nextwork.org

Website Delivery with CloudFront

AN antcruz567@gmail.com

AN
antcruz567@gmail.com
NextWork Student

Introducing Today's Project!

In this project, I will create a storage space in S3 for my website's files. Set up CloudFront to distribute my website globally. Manage permissions for both S3 and CloudFront. Compare different methods for hosting your website and analyze their performance.

Tools and concepts

Services I used were OAC, S3 Bucket policy, and CloudFront. Key concepts I learnt include content delivery network (CDN), Origin access control, configuration for my S3 bucket, and cloudfront integration.

Project reflection

This project took me approximately 2 hours. The most challenging part was troubleshooting why the website was not accessible after making the necessary changes to the bucket policy. It was most rewarding to finally access the website through the cloudfront distribution

I did this project to showcase how a CloudFront distribution can be integrated with an S3 bucket.

AN
antcruz567@gmail.com
NextWork Student

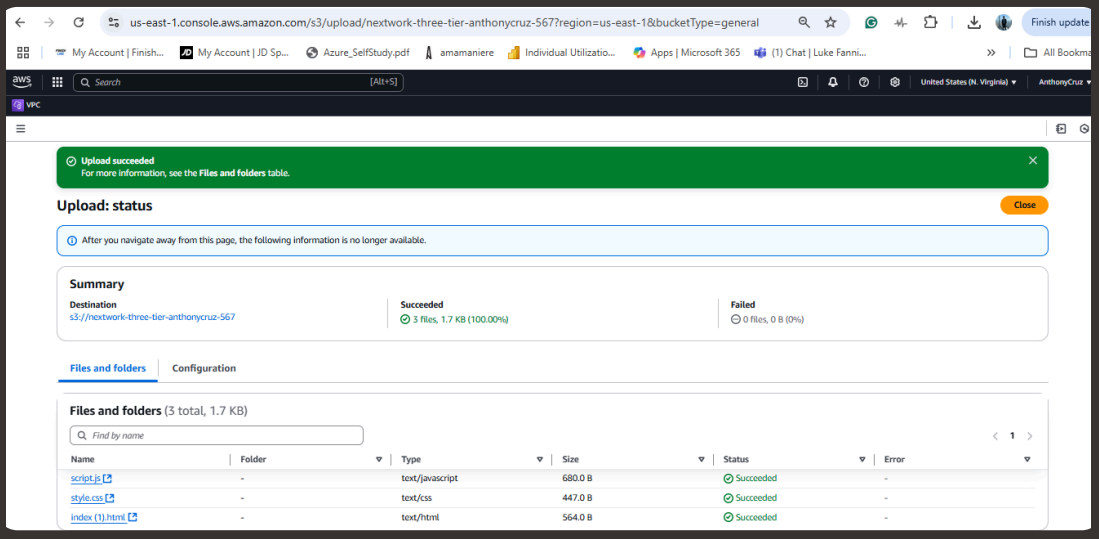
Set Up S3 and Website Files

I started the project by creating an S3 bucket to hold the files that make up my website. I can't use CloudFront for this task because CloudFront is not a storage solution. CloudFront is a content delivery network that simply hosts content that is stored somewhere else, like Amazon S3.

The three files that make up my website are index.html, which is the main file for a website. It's where you organise the text, pictures, and everything that makes up your webpage. Style.css, which is where you write down the visual appearance of your website's HTML elements, and script.js, which is a JavaScript file that adds interaction to your website. It's where you would write the instructions for making things on your website move or change when you click a button or submit a form.

I validated that my website files work by opening the index.html in my browser.

