Hosting Website using Amazon S3 and accelerate content delivery using Cloudfont in AWS

A Course End Project Report in Cloud Computing and Virtualization Laboratory

(Course Code - A8524)

Submitted in the Partial Fulfilment of the

Requirements

for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted

By

M.Mani Prasoon	**22881A05N8**
D.Sai Praneeth	**22881A05M1**
M.Hari Vardhan	**22881A05P1**

Under the Esteemed Guidance of

Dr. Ravula Arun Kumar
Assistant Professor

VARDHAMAN
COLLEGE OF ENGINEERING

VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

April, 2025

VARDHAMAN
COLLEGE OF ENGINEERING

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Course End Project titled **"Hosting a website using Amazon S3 and accelerate content delivery using Cloudfront in AWS"** is carried out by **"M.Mani Prasoon"** **with** Roll Number **"22881A05N8", "D.Sai Praneeth"** **with** Roll Number **"22881A05M1", "M.Hari Vardhan"** **with** Roll Number **"22881A05P1"** towards **A8524 –Cloud Computing and Virtualization Laboratory** course in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** during the Academic year 2024-25.

Signature of the Course Faculty
Dr. Ravula Arun Kumar
Assistant Professor, CSE

Signature of the HOD

Dr. Ramesh Karnati
HOD, CSE

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr. Ravula Arun Kumar** Assistant Professor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for her able guidance and useful suggestions, which helped us in completing the design part of potential project in time.

We particularly thankful to **Dr. Ramesh Karnati**, Associate Professor & Head, Department of Computer Science and Engineering for his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr.Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

M.Mani Prasoon – 22881A05N8
D.Sai Praneeth--22881A05M1
M.Hari Vardhan--22881A05P1

INDEX

1. Introduction.....1
2. AWS Services Used.....1
3. Architecture Diagram.....2
4. Project Implementation steps.....2
5. Security Configuration.....3
6. Testing and Output.....3
7. Results.....4
8. Challenges Faces.....4
9. Conclusion.....4
10. References.....9

1. Introduction

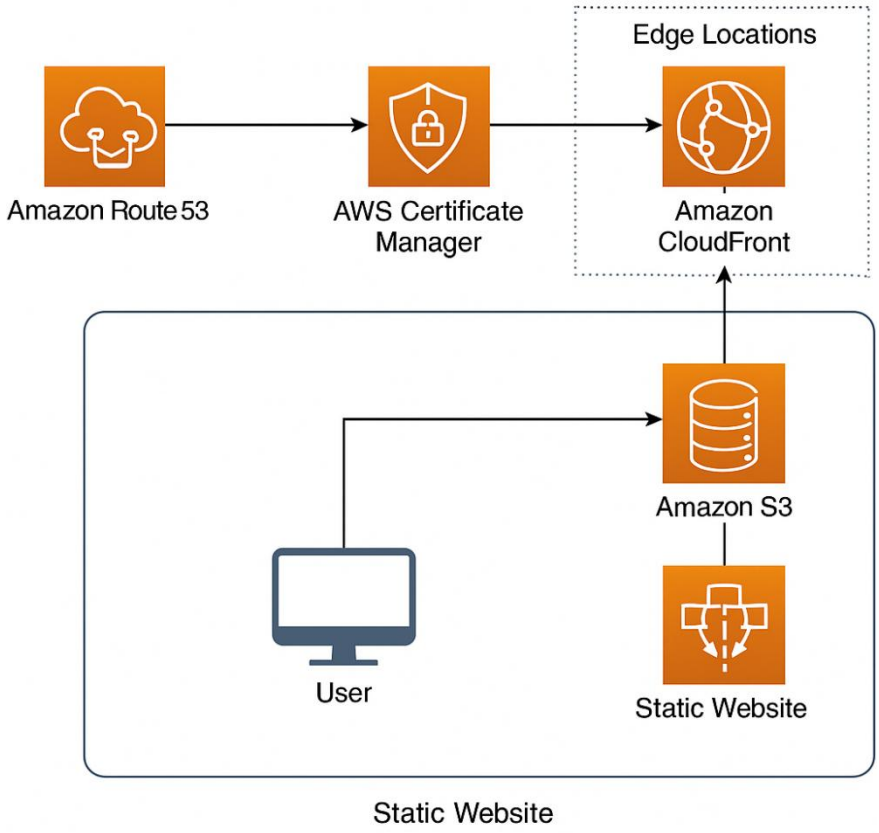
This project focuses on deploying a highly scalable, low-latency static website using Amazon Web Services (AWS). Leveraging Amazon S3 (Simple Storage Service), we will host static assets—HTML, CSS, JavaScript, and media files—ensuring durability, high availability, and cost-efficiency. To supercharge performance and deliver content globally with minimal latency, Amazon CloudFront—a content delivery network (CDN)—will be integrated.

