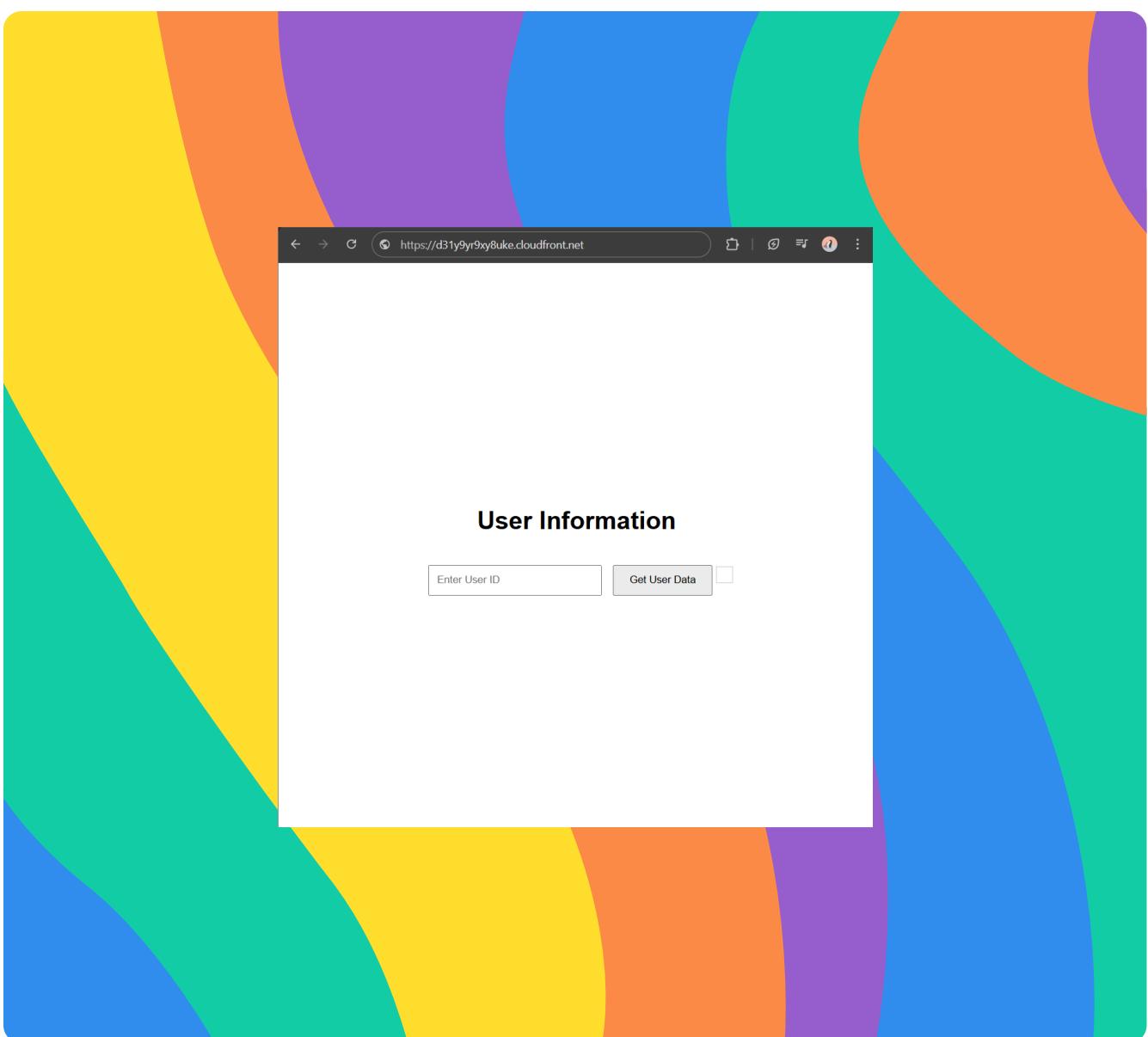


[nextwork.org](https://nextwork.org)

# Website Delivery with CloudFront



jyothesh karnam





# Introducing Today's Project!

In this project, I will demonstrate how to use CloudFront to deliver a website globally, I'm doing this project to learn about CDNs (content delivery networks), and to set up the presentation tier of a three-tier architecture

