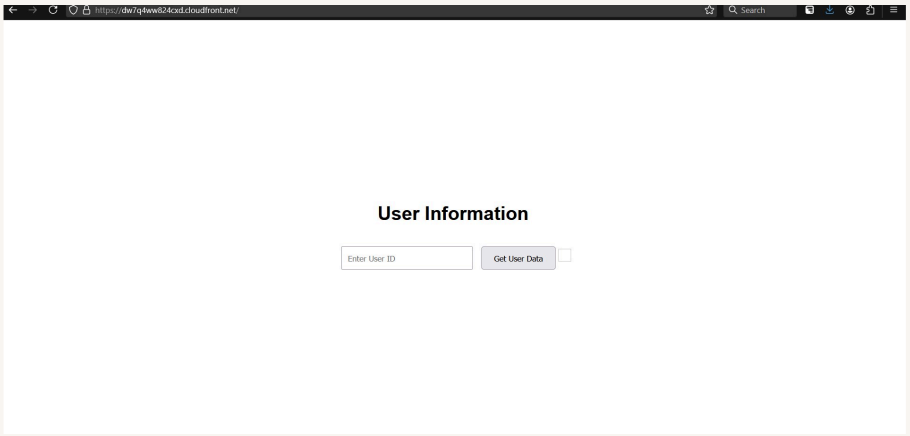
nextwork.org

Website Delivery with CloudFront

Chrispinus Jacob



Chrispinus Jacob
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use Amazon CloudFront to deliver a website globally. I'm doing this project to learn about CDNs and how to set up the presentation tier of a three-tier architecturefor scalable and efficient web applications.

Tools and concepts

Services I used were Amazon S3 and Amazon CloudFront. Key concepts I learnt include content delivery network (CDN), origin access control (OAC), bucket policies, static website hosting, and the presentation tier of a three-tier architecture.

Project reflection

This project took me approximately a 2 hours to complete. The most challenging part was troubleshooting the access denied errors and correctly configuring permissions between S3 and CloudFront. It was most rewarding to see the website load successfully through both S3 and CloudFront, and to understand the benefits of using a CDN in a real-world setup.

placeholder

I did this project today to learn how content delivery networks like Amazon CloudFront work and how they fit into a three-tier architecture by handling the presentation layer.

Yes, this project met my goals because I was able to set up CloudFront, connect it to an S3 origin, troubleshoot permission issues, and compare it with S3 static website hosting to understand the performance and security benefits of using a CDN.

Chrispinus Jacob
NextWork Student

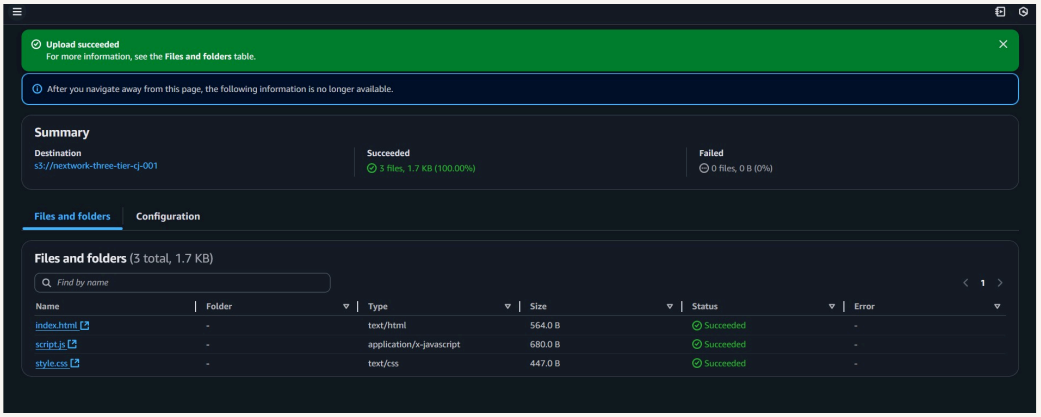
Set Up S3 and Website Files

In this step, I started the project by creating an S3 bucket to store website files. I can't use CloudFront for this task because CloudFront is not a storage service; it is a content delivery network used to distribute files that are stored somewhere else.