AN
antcruz567@gmail.com
NextWork Student



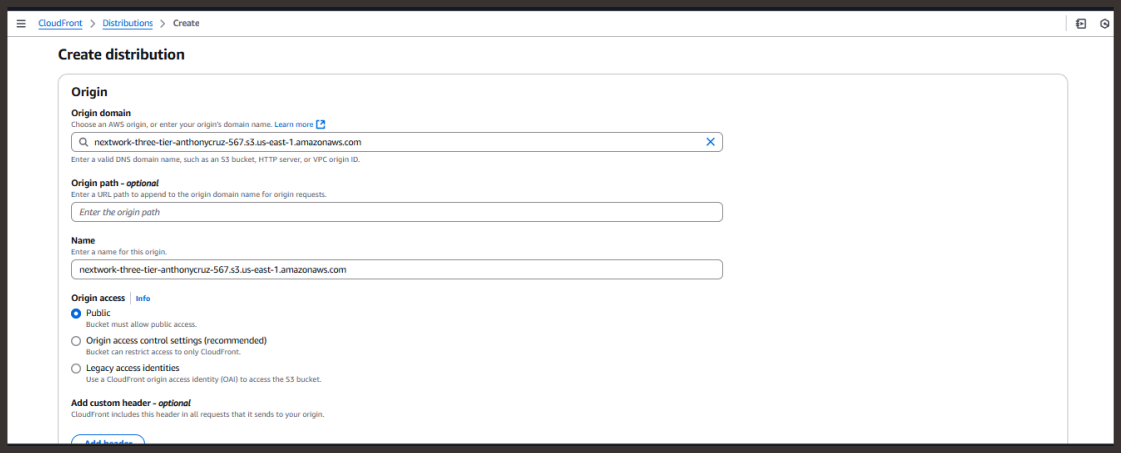
AN
antcruz567@gmail.com
NextWork Student
NextWork.org

Exploring Amazon CloudFront

Amazon CloudFront is a content delivery network, which means it speeds up the distribution of your static and dynamic web content, such as .html, .css, .js, and image files. Businesses and developers use CloudFront because requests are routed to the edge location that provides the lowest latency (time delay), so content is delivered with the best possible performance.

To use Amazon CloudFront, you set up distributions, which are a set of instructions that tell CloudFront how to deliver your content. I set up a distribution for my S3 bucket. The origin is nextwork-three-tier-anthonycruz-567.s3.us-east-1.amazonaws.com

My CloudFront distribution's default root object is index.html. This means this is the file that CloudFront should serve when someone visits the root URL of my website.


CloudFront > Distributions > Create
Create distribution
Origin
Origin domain
Choose an AWS origin, or enter your origin's domain name. Learn more
nextwork-three-tier-anthonycruz-567.s3.us-east-1.amazonaws.com
Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.
Origin path - optional
Enter a URL path to append to the origin domain name for origin requests.
Enter the origin path
Name
Enter a name for this origin.
nextwork-three-tier-anthonycruz-567.s3.us-east-1.amazonaws.com
Origin access | Info
Public
Bucket must allow public access.
Origin access control settings (recommended)
Bucket can restrict access to only CloudFront.
Legacy access identities
Use a CloudFront origin access identity (OAI) to access the S3 bucket.
Add custom header - optional
CloudFront includes this header in all requests that it sends to your origin.

