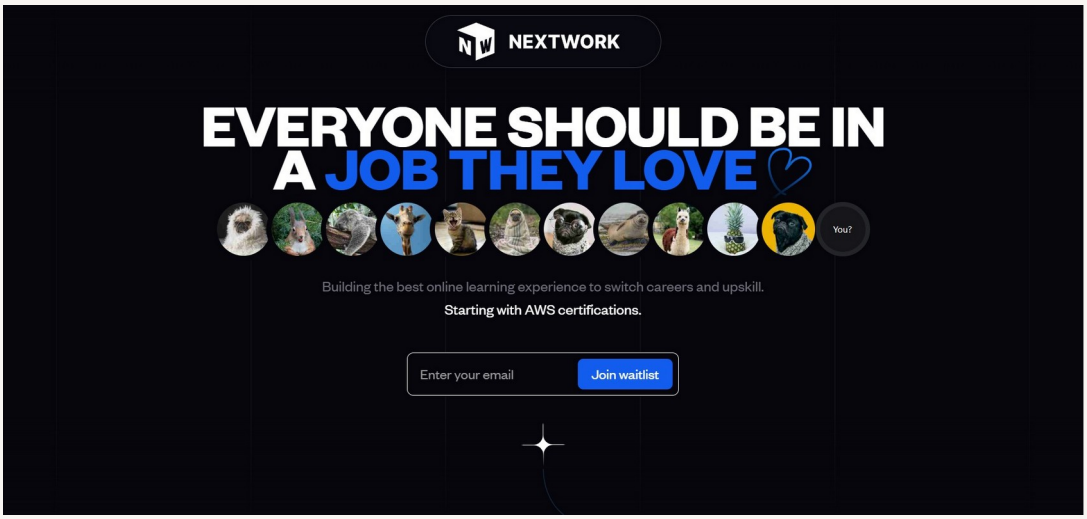
Host a Website on Amazon S3

Melvin green



Introducing Today's Project!

Project overview

In this project, I will demonstrate how to use S3 to host a static website. I am 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 a website.

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, ACLs, and how they control access to my bucket objects

Time, challenges, and wins

This project took me approximately 45 minutes, including demo time and documentation. The most challenging part was the 403 Forbidden error. It was most rewarding to see the website live and be public!

How I Set Up an S3 Bucket

What I did in this step

In this step, I will open up S3 and then create a storage space inside to start storing website files.

How long it took to create the bucket

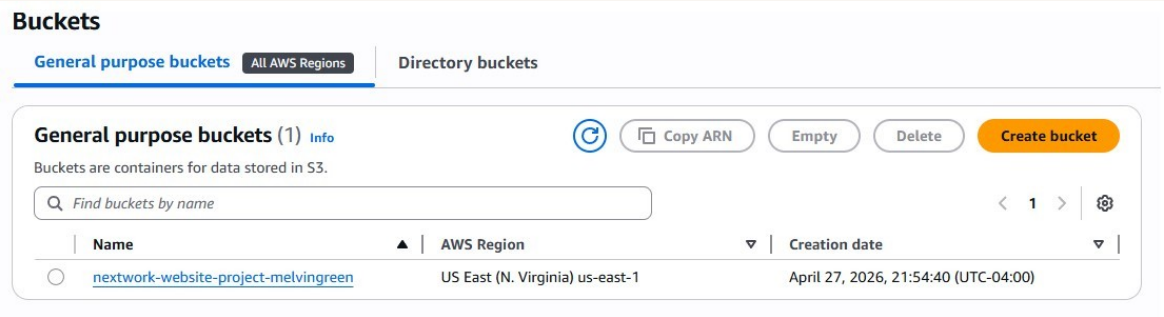
Creating an S3 bucket took me less than 5 minutes. I needed to learn a few new concepts, like block public access and ACLs, but once learning is done, we create buckets in an even shorter time in the future