## Tools and concepts

Services I used include Amazon S3 and Amazon CloudFront. Key concepts I learnt include content delivery network (CDN), distributions, origin access control (OAC), performance load times, S3 static hosting vs CloudFront

## Project reflection

This project took me approximately 2 hours including all of the demos. The most challenging part was understanding origin access control (OAC). It was most rewarding to compare performance load times and see CloudFront outperforming S3

I did this project today to learn about CloudFront and content delivery. This project met my goals and I've successfully built the presentation tier of the AWS three-tier architecture!

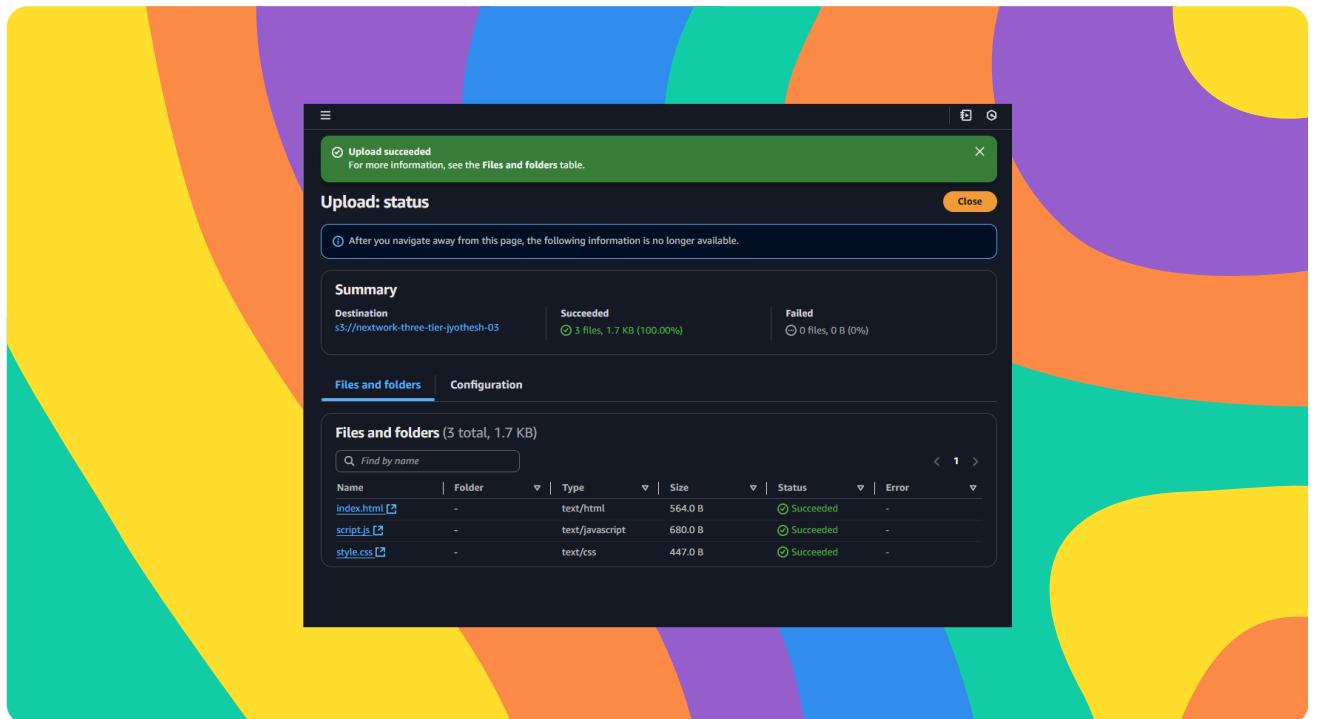


# Set Up S3 and Website Files

I started the project by creating an S3 bucket to store my websites files. I can't use CloudFront for this task because CloudFront is not a storage service. It is a content delivery service for objects that are already stored somewhere.

The three files that make up my website are index.html, which outlines the text and images which is the structure of the website. style.css, which adds styling to my HTML elements and script.js, which adds interactive elements like form submissions.

