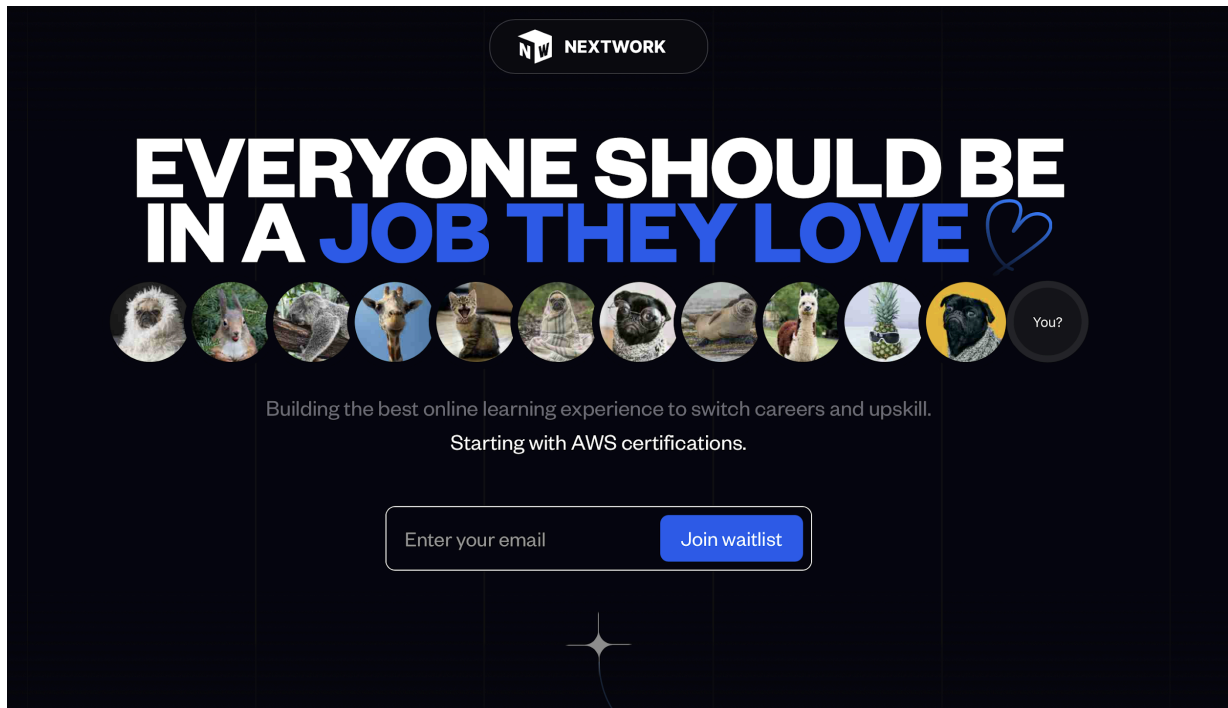


Host a Website on Amazon S3

Erik Gonzalez



Introducing Today's Project!

In this project, I will demonstrate how to use S3 to host a static website. I am doing this project to learn more about AWS and cloud services and how they can be used to store objects in the cloud and even host websites.

Tools and concepts

The services I used were Amazon S3. Key concepts I learned include bucket policies, uploading static website files, index.html, bucket endpoint URLs, ACLs and how they control access to my bucket objects.

Project reflection

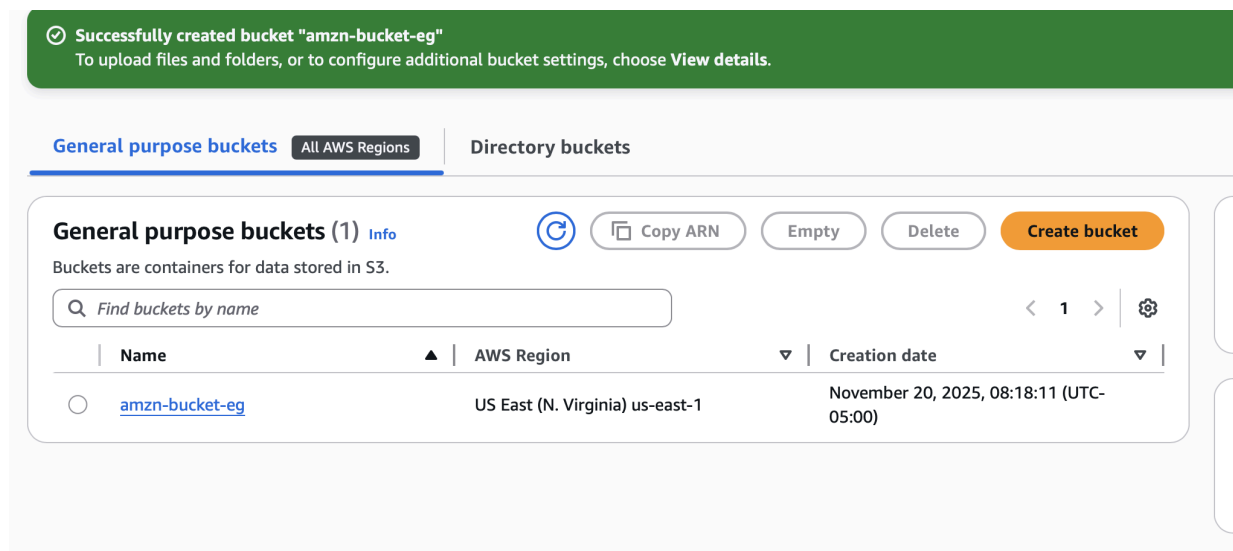
This project took me approximately one and half hours including demo time, quiz time, and secret mission time. The most challenging was trying to remedy the 403 forbidden error. It was most rewarding to see the webpage go live.

