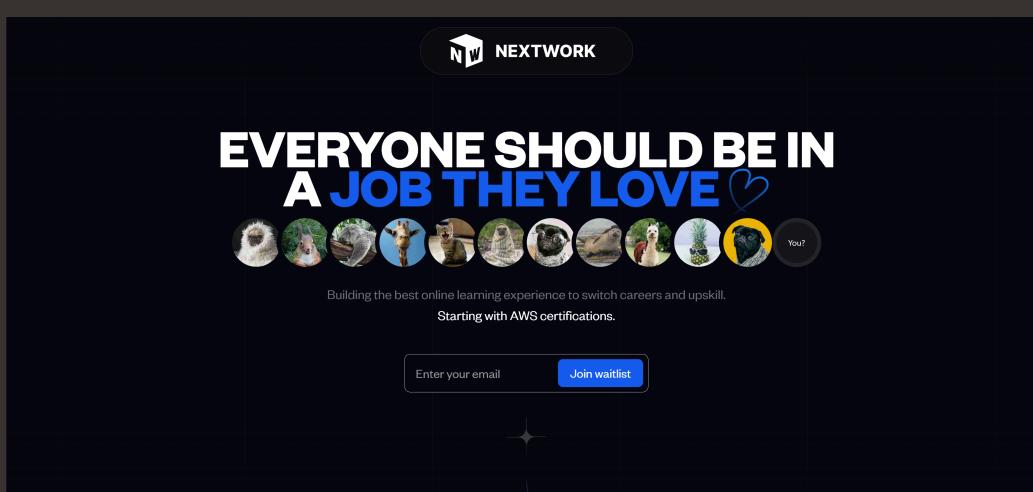
When I first visited the bucket endpoint URL, I saw a 403 Forbidden error! The reason for this error was that objects in a bucket are public by default - even though we've switched off "Block all Public Access" the website files are still completely private. We need to manage their access settings separately - they need to be public files too for the public to see the content of my website

## 403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: WDZJZXPKP0B533GG
- HostId:  
wkZo0i5d0LinsVQXBF7IIL4ETrZ5Bkxspj4afTtcK8tsAWQmBLDgh3ZzgHg0GbgIijtDmcU04NnyLwN0N1VwtLptxzMuvY+ugI1KpZUGqXY=

# Success!

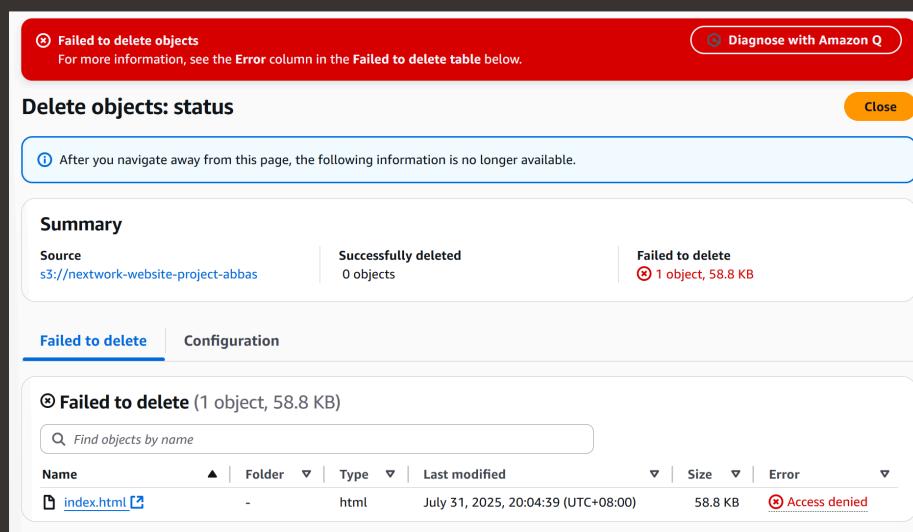
To resolve this 403 Forbidden error, I updated the access settings of the files inside the bucket too. Using ACLs, I made the bucket's files public! Once I checked the S3 endpoint, I can see a website page all loaded up :)



# Bucket Policies

An alternative to ACLs are bucket policies, which are rules that determine who is allowed (or Not allowed) to do something. The benefit of using bucket policies is that you can have even greater control of the ACTIONS that people are/aren't allowed to do, while ACLs are useful for controlling public access to individual objects inside the bucket.

My bucket policy denies everyone from deleting the index.html file in the bucket. I tested this by trying to delete index.html and I saw a permission denied error! This means the bucket policy worked, it successfully stopped me from deleting the object I wanted to protect.





[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

