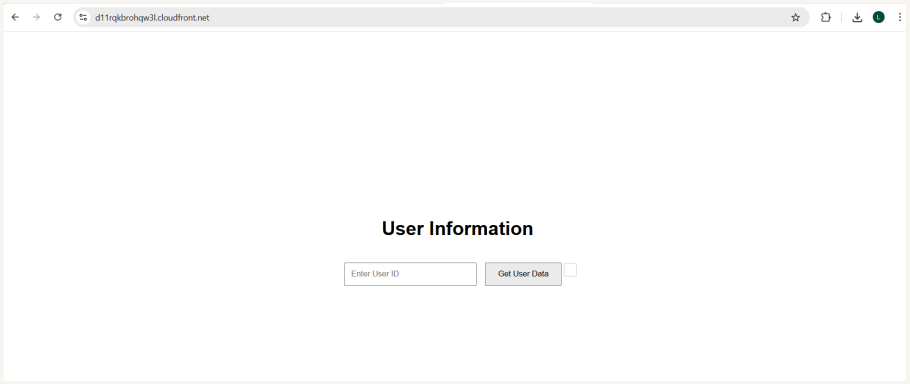
nextwork.org

Website Delivery with CloudFront

Louis Moyo



L
Louis Moyo
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to deploy a website using Amazon CloudFront. I'm doing this project to learn how to improve website speed, reliability, and scalability with AWS. I will store files in S3, set up CloudFront for global distribution, configure permissions, and compare hosting methods. This focuses on the presentation tier of a three-tier architecture, essential for roles like Cloud Network Engineer, DevOps, or Frontend Engineer.

Tools and concepts

Services I used were Amazon S3 for storing website files and Amazon CloudFront for global content delivery. Key concepts I learnt include content delivery networks (CDNs), caching at edge locations, static website hosting in S3, origin access control (OAC), bucket policies for permissions, and how load times differ between CloudFront and S3 hosting.

Project reflection

This project took me approximately 2 hours to complete. The most challenging part was resolving the access denied and forbidden errors by correctly configuring bucket policies and origin access. It was most rewarding to see the site finally load through both S3 and CloudFront, and to understand why CloudFront provides a faster, more secure solution.

L

Louis Moyo
NextWork Student

nextwork.org

I did this project today to strengthen my hands-on AWS skills and gain practical experience with deploying a website in a three-tier architecture. Yes, this project met my goals by helping me build confidence in configuring S3 and CloudFront, understanding their differences, and preparing me for real-world cloud engineering tasks.

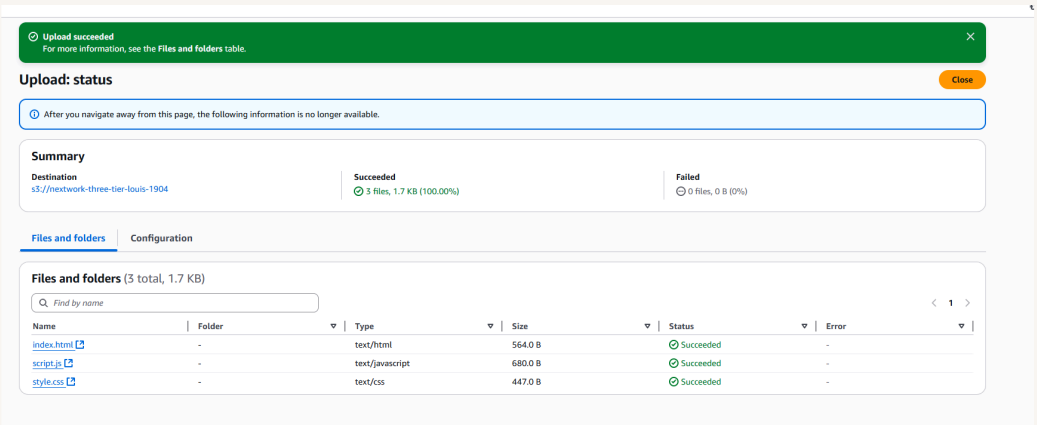
Louis Moyo
NextWork Student

Set Up S3 and Website Files

I started the project by creating an S3 bucket to store the website's files securely and make them accessible for global distribution. I can't use CloudFront for this task because CloudFront is a content delivery network, not a storage service - it relies on S3 or another origin to fetch and deliver the content.

The three files that make up my website are index.html, which is the main structure and content of the webpage, style.css, which defines the layout and visual design of the page, and script.js, which adds interactivity and dynamic functionality to the website.

I validated that my website files work by locating them in my Downloads folder and opening index.html in my browser. The page displayed correctly, confirming that index.html, style.css, and script.js were all downloaded and functioning as intended.