How I Set Up an S3 Bucket

Creating an S3 bucket took me less than 5 minutes. I needed to learn a few more concepts like block public access and ACL's. But once that learning is done, I can create buckets in even shorter time in the future.

The Region I picked for my S3 bucket was Virginia because it is the region that is closest to me. It is best practice to pick the region closest to you because it lowers time to retrieve your things (latency), and costs.

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



Upload Website Files to S3










index.html and image assets

I uploaded two files to my S3 bucket - they were an index.html file that determines the structure-- a blueprint of sorts, and a folder of images and assets. This will fill 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 context of the website, i.e. if html says insert image here, it may not have an image to display. You need to supply.



amzn-bucket-eg Info

[Objects](#) | [Metadata](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2)         

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)



<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 index.html	html	November 20, 2025, 08:32:12 (UTC-05:00)	58.8 KB	Standard
<input type="checkbox"/>	 NextWork - Everyone should be in a job they love_files/	Folder	-	-	-

Static Website Hosting on S3

Website hosting means putting the website files on a web server which is a 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 the bucket and enabled static website hosting and also labeled "index.html" as our index document (this is the document I am trying to host).

An ACL (Access Control List) is a way to configure permission settings inside a bucket. ACL is enabled to control access to a website's files 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.

Error document - optional

This is returned when an error occurs.

Redirection rules - optional

Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#) [↗](#)

Bucket Endpoints

Once a static website is enabled, S3 produces a bucket endpoint URL, which is a URL that takes you or anyone on the internet to the website that I am hosting.

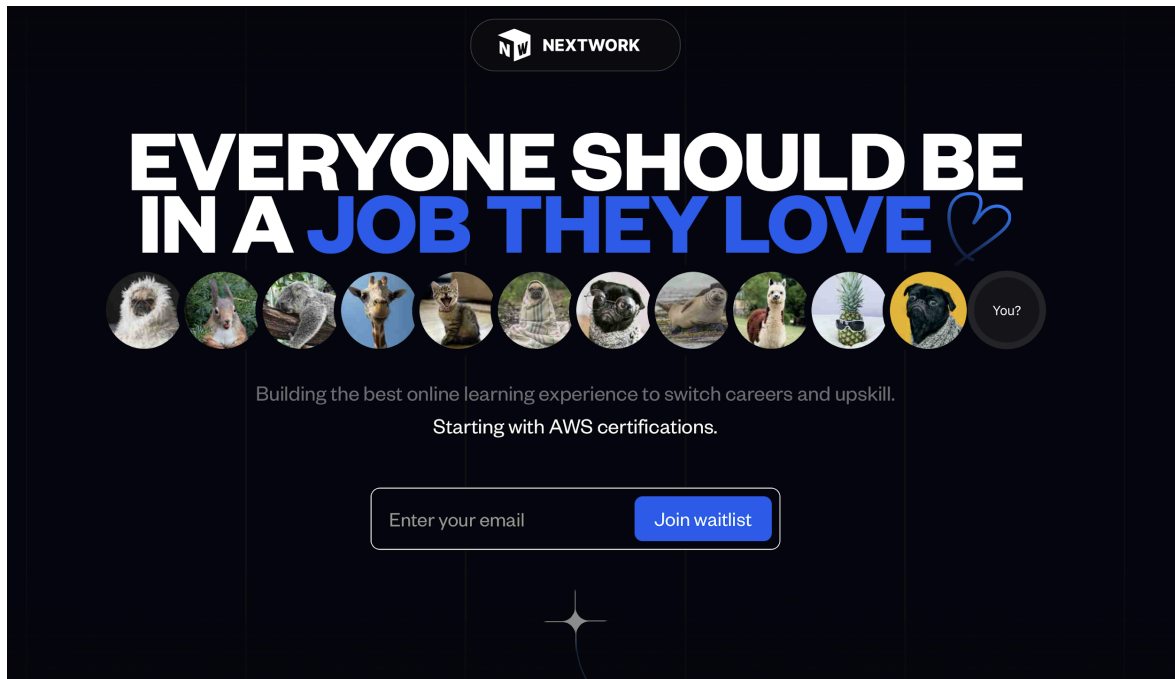
When I first visited the bucket endpoint URL, I saw a 403 forbidden error! The reason for this error is because objects in a bucket are public by default- even though I switched off "block all public access", the website files are still private.

