

PROJECT REPORT

DEPLOYMENT OF A REACT APP USING AWS S3 AND CLOUDFRONT

PREPARED BY: NDUBUISI AGUGHARA

DATE: 17TH /2/ 2025



1. Introduction

Deploying a React application on AWS S3 and CloudFront is a highly scalable and cost-effective solution for hosting static web applications. AWS S3 serves as a storage solution for the application files, while AWS CloudFront provides a content delivery network (CDN) to ensure low-latency access and improved performance for users across different regions.



Ndubuisi AgugharaJob

2. Project Objectives

The primary objectives of this project include:

- Hosting a React application using AWS services.
- Utilizing Amazon S3 for static file storage.
- Implementing AWS CloudFront for efficient content delivery and caching.
- Ensuring security, scalability, and cost-efficiency.



Ndubuisi AgugharaJob

3. Technologies Used

- React.js: Front-end JavaScript framework for building the application.
- AWS S3: Object storage service used to store static files.
- AWS CloudFront: CDN service to distribute the content globally with low latency.
- AWS Route 53: (Optional) Domain Name System (DNS) service for custom domain configuration.
- AWS Certificate Manager (ACM): For securing the website using SSL/TLS.



Ndubuisi AgugharaJob

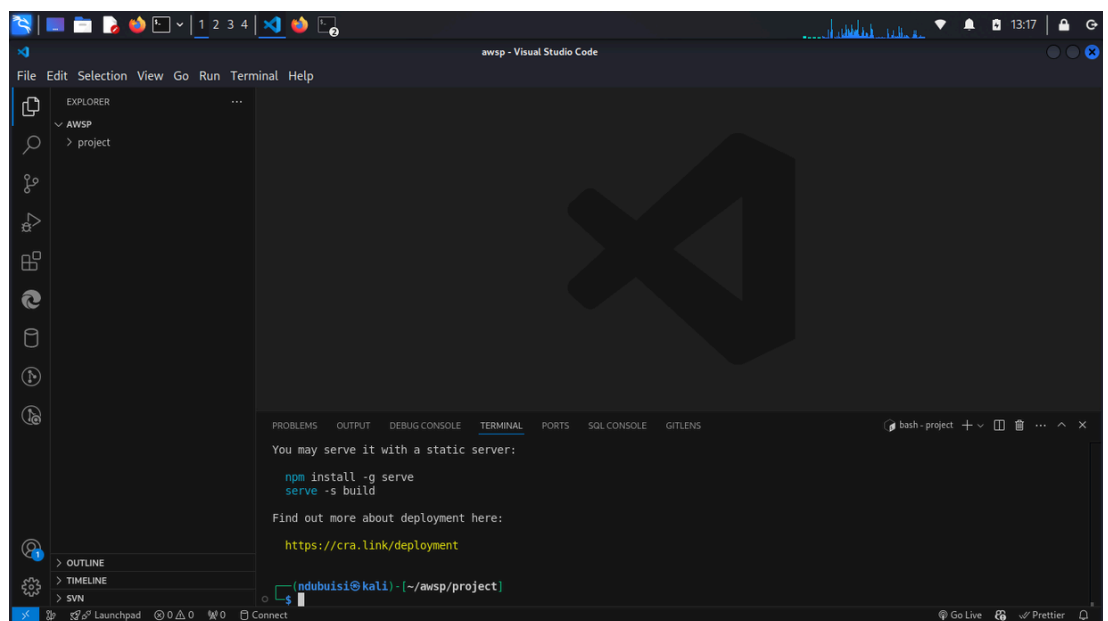
SETTING UP REACT APP

With the command 'npx create-react-app my-project' I created a React application, which sets up a new project structure along with default configurations and dependencies and initial scripts.

I used the npm run build command to generate static files. It builds the React application and optimizes it for production by minifying the JavaScript, CSS etc.

The output is in the build/ directory which has all the files required to deploy the app to a web server such as AWS S3.