AN
antcruz567@gmail.com
NextWork Student
NextWork.org

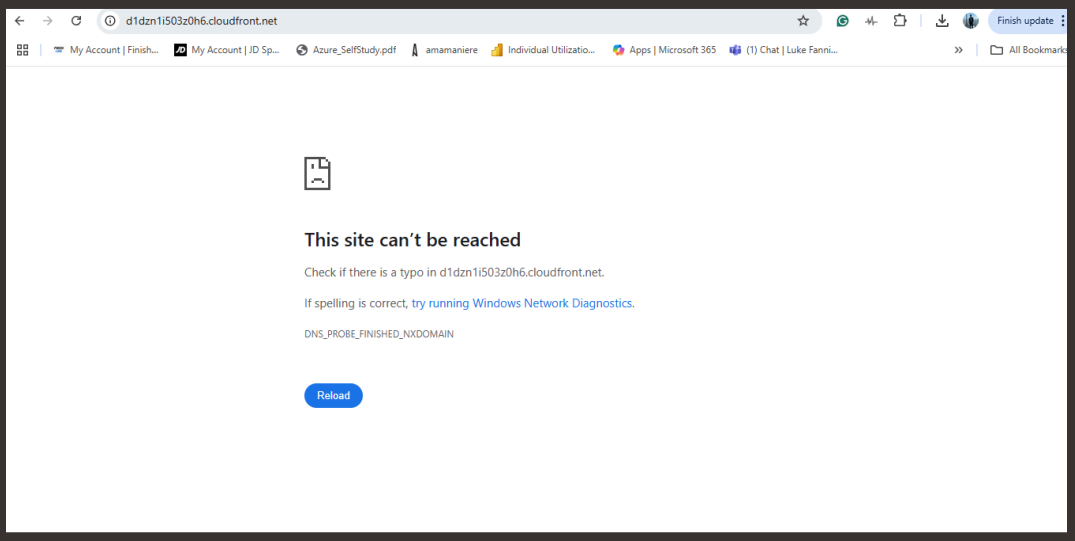
Handling Access Issues

When I tried visiting my distributed website, I ran into an access denied error because cloudfront does not have access to retrieve from my S3 bucket, yet.

My distribution's origin access settings were public, it means anyone can access the content directly from the origin, such as my S3 bucket, not just via CloudFront. This can become a security risk if sensitive data is accessible. This caused the access denied error because setting the origin access to public doesn't automatically update the permissions of the objects in your S3 bucket - for security, the objects are private by default. I will still need to go to my S3 bucket and set all my objects to public for everyone to access them.

To resolve the error, I set up origin access control (OAC). OAC lets me keep my S3 bucket and objects not publicly accessible, while still making sure they can be accessed through CloudFront.

AN
antcruz567@gmail.com
NextWork Student

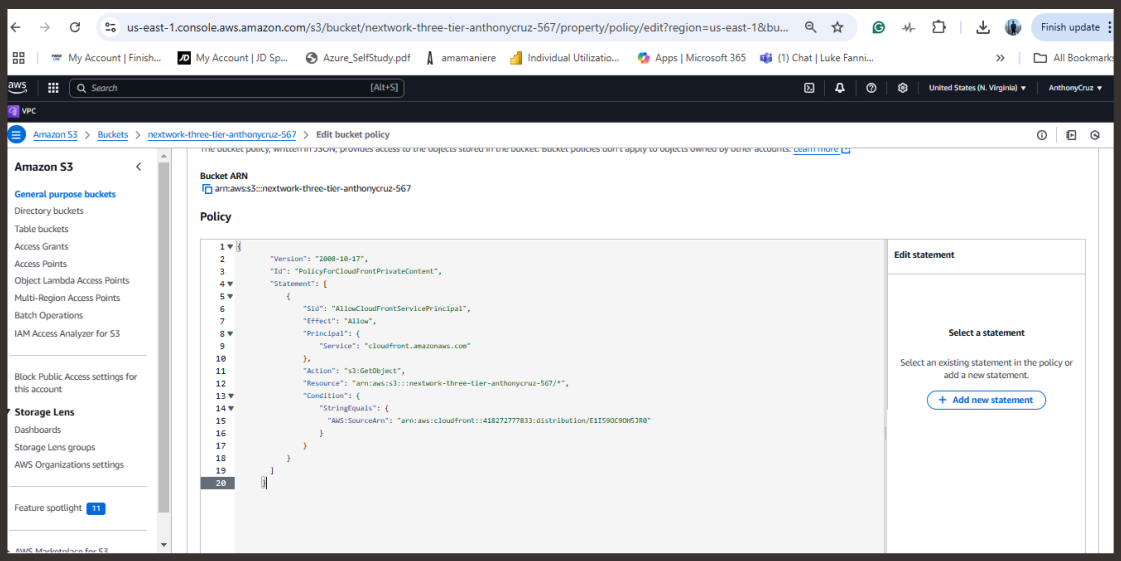


AN
antcruz567@gmail.com
NextWork Student

Updating S3 Permissions

Once I set up my OAC, I still needed to update my bucket policy because the OAC's role is that it makes sure only CloudFront can access the files stored in the S3 bucket. For this restricted access to be effective, the S3 bucket's policy still needs to explicitly grant the OAC permission to the bucket's contents.

Creating an OAC automatically gives me a policy I could copy, which grants cloudfront access to the S3 bucket and the necessary permissions.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects