

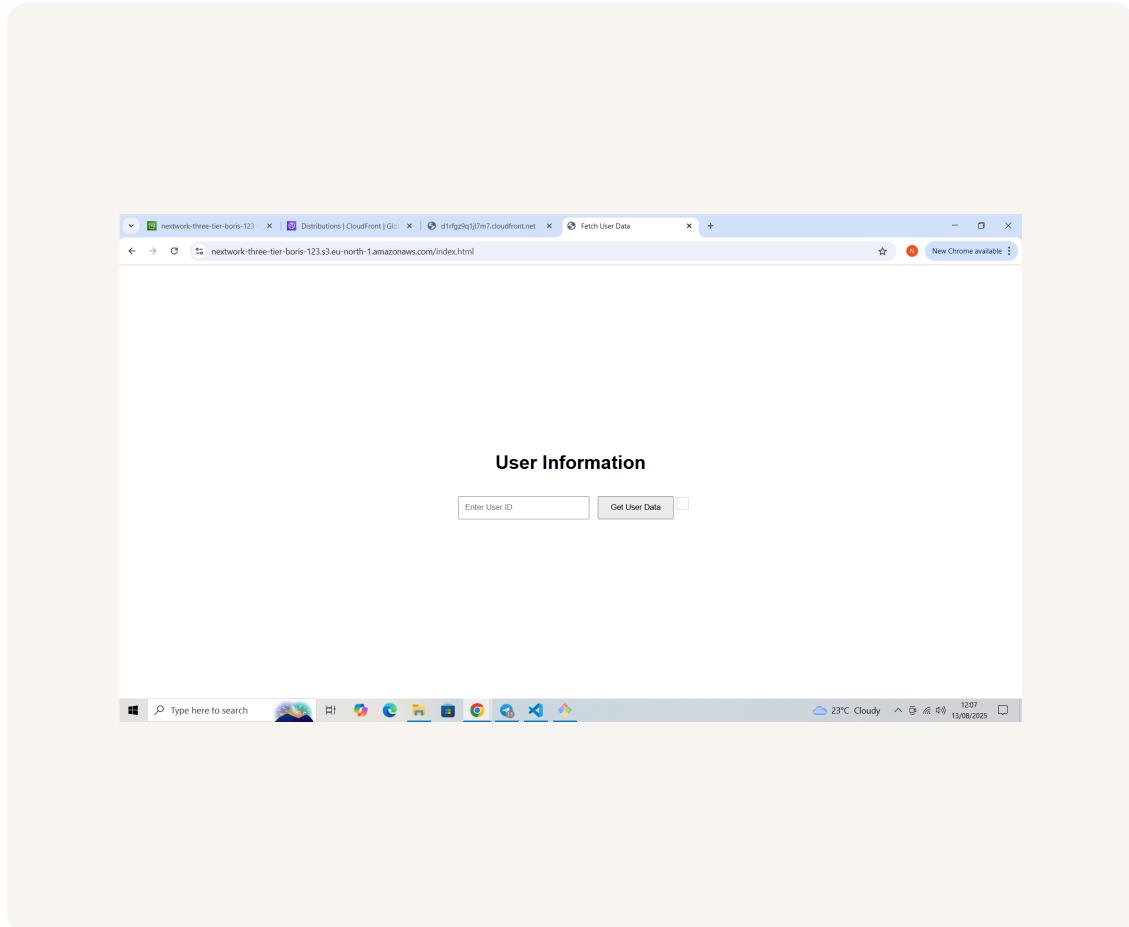


[nextwork.org](http://nextwork.org)

# Website Delivery with CloudFront



Nchindo Boris





# 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 my website and analyze their performance.

## Tools and concepts

Services I used were; S3 and CloudFront. Key concepts I learnt include content delivery network (CDN), Securing an S3 bucket using Origin Access Identities

## Project reflection

This project took me approximately 1 hour 20 minutes to complete. The most challenging part was troubleshooting for denied access and errors. It was most rewarding to finally see the website successfully distributed

I did this project today as it is one of the essential projects and goals in getting a cloud career



# Set Up S3 and Website Files

I started the project by creating an S3 bucket so it can hold the files that make up our 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. It controls everything from font sizes and colors to layout designs, helping you keep a consistent style across your website and - script.js, which refers to 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 file with my browser. It displayed a simple webpage



The screenshot shows the AWS S3 console interface. The bucket name is 'nextwork-three-tier-boris-123'. The 'Objects' tab is selected, displaying three files:

Name	Type	Last modified	Size	Storage class
index.html	html	August 13, 2025, 10:06:56 (UTC+01:00)	564.0 B	Standard
script.js	js	August 13, 2025, 10:06:55 (UTC+01:00)	680.0 B	Standard
style.css	css	August 13, 2025, 10:06:55 (UTC+01:00)	447.0 B	Standard



# 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 content is delivered fast with the best possible performance.

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

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



The screenshot shows the 'Specify origin' step of the CloudFront distribution creation wizard. The navigation bar indicates 'Step 2 Specify origin' is selected. The 'Origin type' section is active, showing five options:

- Amazon S3** (selected): Deliver static assets like files and images, statically generated websites or single page applications (SPA).
- Elastic Load Balancer**: Deliver applications hosted behind ELB such as dynamic websites, web services, and APIs.
- API Gateway**: Deliver API endpoints for REST APIs hosted on API Gateway.
- Elemental MediaPackage**: Deliver end-to-end live events or video on demand (VOD).
- VPC origin**: Deliver applications and content hosted within private VPCs, such as EC2 instances and Application Load Balancers.
- Other**: Refer to any AWS or non-AWS origin through its publicly resolvable URL.