The static files provided by React enable the application to operate efficiently in production without the requirement of a Node.js server



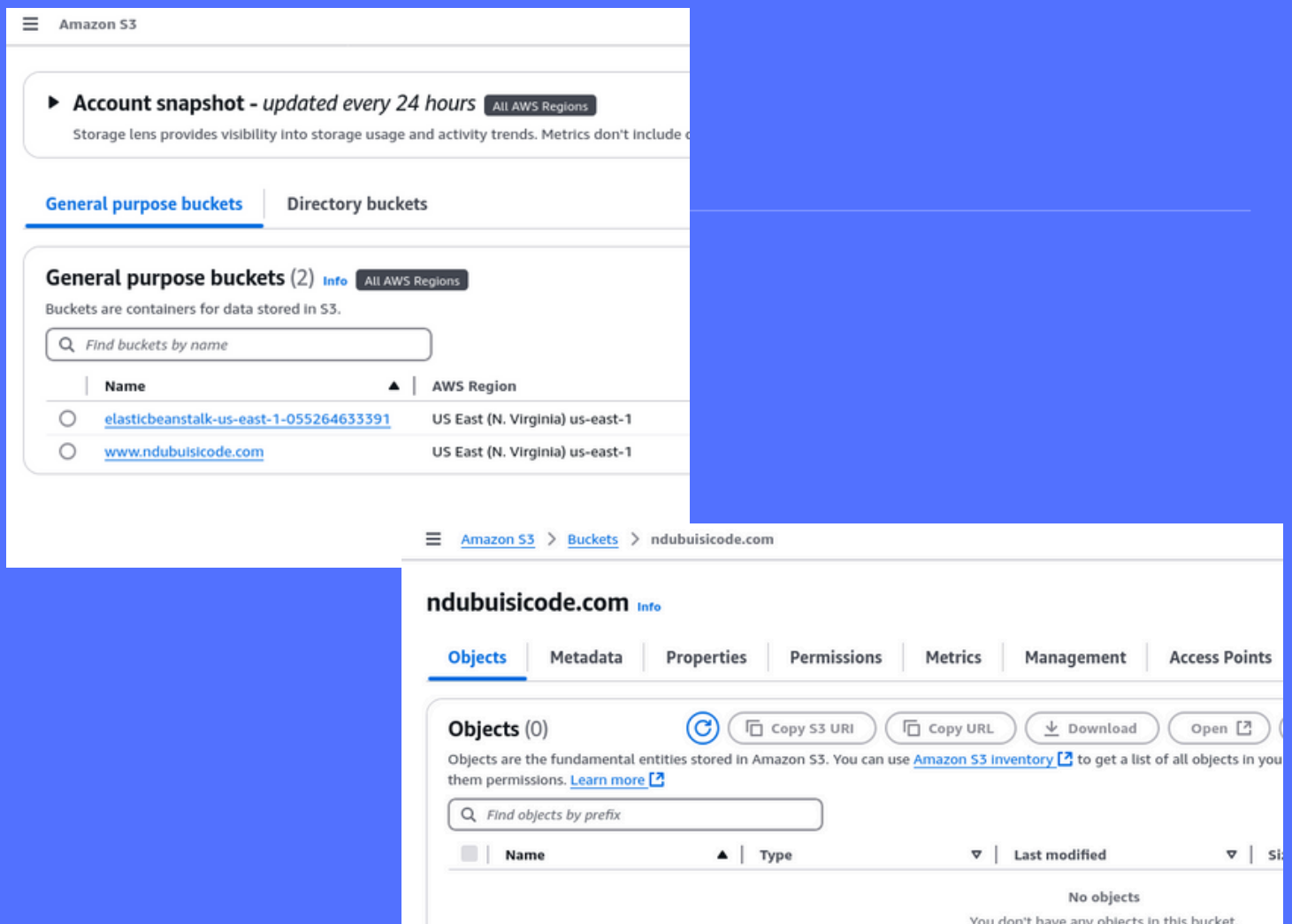
Ndubuisi AgugharaJob

Configure AWS S3 Bucket

For this project, I created two separate S3 buckets:

1. **Primary Bucket** – This bucket is named after the website domain without the "www" prefix (e.g., mywebsite.com). It serves as the authoritative source, hosting the static files for the React application.
2. **Redirect Bucket** – This bucket includes the "www" prefix (e.g., www.mywebsite.com). It is configured to redirect all traffic to the primary bucket, ensuring a seamless user experience regardless of whether visitors use "www" in the URL.