By combining S3's serverless architecture with CloudFront's edge network, the project aims to deliver a blazing-fast, secure, and highly available static web experience. It also embraces modern DevOps practices by enabling versioning, access control, and automatic content invalidation. This setup is ideal for websites, portfolios, landing pages, or any static content that demands global reach and enterprise-grade performance.

2. AWS Services Used

AWS Service	**Purpose**
Amazon S3 (Simple Storage Service)	Host the static website files (HTML, CSS, JS, images). Provides durable, scalable, and cost-effective storage with static website hosting capability.
Amazon CloudFront	Distribute content globally with low latency and high transfer speed via edge locations. Accelerates delivery and reduces load on S3.
AWS Certificate Manager (ACM)	Provision and manage SSL/TLS certificates for enabling HTTPS on the CloudFront distribution.

3. Architecture Diagram



4. Project Implementation Steps

1.Create and Configure S3 Bucket

Create an S3 bucket with a name matching your domain (e.g., www.example.com).

Enable static website hosting.

Upload your static website files (HTML, CSS, JS).

Set bucket policy to allow public read access (or use CloudFront OAI for restricted access).

2.Configure Bucket Permissions

Add a bucket policy to allow public access or CloudFront origin access identity (OAI).

Disable block public access settings if allowing direct S3 access.

3.Set Up CloudFront Distribution

Create a CloudFront distribution.

Set the S3 bucket as the origin.

Enable HTTPS and configure default root object (e.g., index.html).

Enable caching, compression, and geographic restrictions if needed.

4.Request and Attach SSL Certificate

Use AWS Certificate Manager (ACM) to generate an SSL certificate for your domain.

Attach the certificate to your CloudFront distribution.

5.Test the Website

Verify public access via S3 and/or CloudFront.

Check for HTTPS, performance, and correct rendering.

6.Optimize & Monitor

Enable CloudFront logging and CloudWatch for analytics.

Set invalidations for cache refresh during updates.

5.Security Configuration

1.S3 Bucket Security

Block Public Access: Enable block all public access unless using CloudFront OAI.

Bucket Policy: If using OAI, remove public-read permissions and allow access only to CloudFront via Origin Access Identity or Origin Access Control (OAC).

Enable Versioning: Protect against accidental overwrites or deletions.

2.CloudFront Security

Use HTTPS: Enforce HTTPS using viewer protocol policy (Redirect HTTP to HTTPS).

Attach SSL Certificate: Use AWS Certificate Manager (ACM) for free TLS certs.

Enable Signed URLs/Cookies: Restrict access to premium/private content if needed.

Enable Geo-Restriction: Block access from unwanted countries.

5. Testing and Output

1.S3 Website Access

Open the S3 website link.

It should load your site if public access is allowed.

If using CloudFront with restricted access, the S3 link should not work directly.