The three files that make up my website are index.html, which defines the structure and content of the webpage; style.css, which controls the visual appearance and layout; and script.js, which adds interactivity and dynamic behavior to the site.

I validated that my website files work by opening index.html in my local browser to check that the layout loads correctly, the style.css file is applied properly, and the script.js functions as intended for interactivity.

Chrispinus Jacob
NextWork Student



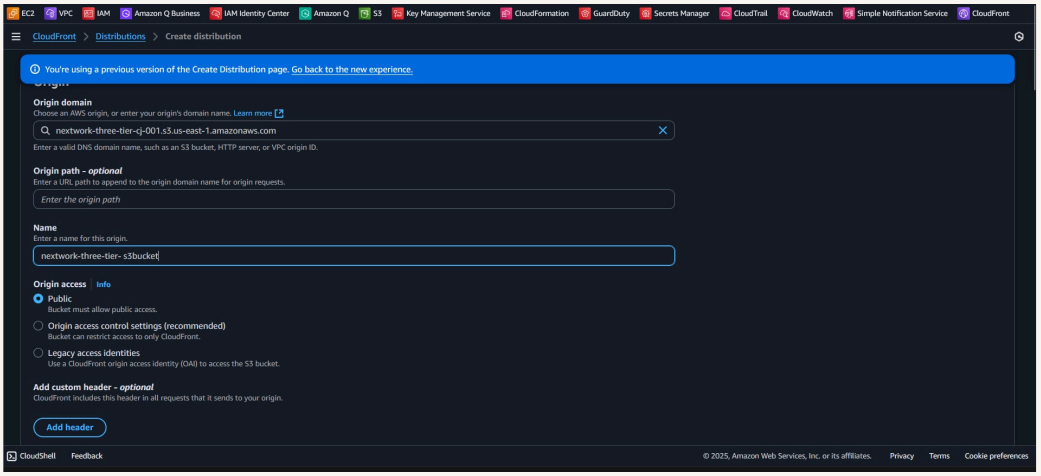
Chrispinus Jacob
NextWork Student

Exploring Amazon CloudFront

Amazon CloudFront is a content delivery network, which means it delivers website content to users from the nearest edge location to reduce latency and improve performance. Businesses and developers use CloudFront because it speeds up content delivery, enhances user experience, supports security features like HTTPS, and scales automatically to handle traffic from anywhere in the world.

To use Amazon CloudFront, you set up distributions, which are configurations that tell CloudFront where to fetch content from and how to deliver it to users. I set up a distribution for my website files stored in an S3 bucket. The origin is the S3 bucket that contains my index.html, style.css, and script.js files.

My CloudFront distribution's default root object is index.html. This means that when users visit the root URL of my website, CloudFront automatically serves the index.html file as the main page without requiring the full file path.


EC2 VPC IAM Amazon Q Business IAM Identity Center Amazon Q S3 Key Management Service CloudFormation GuardDuty Secrets Manager CloudTrail CloudWatch Simple Notification Service CloudFront
CloudFront > Distributions > Create distribution
You're using a previous version of the Create Distribution page. Go back to the new experience.
Origin domain
Choose an AWS origin, or enter your origin's domain name. Learn more
nextwork-three-tier-cj-001.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- s3bucket
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.
Add header
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