FIRST BUCKET I CREATED



The image shows two screenshots of the Amazon S3 console. The top screenshot displays the 'General purpose buckets' section, listing two buckets: 'elasticbeanstalk-us-east-1-055264633391' and 'www.ndubuisicode.com', both located in the 'US East (N. Virginia) us-east-1' region. The bottom screenshot shows the details of the 'ndubuisicode.com' bucket, which is currently empty (0 objects). The bucket's configuration page includes tabs for Objects, Metadata, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is active, showing a search bar and a table with columns for Name, Type, Last modified, and Size. A message at the bottom states 'No objects' and 'You don't have any objects in this bucket.'

SECOND BUCKET I CREATED

BUCKET NAME:

WWW.NDUBUISICODE.COM

www.ndubuisicode.com [Info](#)

[Objects](#)

[Metadata](#)

[Properties](#)

[Permissions](#)

[Metrics](#)

[Management](#)

Objects (0)



Copy S3 URI



Copy URL



Download

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get them permissions. [Learn more](#)



Find objects by prefix



Name



Type



Last modified

No objects

You don't have any objects

[Upload](#)

General purpose buckets (3) [Info](#)

All AWS Regions

Buckets are containers for data stored in S3.



Find buckets by name



Name



AWS Region



[elasticbeanstalk-us-east-1-055264633391](#)

US East (N. Virginia) us-east-1



[ndubuisicode.com](#)

US East (N. Virginia) us-east-1



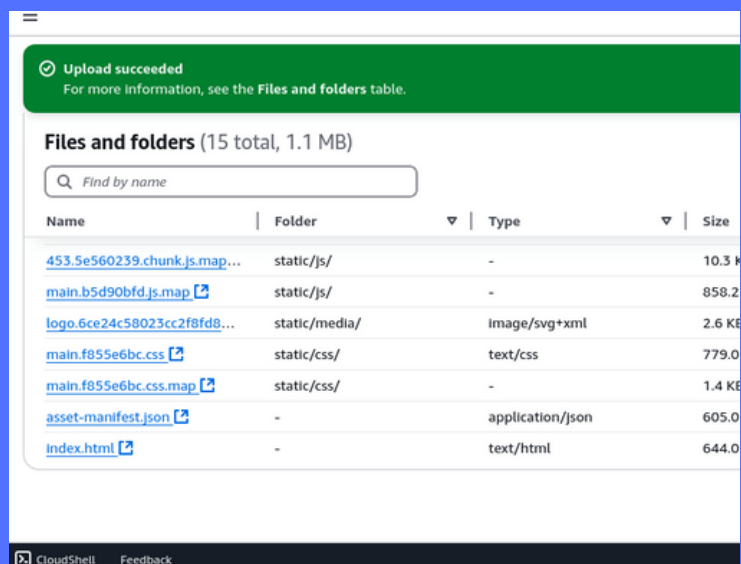
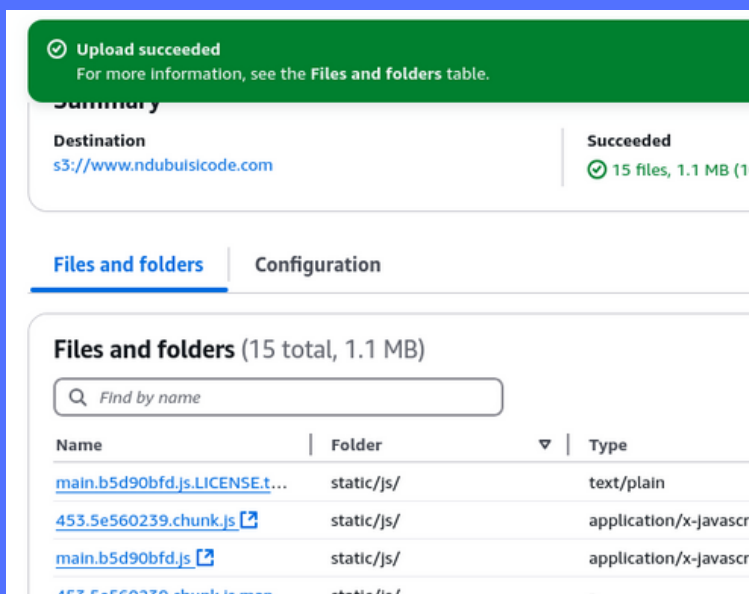
[www.ndubuisicode.com](#)