The 'Origin' section below shows the input field 'S3 origin' containing 'nextwork-three-tier-boris-123.s3.eu-north-1.amazonaws.com' and a 'Browse S3' button. The 'Origin path - optional' field is empty. The bottom of the window shows the standard AWS navigation bar with links for CloudShell, Feedback, and various services like Lambda, CloudWatch, and CloudFront.

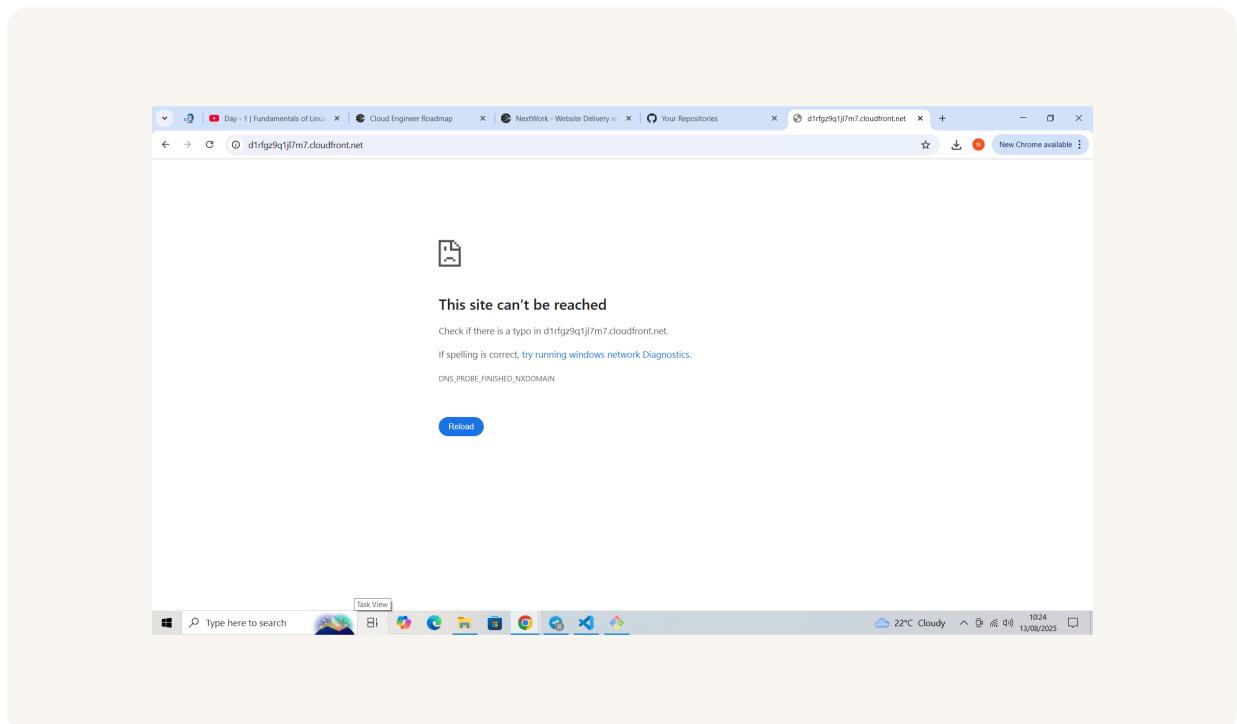


# Handling Access Issues

When I tried visiting my distributed website, I ran into an access denied error because I haven't given CloudFront permission to access the S3 bucket yet. By default, S3 buckets are private. CloudFront needs explicit permission to access the files in your bucket.

My distribution's origin access settings were public. 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. You'll still need to go to your S3 bucket and set all your objects to public for everyone to access them. So, even if origin access is public, CloudFront can't access your website's content until your objects' access settings are also set to public.

To resolve the error, I set up origin access control (OAC). OAC is a special user for CloudFront that prevents this. An OAC lets you keep your S3 bucket and objects not publicly accessible, while still making sure they can be accessed through CloudFront.





# Updating S3 Permissions

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

Creating an OAC automatically gives me a policy I could copy, which grants access into the bucket.

The screenshot shows the AWS S3 Bucket Policy editor for a bucket named "nchindo-boris-123". The policy document is displayed in a code editor:

```
1 Version: "2008-10-17",
2 Id: "PolicyForCloudFrontPrivateContent",
3 Statement: [
4     {
5         Sid: "AllowCloudFrontServicePrincipal",
6         Effect: "Allow",
7         Principal: {
8             Service: "cloudfront.amazonaws.com"
9         },
10        Action: "s3:GetObject",
11        Resource: "arn:aws:s3:::nchindo-boris-123/*",
12        Condition: {
13            StringEquals: {
14                "AWS:SourceArn": "arn:aws:cloudfront::55074477562:distribution/E234P257KU5WV"
15            }
16        }
17    }
18 ]
19 ]
20 ]
```

The right side of the interface shows a "Select a statement" dropdown and a "+ Add new statement" button. The top navigation bar shows the URL "eu-north-1.console.aws.amazon.com/s3/bucket/nchindo-boris-123/property/policy/edit?region=eu-north-1&bucketType=general".



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

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