I validated that my website files work by opening index.html locally on my browser. I saw a simple web page without errors, so the three files can be uploaded into S3





jyothesh karnam  
NextWork Student

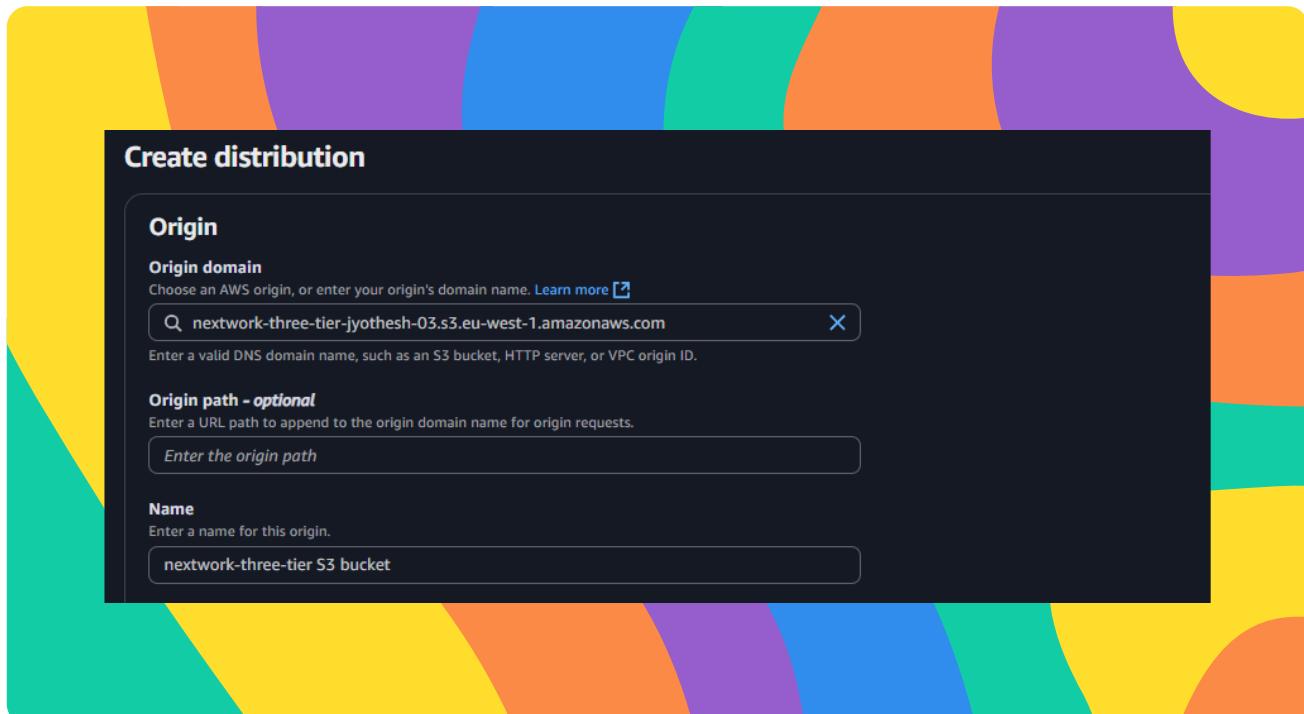
[NextWork.org](https://NextWork.org)

# Exploring Amazon CloudFront

Amazon CloudFront is a content delivery network, which means it caches web contents which are website files to edge locations/server around the world. Businesses and developers use CloudFront because it speeds up their website performance.

To use Amazon CloudFront, you set up distributions, which are instructions that tell CloudFront how to deliver your content. I set up a distribution for my website and the origin is my S3 bucket.

My CloudFront distribution's default root object is index.html. This means when the users visit my website's root URL, they will see the content described in index.html.