US East (N. Virginia) us-east-1

Uploaded the React build files to the S3 bucket.

uploaded the React build files to the primary S3 bucket, ensuring that all static assets, including HTML, CSS, and JavaScript files, are available for hosting. This allows the application to be served directly from S3.

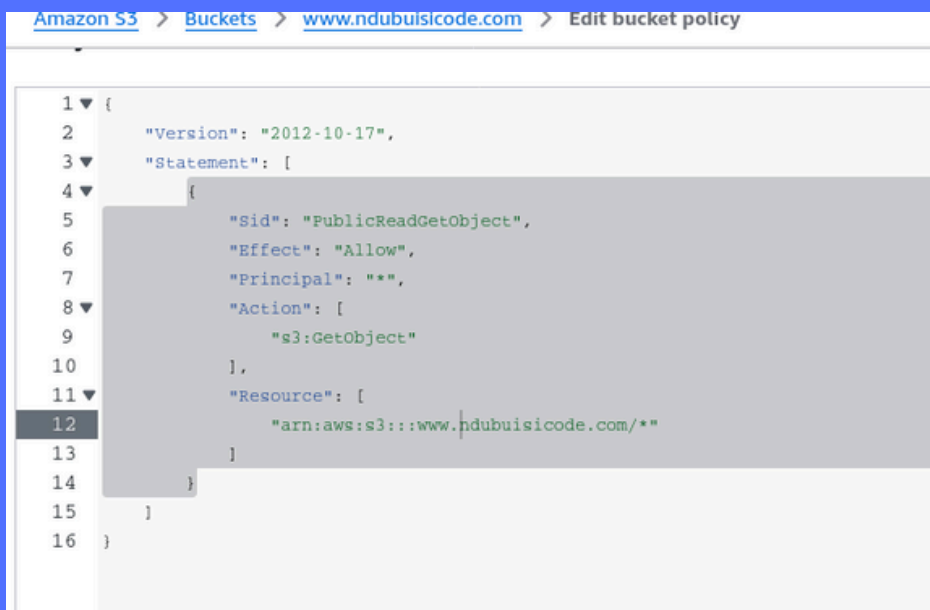
Additionally, I configured the bucket for public access (if needed) and enabled static website hosting to serve the React app properly.