Region selection

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

Understanding bucket name uniqueness

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



Upload Website Files to S3

What I did in this step

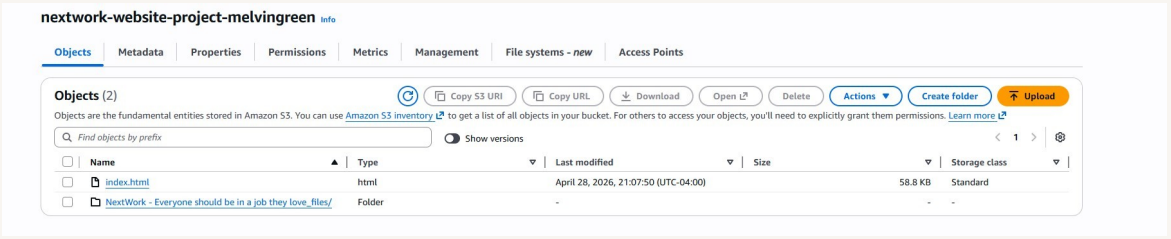
In this step, I will upload the website files to my S3 bucket. This is important because there are no files = no website! My bucket is still empty. I will get the files inside, so I have a website to host!

Files I uploaded

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.

How the files work together

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 index.html says" insert image here", it might not have the actual image to display - I would need to supply that image separately. That's why we have multiple files uploaded - the index.html file to direct what is inside the website page, but also a bunch of assets (like images) that the website is trying to display.


nextwork-website-project-melvingreen Info
Objects
Metadata
Properties
Permissions
Metrics
Management
File systems - new
Access Points
Objects (2)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
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
Find objects by prefix
Show versions
1
Name
Type
Last modified
Size
Storage class
index.html
html
April 28, 2026, 21:07:50 (UTC-04:00)
58.8 KB
Standard
NextWork - Everyone should be in a job they love_files/
Folder
-
-
-

Static Website Hosting on S3

What I did in this step

In this step, I will make my website available to view! This is called static website hosting, and it's important because our website files will stay as 'just files' and not turn into a website without this step.

Understanding website hosting

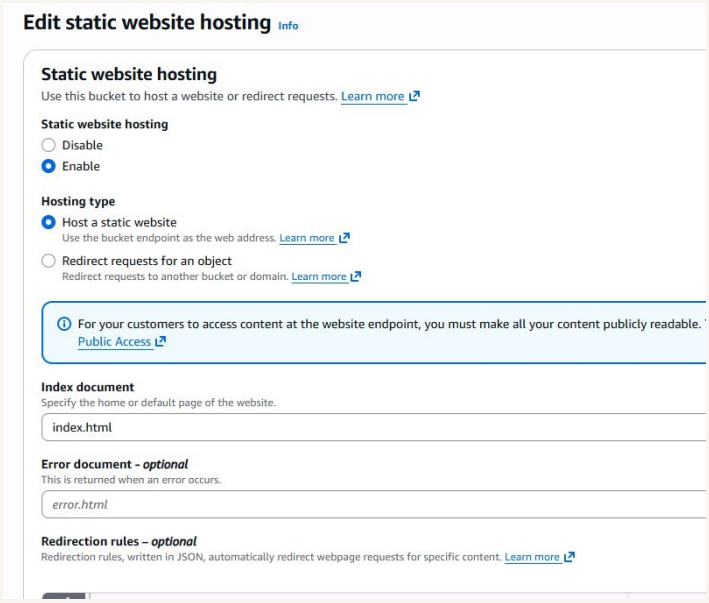
Website hosting means putting my website files on a website server, which is a computer designed to turn my files into a website page that people can visit.

How I enabled website hosting

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

Access Control Lists (ACLs)

An ACL (Access Control List) is a way to configure permission settings within a bucket. We enabled ACLs so that we can control access to our website files later. There was a pop-up stating that AWS recommends disabling ACLs, but to keep them enabled to learn how ACLs work and compare them 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.
Public Access
Index document
Specify the home or default page of the website.
index.html
Error document - optional
This is returned when an error occurs.
error.html
Redirection rules – optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more

Bucket Endpoints

Understanding bucket endpoint URLs

Once a 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 is being hosted!