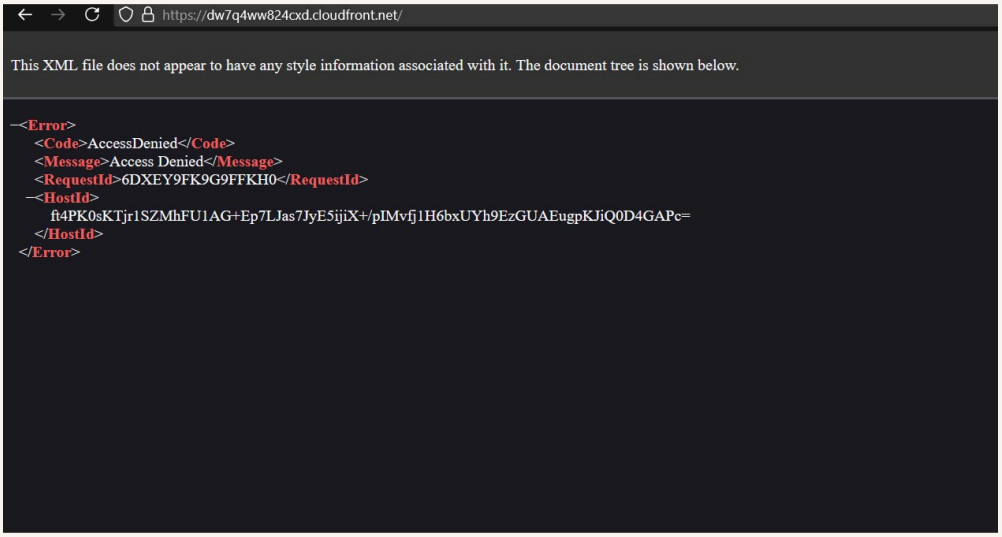
Chrispinus Jacob
NextWork Student

Handling Access Issues

When I tried visiting my distributed website, I ran into an access denied error because my S3 bucket did not have the correct permissions to allow CloudFront to access the files. The bucket policy was either missing or not properly configured to grant read access to the public or to CloudFront specifically.

My distribution's origin access settings were set to public. This caused the access denied error because even though the S3 bucket allowed public access, CloudFront requires specific permissions through origin access control or a proper bucket policy to retrieve files securely. Without this, CloudFront requests are still blocked.

To resolve the error, I set up origin access control (OAC). OAC is a security feature that allows CloudFront to access content in an S3 bucket securely without making the bucket publicly accessible. It replaces older access methods like origin access identity (OAI) and helps ensure that only CloudFront can retrieve the files from the bucket.

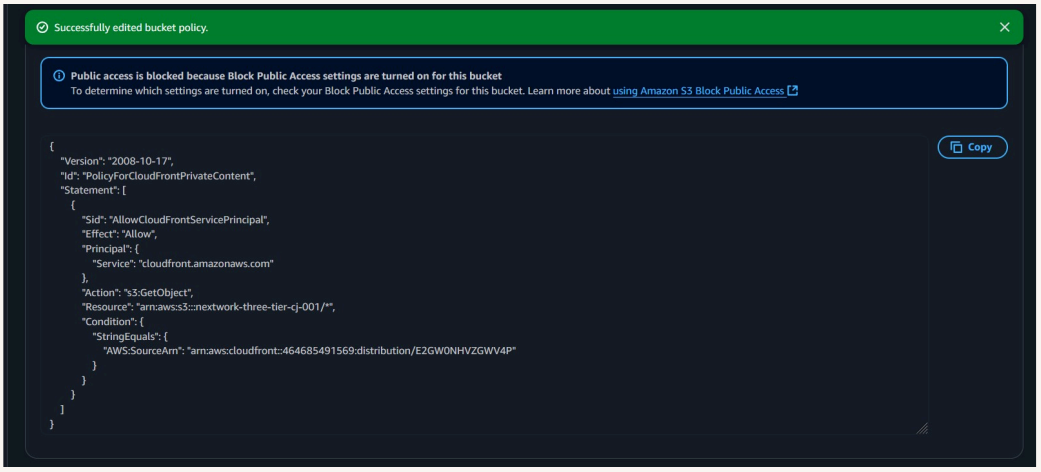

https://dw7q4ww824cxd.cloudfront.net/
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>6DXEY9FK9G9FFKH0</RequestId>
<HostId>
ft4PK0sKTjr1SZMhFU1AG+Ep7LJas7JyE5ijiX+/pIMvfj1H6bxUYh9EzGUAEugpKJiQ0D4GAPc=
</HostId>
</Error>

Chrispinus Jacob
NextWork Student

Updating S3 Permissions

Once I set up my OAC, I still needed to update my bucket policy because the S3 bucket must explicitly allow access from the CloudFront service using the OAC's signed requests. Without updating the policy, S3 would continue to block access even though OAC was in place.