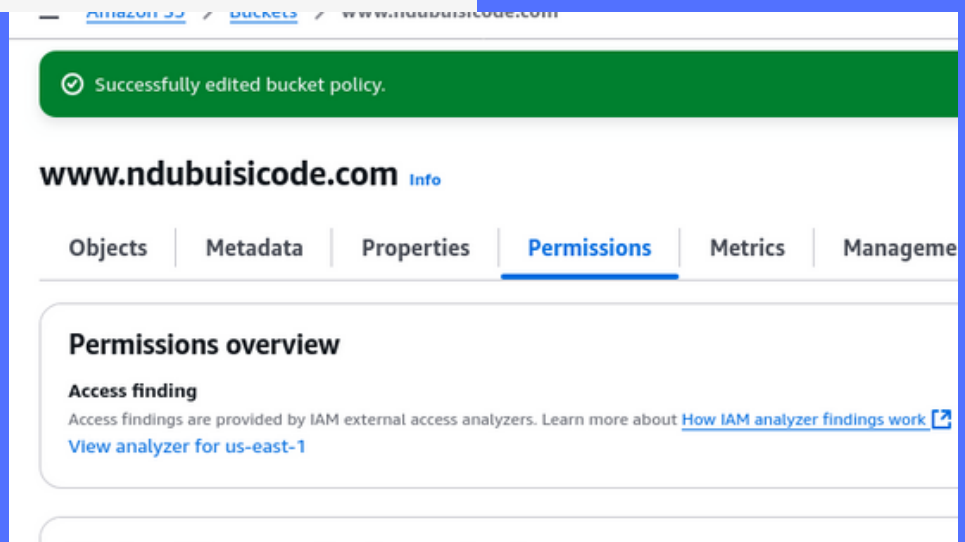
Configured bucket policies to allow public read access (or used AWS IAM policies for access control).

I configured the S3 bucket policies to allow public read access, enabling users to access the React app through a web browser. This involved updating the bucket policy to grant `s3:GetObject` permissions for all objects in the bucket while ensuring security best practices.

Alternatively, I could use AWS IAM policies to manage access control, restricting permissions to specific users or services while maintaining the necessary access for public content delivery.



```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "PublicReadGetObject",  
6       "Effect": "Allow",  
7       "Principal": "*",  
8       "Action": [  
9         "s3:GetObject"  
10      ],  
11      "Resource": [  
12        "arn:aws:s3:::www.ndubuisicode.com/*"  
13      ]  
14    }  
15  ]  
16 }
```





My Error

I encountered an error when trying to access my link, which resulted in a blank page.

Link: <https://s3.us-east-1.amazonaws.com/www.ndubuisicode.com/index.html>.



This is because is loading just the file and not all the whole react file.

But at this stage , the bucket is public.

Edit static website hosting

Edit static website hosting [Info](#)

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable

☒ Enable

Hosting type

☒ Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Info For your customers to access content at the website endpoint, you must make all your content more information, see [Using Amazon S3 Block Public Access](#)

Configure records

To make the S3-hosted React app accessible via its domain, I configured DNS records in AWS Route 53.

You can create multiple records at a time that have the same routing policy.

Simple routing records to add to ndubuisicode.com [Info](#)

[Edit](#)[Delete](#)

Use if you want all of your clients to receive the same response(s).

<input type="checkbox"/>	Record name	Type	Value/Route traffic to
<input type="checkbox"/>	www.ndubuisicode.com	A	s3-website-us-east-1.amazo...

► Existing records

[Cancel](#)

Records (4) [Info](#)

[Delete record](#)[Import zone file](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes](#)

[Type ▼](#)[Routing p... ▼](#)[Alias ▼](#)

<input type="checkbox"/>	Record ... ▼	Type ▼	Routin... ▼	Differ... ▼	Alias ▼	V
<input type="checkbox"/>	ndubuisic...	A	Simple	-	Yes	s3
<input type="checkbox"/>	ndubuisic...	NS	Simple	-	No	ns

Simple routing records to add to ndubuisicode.com [Info](#)

Use if you want all of your clients to receive the same response(s).

<input type="checkbox"/>	Record name	Type
<input type="checkbox"/>	www.ndubuisicode.com	A
<input type="checkbox"/>	ndubuisicode.com	A