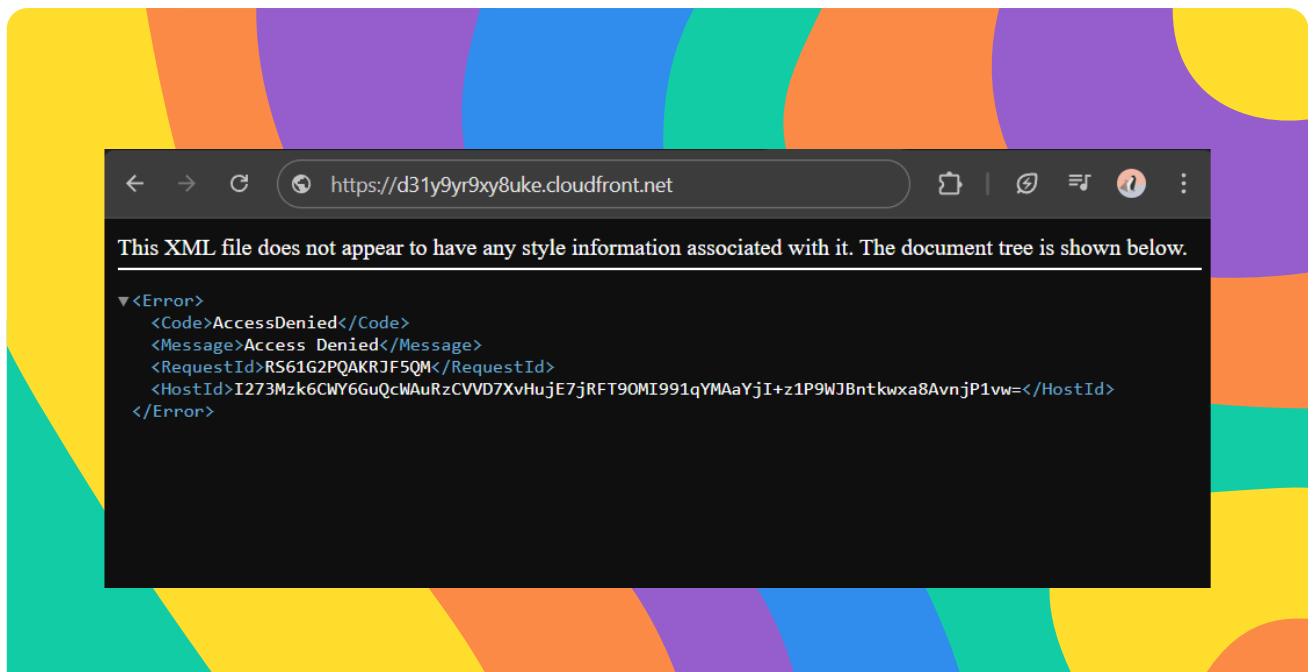


# Handling Access Issues

When I tried visiting my distributed website, I ran into an access denied error because S3 bucket is private by default. So i need to provide explicit permissions for CloudFront to access the S3 bucket.

My distribution's origin access settings were set to Public which means the bucket needs to be public too. This caused the access denied error because CloudFront did not have permissions to access the S3 bucket.

To resolve the error, I set up origin access control (OAC). OAC is a special user that is created to connect S3 with CloudFront which means giving CloudFront access while still keeping out public access to the buckets objects. This improves security for our content.