Creating an OAC automatically gives me a policy I could copy, which grants CloudFront permission to access the S3 bucket using the OAC's signed requests. This policy ensures that only CloudFront can read the objects, while keeping the bucket private from the public.



Chrispinus Jacob
NextWork Student

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 the object publicly available, so the browser couldn't access the files directly.

I tried resolving this by making the bucket public and enabling static website hosting. I still ran into an error because the individual objects remained private due to S3's default settings, and the bucket policy did not yet allow public read access to those objects.

I could finally see my S3 hosted website when I updated the bucket policy to allow public read access and disabled the block public access settings. This worked because it gave browsers permission to load the objects directly from S3 through the static website hosting endpoint.

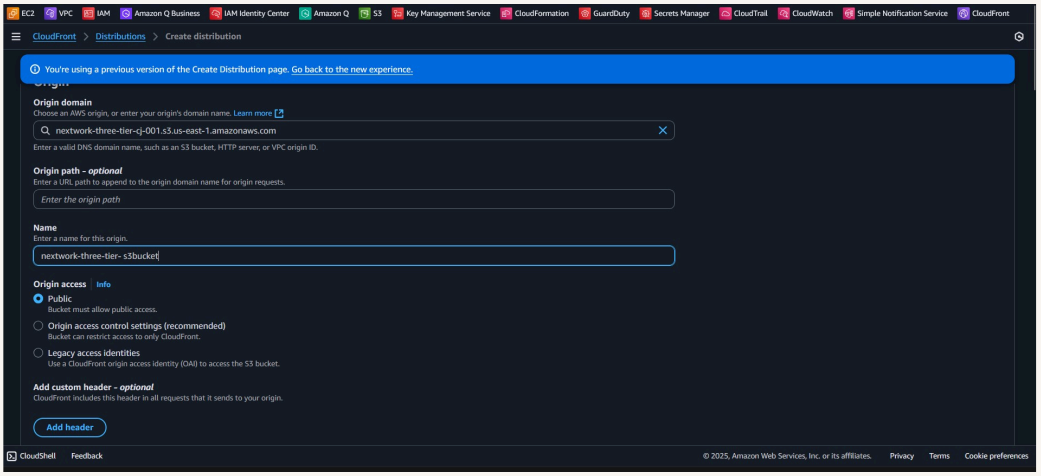
Compared to the permission settings for my CloudFront distribution, using S3 meant making the bucket and its objects publicly accessible, which is less secure. I preferred the CloudFront setup because it allows access through signed requests using origin access control, keeping the S3 bucket private while still delivering content efficiently.

Chrispinus Jacob
NextWork Student

S3 vs CloudFront Load Times

Load time means the amount of time it takes for a website to fully load and become usable for the visitor. The load times for the CloudFront site were faster than the S3 site because CloudFront uses edge locations around the world to deliver content closer to the user, reducing latency and improving performance.

A business would prefer CloudFront when it needs fast, secure, and reliable global content delivery with low latency and support for large-scale traffic. S3 static website hosting might be sufficient when the website is small, has low traffic, and is mainly accessed by users in a specific region


EC2 VPC IAM Amazon Q Business IAM Identity Center Amazon Q S3 Key Management Service CloudFormation GuardDuty Secrets Manager CloudTrail CloudWatch Simple Notification Service CloudFront
CloudFront > Distributions > Create distribution
You're using a previous version of the Create Distribution page. Go back to the new experience.
Origin domain
Choose an AWS origin, or enter your origin's domain name. Learn more
nextwork-three-tier-cj-001.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- s3bucket
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.
Add header
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The place to learn & showcase your skills

Check out nextwork.org for more projects