► Existing records

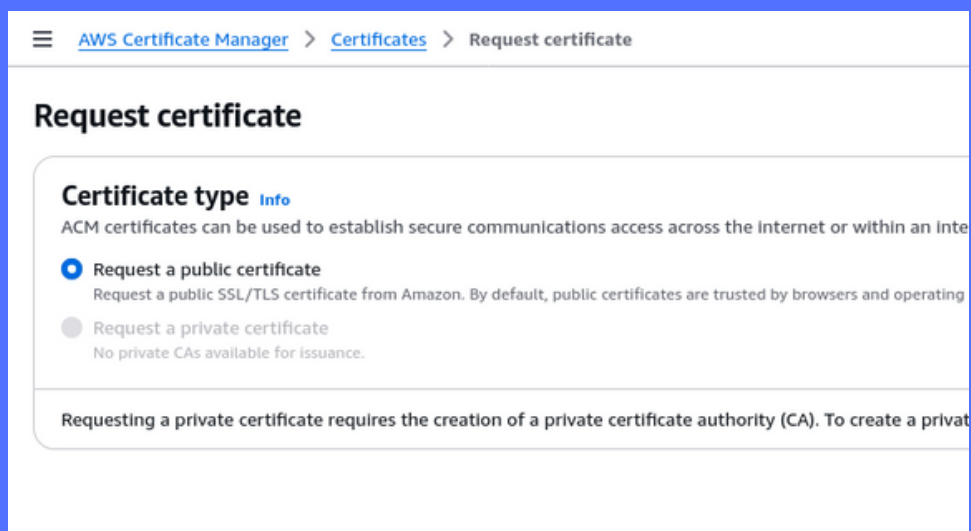
Set Up AWS CloudFront

To improve performance, security, and global accessibility, I integrated AWS CloudFront with the S3-hosted React app:

why cloudfrent

- Performance Improvement (CDN)
- Caches content at edge locations for faster load times.
- Global Distribution
- Serves content from a global network of edge locations, reducing latency.
- Scalability
- Handles traffic spikes without affecting performance.
- Security
- SSL/TLS encryption for secure HTTPS access.
- Restricts direct S3 access to CloudFront.
- Provides automatic DDoS protection.
- Cost Savings
- Reduces data transfer costs by caching content and offloading traffic from

Set Up Our Certificate From Our Certificate Manager



Set Up Our Certificate From Our Certificate Manager

☰ AWS Certificate Manager > Certificates > Request certificate

Request certificate

Certificate type Info

ACM certificates can be used to establish secure communications access across the Internet or within an internal network.

☒ Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

☐ Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, see [Requesting a private certificate](#).