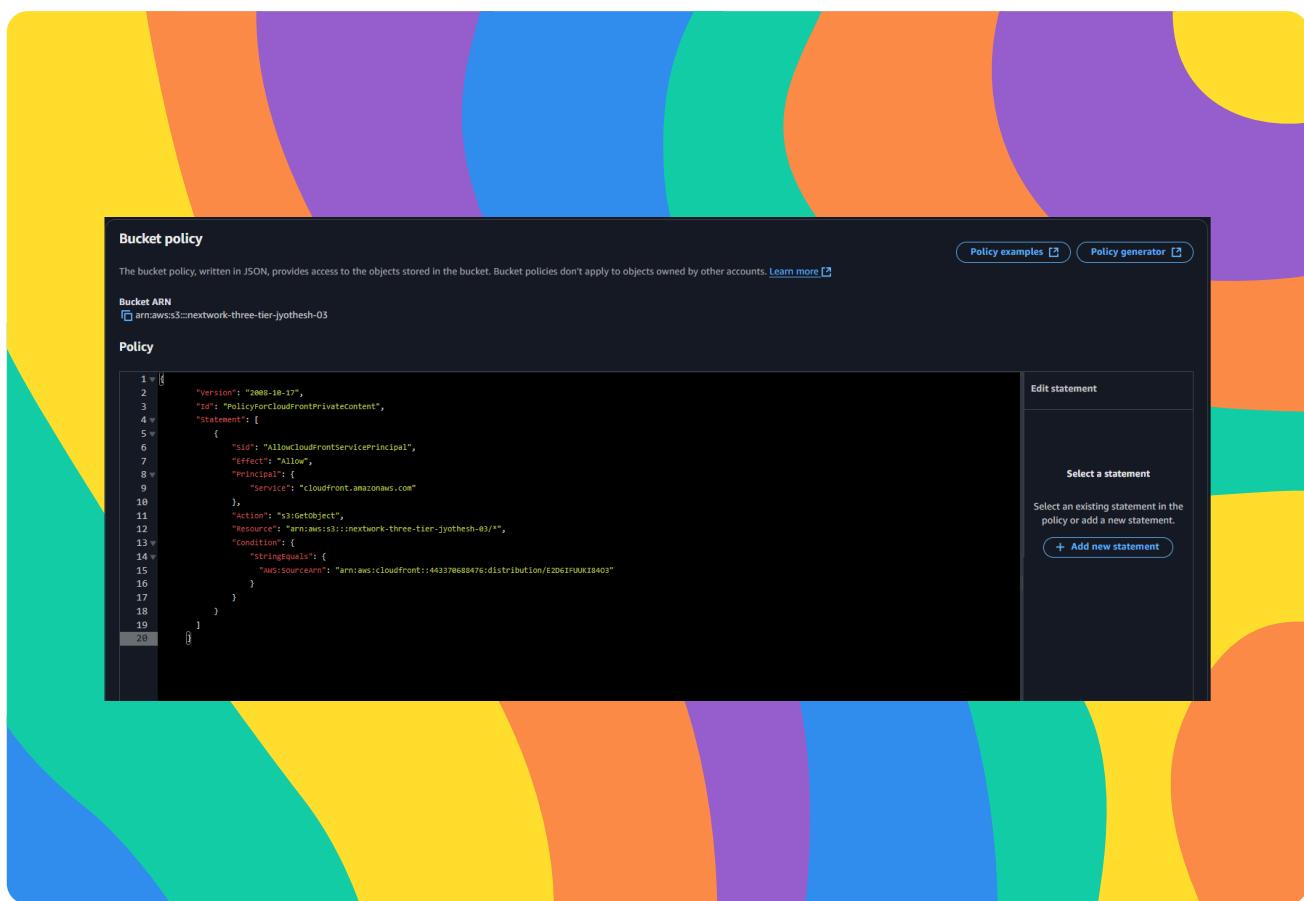
jyothesh karnam  
NextWork Student

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

# Updating S3 Permissions

Once I set up my OAC, I still needed to update my bucket policy because its objects are still private to everyone including CloudFront. Just creating the OAC does not mean the CloudFront has automatic access to S3.

Creating an OAC automatically gives me a policy I could copy, which grants CloudFront distribution access to all of the objects inside our S3 bucket.





# S3 vs CloudFront for Hosting

'For my project extension, I'm comparing S3 with CloudFront. I initially had an error with static website hosting because we enabled S3 static website hosting without making our objects publicly available.

I tried resolving this by turning off block public access in our S3 bucket. I still ran into an error (403 Forbidden) when accessing the static website URL, because we still need to update the bucket policy which still doesn't allow public access to the objects yet.

I could finally see my S3 hosted static website when I updated the bucket policy to allow the public to access the bucket's objects. This worked because our bucket policy previously only allowed CloudFront access. So changing S3 permissions is required.

