



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

# Website Delivery with CloudFront

ME

Melvin J Bonner

## User Information

Enter User ID	Get User Data	<input type="checkbox"/>
---------------	---------------	--------------------------

# Introducing Today's Project!

In this project I will be setting a storage space up with S3 and deploying my website with CloudFront.

## Tools and concepts

Services I used were S3 and CloudFront. Key concepts I learned include content delivery network (CDN) and Origin Access Control (OAC).

## Project reflection

The project took me approximately 2 hours. The most challenging part was setting up CloudFront distribution. It was most rewarding to complete the secret missions.

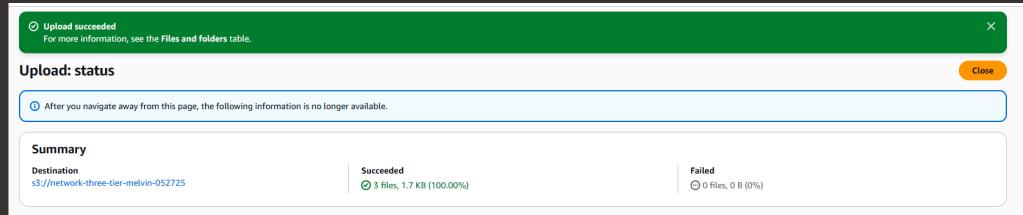
I wanted to be challenged by a project that allowed me to work with different AWS services. This project met my goals. I especially enjoyed comparing S3 and CloudFront performance times.

# Set Up S3 and Website Files

I started this project by creating an S3 bucket to store the files that make up the website, because CloudFront is not a storage solution.

The three files that make up my website are index.html, which is the main file for my website, style.css, which is where the visual appearance of the website comes from, and script.jc, which add interaction to the website.

I validate that my website files work by opening the html file in my browser.



# Exploring Amazon CloudFront

Amazon CloudFront is a content delivery network, which means it speeds up the distribution of our static and dynamic web content. Businesses and developers use CloudFront because CloudFront uses edge locations that provide low latency, 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 the origin. The origin specifies where the website files are stored and how they should be cached.

My CloudFront distribution's default root object is index.html. This means when a user visits the root URL the web browser is looking for index.html.

ME

# Melvin J Bonner

## NextWork Student

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

**Origin**

**Origin domain**  
Choose an AWS origin, or enter your origin's domain name. [Learn more](#) 

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.

**Name**  
Enter a name for this origin.

**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**

# Handling Access Issues

When I tried visiting my distributed website, I ran into an access denied error because, CloudFront wasn't given permission to access the S3 bucket.

My distribution access settings were set to public. This caused the access denied error because the bucket permissions on S3 were set to block public access.

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

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>Z5K869AZ2DX7XTC</RequestId>
  <HostId>Ulr6ls7D41US1d1Cb7I9UOXRNhIONBfv8S0tc+i9QxxFybjclw8kjswYcu4pHvOozSN37ok1y8oyAwON2o2wZogD9mm1HHVJtjw/LVntueQ=</HostId>
</Error>
```

# Updating S3 Permissions

Once I set up my OAC, I still needed to update my bucket policy because 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 provides CloudFront access to objects stored in the S3 bucket.

```
Policy
1▼ [
2    "Version": "2008-10-17",
3    "Id": "PolicyForCloudFrontPrivateContent",
4▼   "Statement": [
5▼     {
6        "Sid": "AllowCloudFrontServicePrincipal",
7        "Effect": "Allow",
8▼       "Principal": {
9          "Service": "cloudfront.amazonaws.com"
10         },
11        "Action": "s3:GetObject",
12        "Resource": "arn:aws:s3:::network-three-tier-melvin-052725/*",
13▼       "Condition": {
14▼         "StringEquals": {
15            "AWS:SourceArn": "arn:aws:cloudfront::812541449234:distribution/E3CMRERPMIGBVC"
16          }
17        }
18      }
19    ]
20]
```

# S3 vs CloudFront for Hosting

For my project extension, I'm comparing S3 and CloudFront based on the hosted website's URLs, permission settings and performance. I initially had an error with static website hosting because public access is blocked.

I tried resolving this by removing block all public access. I still ran into an error because I didn't grant permission to access the objects.

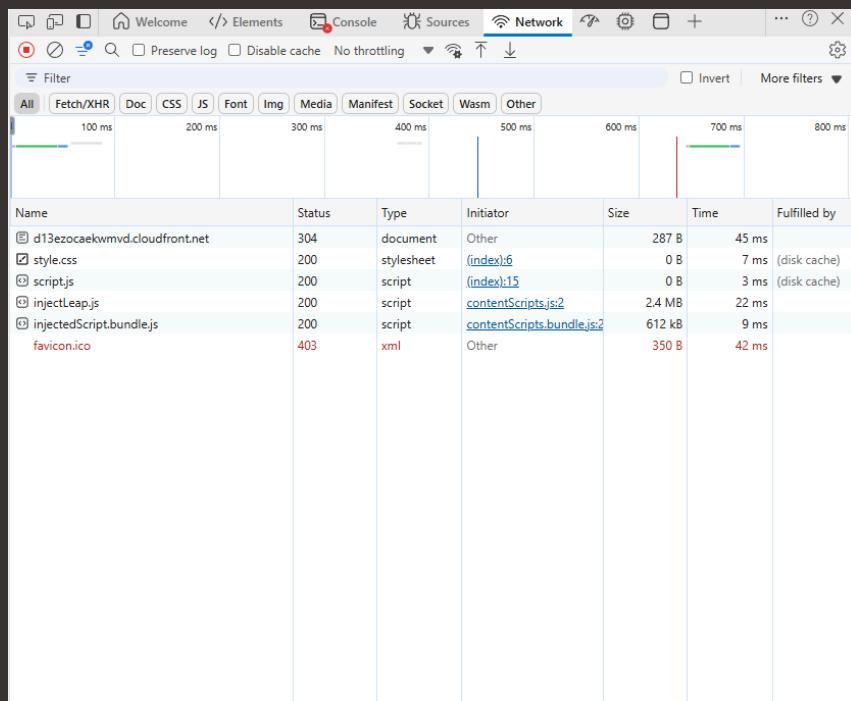
I could finally see my S3 hosted website when I updated the bucket policy. This worked because I allowed access to the objects.

Compared to the permission settings for my CloudFront distribution, using S3 meant I am giving access to the files in the bucket to anyone on the internet. I preferred the permission setting of CloudFront.

# S3 vs CloudFront Load Times

Load times means how quickly the content on your website loads. The load times for CloudFront site were faster than the S3 site because CloudFront's CDN caches content closer to users and S3 static website hosting serves files directly from a single region.

A business would prefer CloudFront when users are located far from their S3 bucket's region. S3 static website hosting might be sufficient when the users are located in the same region as the S3 bucket.





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

# The place to learn & showcase your skills

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