Upload succeeded
For more information, see the Files and folders table.
Upload: status
Close
After you navigate away from this page, the following information is no longer available.
Summary
Destination
s3://nextwork-three-tier-louis-1904
Succeeded
3 files, 1.7 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (3 total, 1.7 KB)
Find by name
1
Name
Folder
Type
Size
Status
Error
index.html
-
text/html
564.0 B
Succeeded
-
script.js
-
text/javascript
680.0 B
Succeeded
-
style.css
-
text/css
447.0 B
Succeeded
-

L
Louis Moyo
NextWork Student

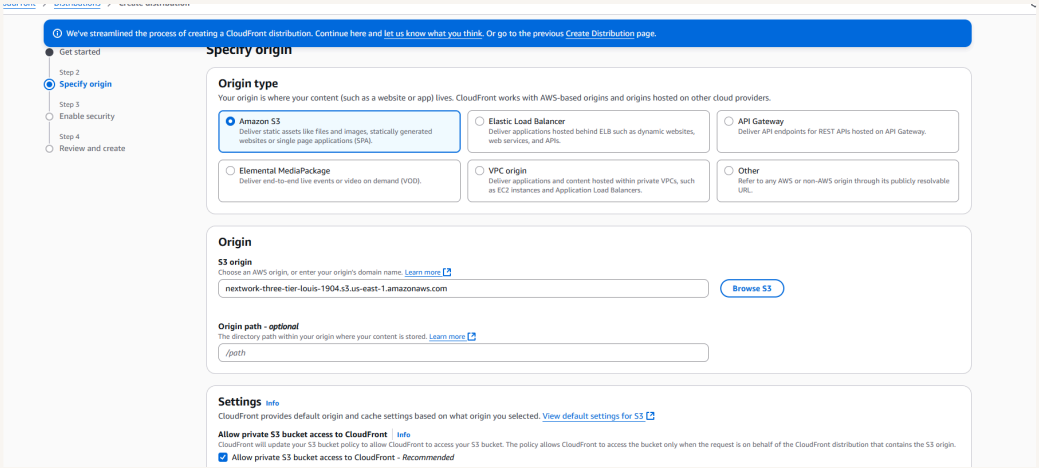
Exploring Amazon CloudFront

Amazon CloudFront is a content delivery network, which means it stores cached copies of website content at edge locations around the world to deliver it faster to users. Businesses and developers use CloudFront because it reduces latency, improves website performance, increases reliability, and helps scale content delivery securely to a global audience.

To use Amazon CloudFront, you set up distributions, which are configurations that tell CloudFront where to fetch your content from and how to deliver it to users. I set up a distribution for my website so CloudFront can serve its files globally. The origin is my S3 bucket, where the website files are stored.

My CloudFront distribution's default root object is index.html. This means when someone visits the root URL of my site, CloudFront automatically serves the index.html file stored in my S3 bucket. I kept the origin access as Public, left the cache behaviour and key settings as default, did not enable WAF, and completed the setup so my content can now be delivered globally through CloudFront.

Louis Moyo
NextWork Student

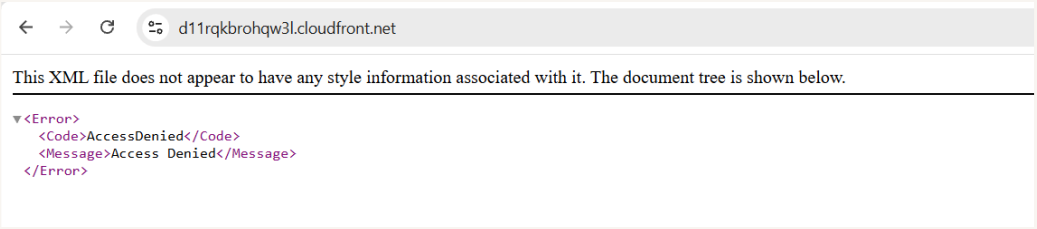


Handling Access Issues

When I tried visiting my distributed website, I ran into an access denied error because my S3 bucket was set to block public access and CloudFront couldn't retrieve the files. Since I kept the origin access as Public without using OAC, the bucket objects also needed to be publicly readable, but the correct bucket policy and permissions weren't applied.

My distribution's origin access settings were set to Public without proper bucket permissions. This caused the access denied error because the S3 bucket still blocked public access by default, preventing CloudFront from retrieving and serving the website files.