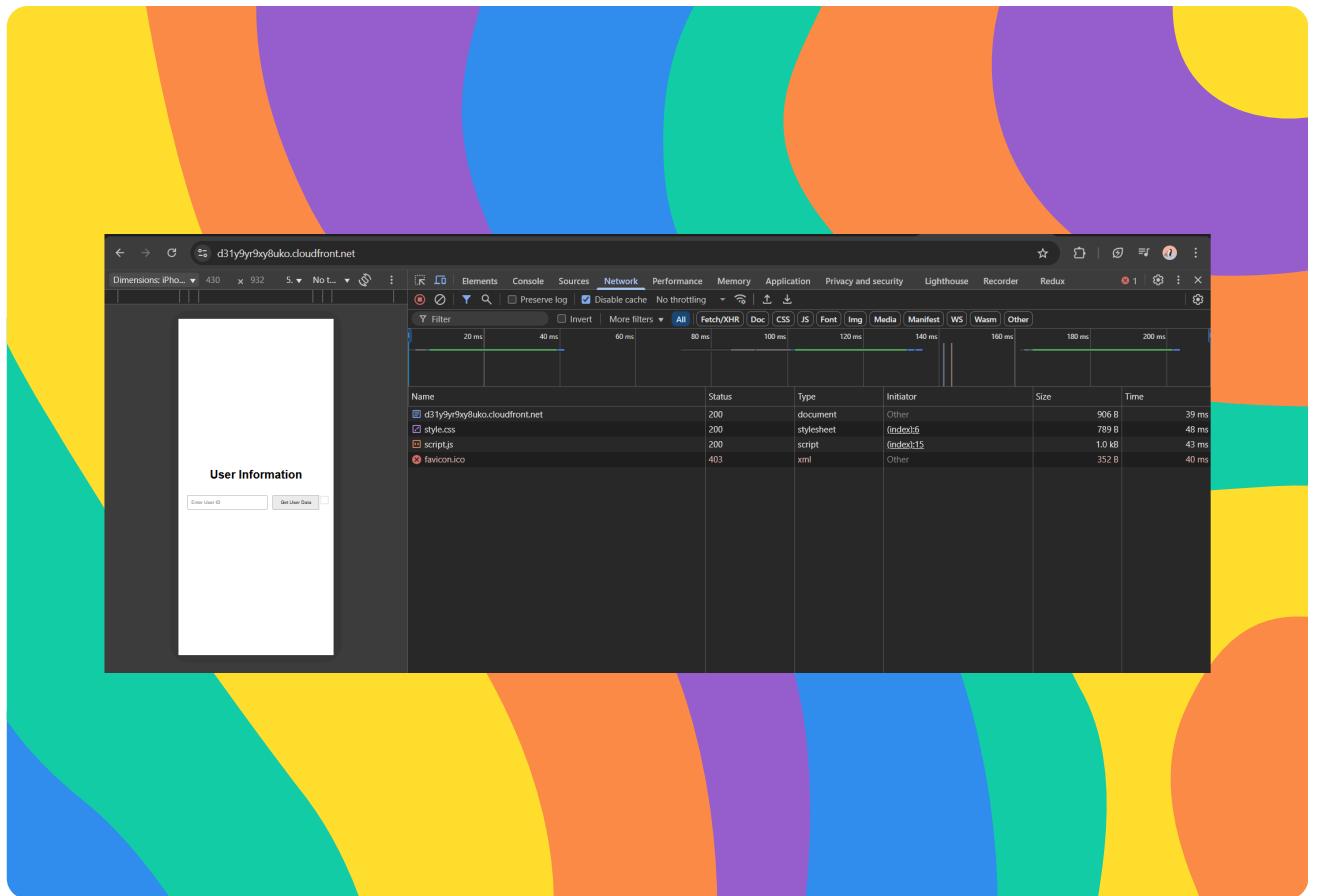
Compared to the permission settings for my CloudFront distribution, using S3 meant I had to enable public access to my bucket. I preferred using CloudFront because it meant bucket access is restricted to CloudFront only which is great for security.



# S3 vs CloudFront Load Times

Load time means how long it takes for our browser to receive requested content and display it for the user. The load times for the CloudFront site were faster than the S3 site because content is cached from edge locations.

A business would prefer CloudFront when performance/load times are important to them. S3 static website hosting might be sufficient when you are hosting a simple website without performance/load time requirements.





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