403 Forbidden

- Code: AccessDenied
 - Message: Access Denied
 - RequestId: BYZEW6HE18MZYW83
 - HostId: 1J+zwLaXyFNywnmptOhADBD/x7QzDcgWK5FjnDWcpEnDxHmTruVtY1HIhOYxbE/U8LKuSWIpVBw=
-

Success!

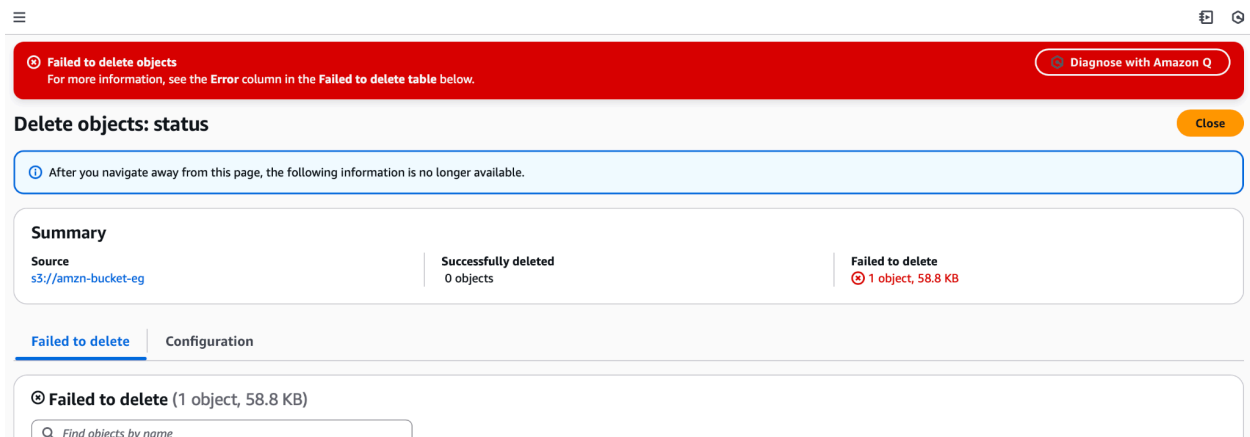
To resolve this 403 Forbidden error, I updated the access settings of the files inside the bucket. Using ACL I made the bucket files public. Once I checked the S3 bucket endpoint, I can see a webpage all loaded up!



Bucket Policies

In this project extension, I'm about to use bucket policies to control access to our bucket files. I am doing this to stop people from deleting the objects inside the file.

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 greater control over the actions that people are or aren't allowed to do while ACLs are useful for controlling public access to individual objects inside the bucket.



Failed to delete objects
For more information, see the Error column in the Failed to delete table below. [Diagnose with Amazon Q](#)

Delete objects: status [Close](#)

After you navigate away from this page, the following information is no longer available.

Summary		
Source s3://amzn-bucket-eg	Successfully deleted 0 objects	Failed to delete 1 object, 58.8 KB

[Failed to delete](#) [Configuration](#)

Failed to delete (1 object, 58.8 KB)

Find objects by name

My bucket policy denies everyone from deleting the index.html file in the bucket. I tested this by trying to delete the index.html file and I saw a red permissions denied error. This means the bucket policy worked. It successfully stopped us from deleting objects we wanted to protect

```
{
  "Version": "2012-10-17",
  "Id": "MyBucketPolicy",
  "Statement": [
    {
      "Sid": "BucketPutDelete",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:DeleteObject",
      "Resource": "arn:aws:s3:::amzn-bucket-eg/index.html"
    }
  ]
}
```


1
}