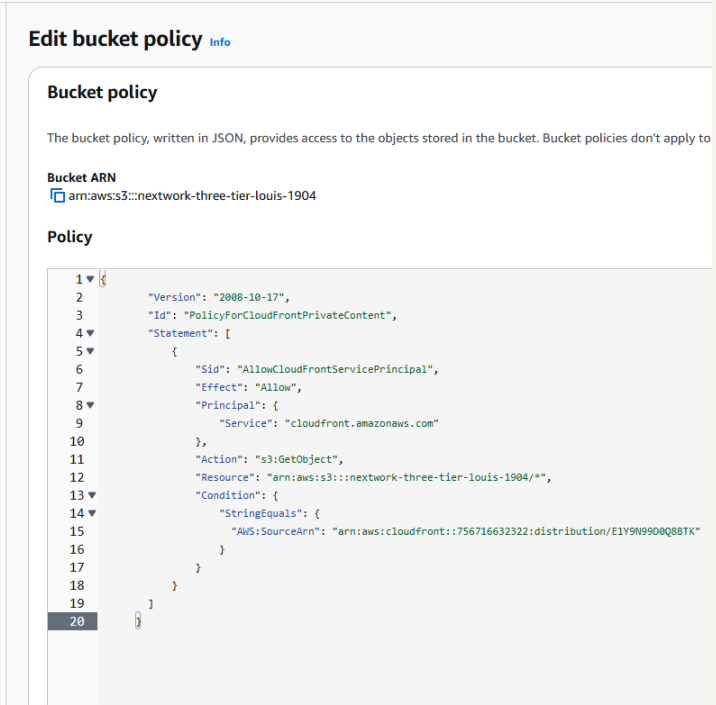
To resolve the error, I set up origin access control (OAC). OAC is a secure method that lets CloudFront authenticate itself when requesting content from an S3 bucket. This ensures CloudFront can access private bucket objects without making the entire bucket public, adding an extra layer of security to the website.


d11rqkbrohqw3l.cloudfront.net
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
</Error>

Updating S3 Permissions

Once I set up my OAC, I still needed to update my bucket policy because the bucket must explicitly allow CloudFront, through the OAC's signed requests, to access its objects. Without this policy update, S3 would continue to block requests, even though OAC was configured.

Creating an OAC automatically gives me a policy I could copy, which grants CloudFront permission to access objects in my S3 bucket. This policy ensures that only CloudFront, using its authenticated identity, can retrieve the files while keeping the bucket private from the public.



Louis Moyo
NextWork Student

S3 vs CloudFront for Hosting

For my project extension, I'm comparing CloudFront and S3 static website hosting. I initially had an error with static website hosting because even though static hosting was enabled, the bucket policy and object permissions weren't updated to allow public access, so S3 blocked requests and returned a 403 Forbidden error.

I tried resolving this by unchecking "Block all public access" in my bucket settings. I still ran into an error because disabling the block doesn't automatically grant permissions - I also need to add a bucket policy that explicitly allows public read access to the objects for S3 static website hosting to work.

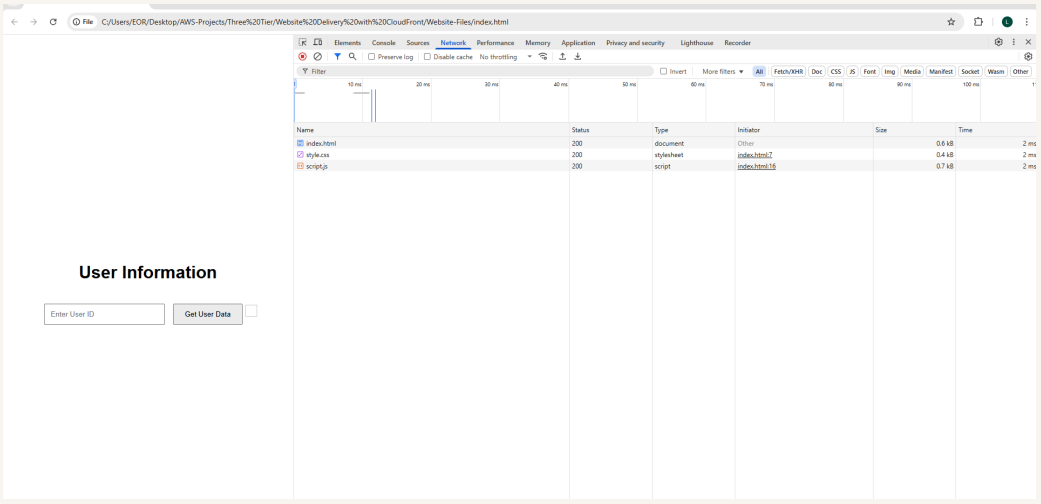
I could finally see my S3 hosted website when I added a bucket policy that explicitly allowed public read access to the objects. This worked because the policy opened access for anyone on the internet to retrieve files from my bucket, which is required for S3 static website hosting.

Compared to the permission settings for my CloudFront distribution, using S3 meant I had to make the bucket publicly accessible, which is less secure. With CloudFront, I could keep the bucket private and use Origin Access Control to let only CloudFront retrieve objects. I preferred the CloudFront setup because it provides better security and global performance.

S3 vs CloudFront Load Times

Load time means how long it takes for a webpage and its resources to fully load in the browser. The load times for the CloudFront site were faster than the S3 site because CloudFront caches content in edge locations closer to users worldwide, while S3 static hosting only serves files from the region where the bucket is hosted.

A business would prefer CloudFront when it needs fast, reliable performance for a global audience, with added security and scalability. S3 static website hosting might be sufficient when the audience is local to one region, performance demands are low, or the site is small and simple.


User Information
Enter User ID
Get User Data

The place to learn & showcase your skills

Check out nextwork.org for more projects