What I saw when I tested the endpoint

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

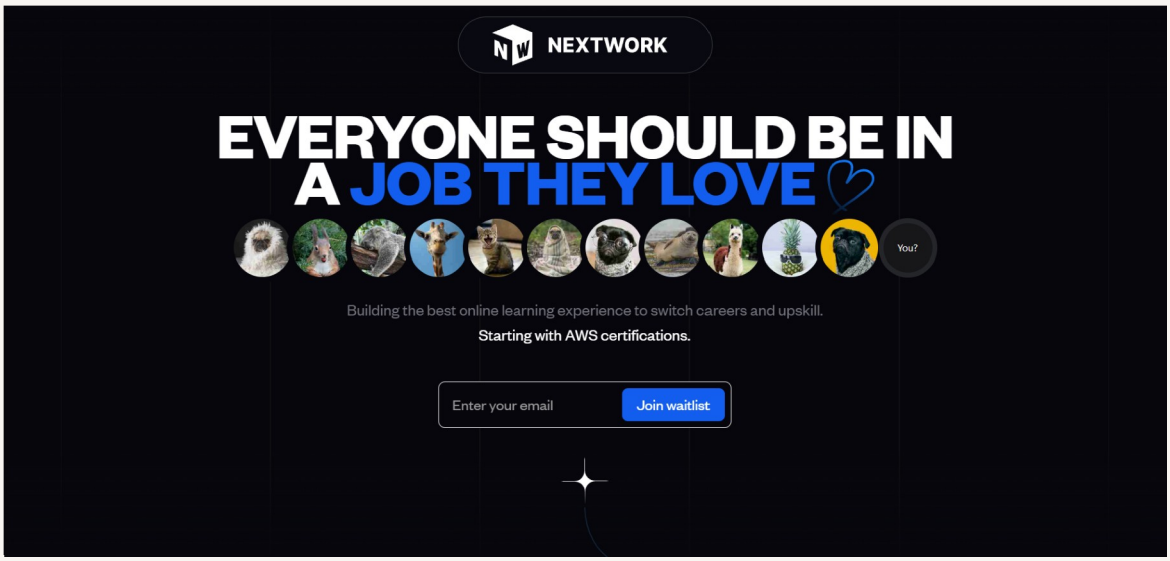
Success!

What I did in this step

In this step, I will go into the files that I uploaded to my bucket and make them public, too. because this will enable us to actually view the contents of the website. Once this is done, the website is officially live!

How I resolved the 403 error

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



Bucket Policies

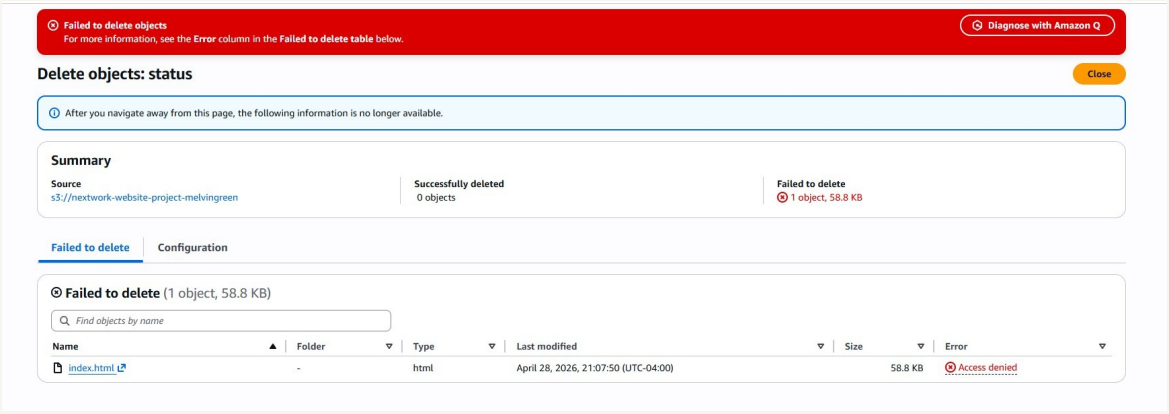
What I did in this extension

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

Understanding bucket policies

An alternative to ACLs is 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 access to the actions that people are or aren't allowed to do. While ACLs

are useful for individual access, for controlling public access to individual objects inside the bucket.



What my bucket policy does

My bucket policy denies everyone from deleting our index.html file in the bucket.

I tested this by trying to delete index.html and saw a permission denied error! This means the bucket police worked. It successfully stopped me from deleting objects I wanted to protect.