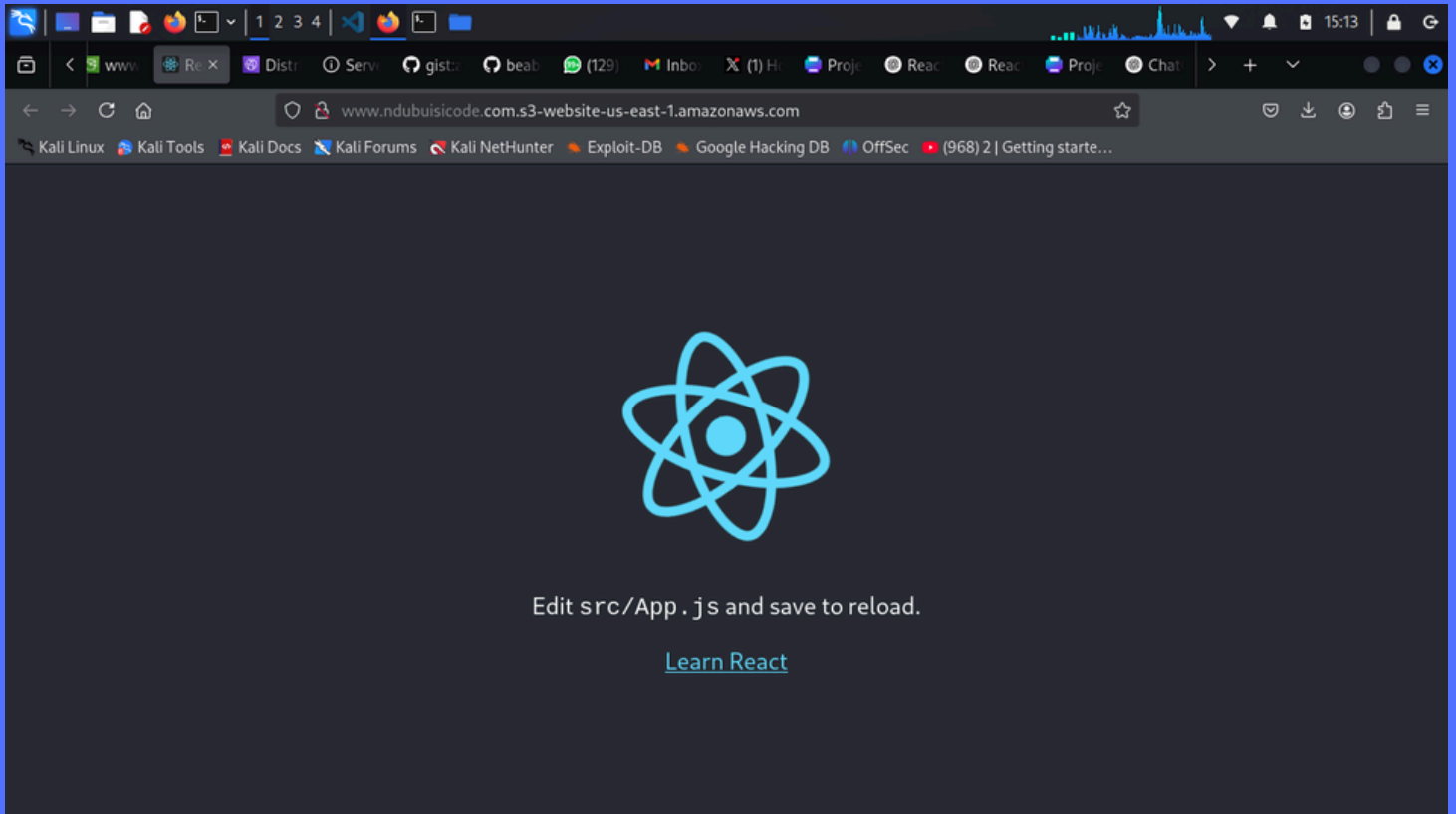
Successfully requested certificate with ID 8b466dca-5685-4d27-a08a-0e5145b31d99
A certificate request with a status of pending validation has been created. Further action is needed for validation and approval of the certificate.

8b466dca-5685-4d27-a08a-0e5145b31d99

Certificate status

Identifier	8b466dca-5685-4d27-a08a-0e5145b31d99	Status	Pending validation
ARN	arn:aws:acm:us-east-1:055264633391:certificate/8b466dca-5685-4d27-a08a-0e5145b31d99		
Type	Amazon Issued		

Filter records by property or value	Type	Routing p...	Alias		
Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic
<input type="checkbox"/> ndubuisic...	A	Simple	-	Yes	s3-website-us-east-
<input type="checkbox"/> ndubuisic...	NS	Simple	-	No	ns-1768.awsdns-29 ns-1227.awsdns-25 ns-490.awsdns-61.c ns-606.awsdns-11.r
<input type="checkbox"/> ndubuisic...	SOA	Simple	-	No	ns-1768.awsdns-29
<input type="checkbox"/> _beb3155...	CNAME	Simple	-	No	_2470cc6af5d5d0f5
<input type="checkbox"/> www.ndu...	A	Simple	-	Yes	s3-website-us-east-
<input type="checkbox"/> _02ee8b8...	CNAME	Simple	-	No	_53f630217af15a8



Summary

This project report details the deployment of a React application using AWS S3 and CloudFront. The goal was to create a scalable, cost-effective, and secure hosting solution.

- AWS S3 was used for storing static files, while CloudFront improved performance through global content distribution.
- The setup included IAM policies for security, CloudFront caching for optimization, and SSL/TLS encryption for secure access.
- Optional configurations, such as Route 53 for domain management and AWS Certificate Manager for SSL, were also considered.
- The deployment was optimized for performance, security, and cost-efficiency, making it a reliable solution for hosting static React applications.

This approach ensures minimal latency, reduced costs, and enhanced reliability for web applications.