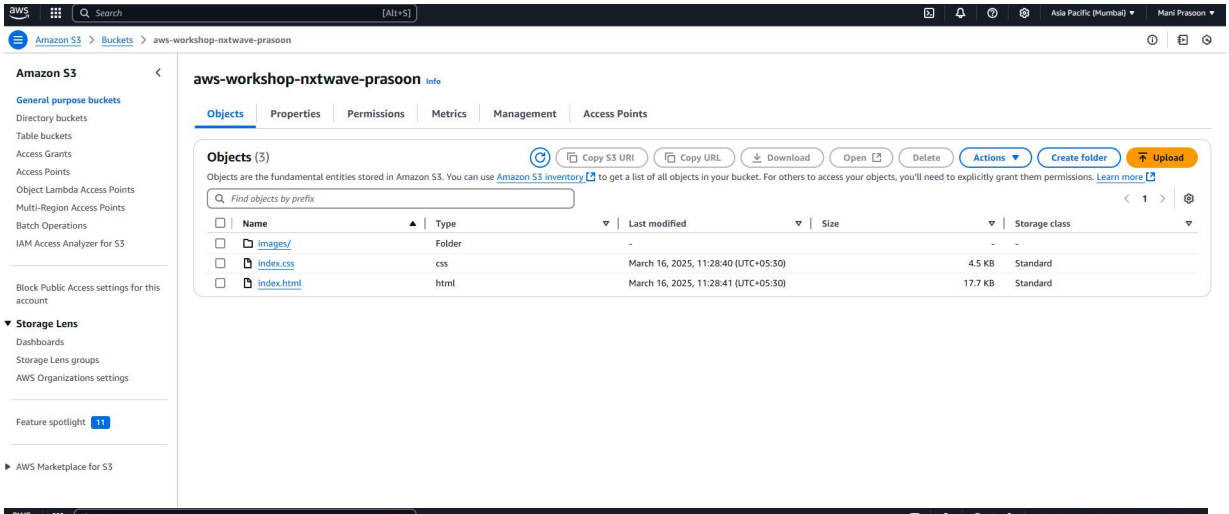
2.CloudFront Access

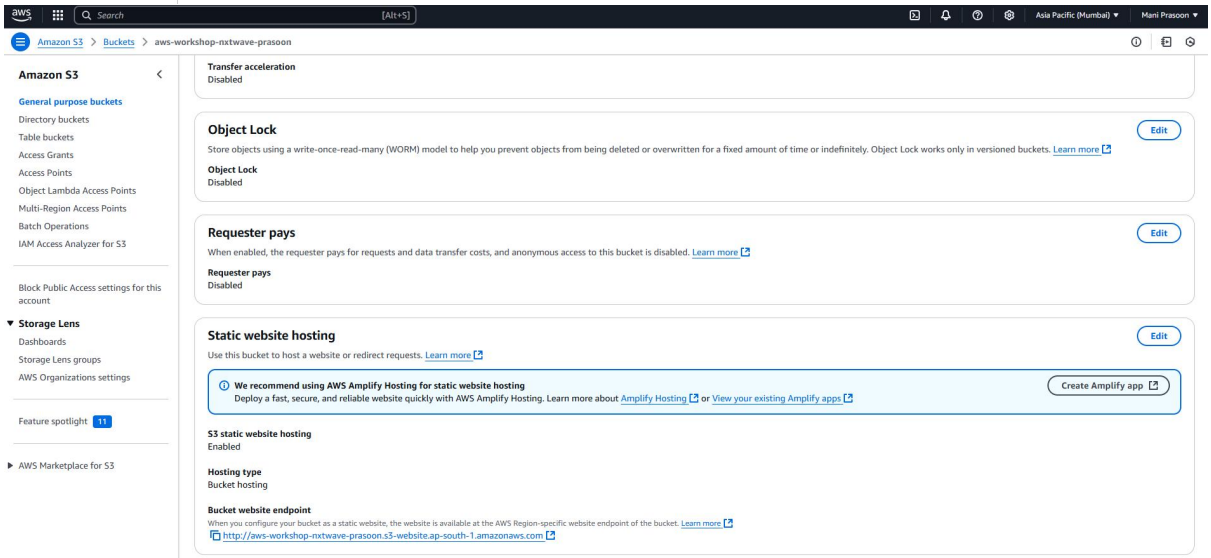
Visit the CloudFront URL (or your custom domain).

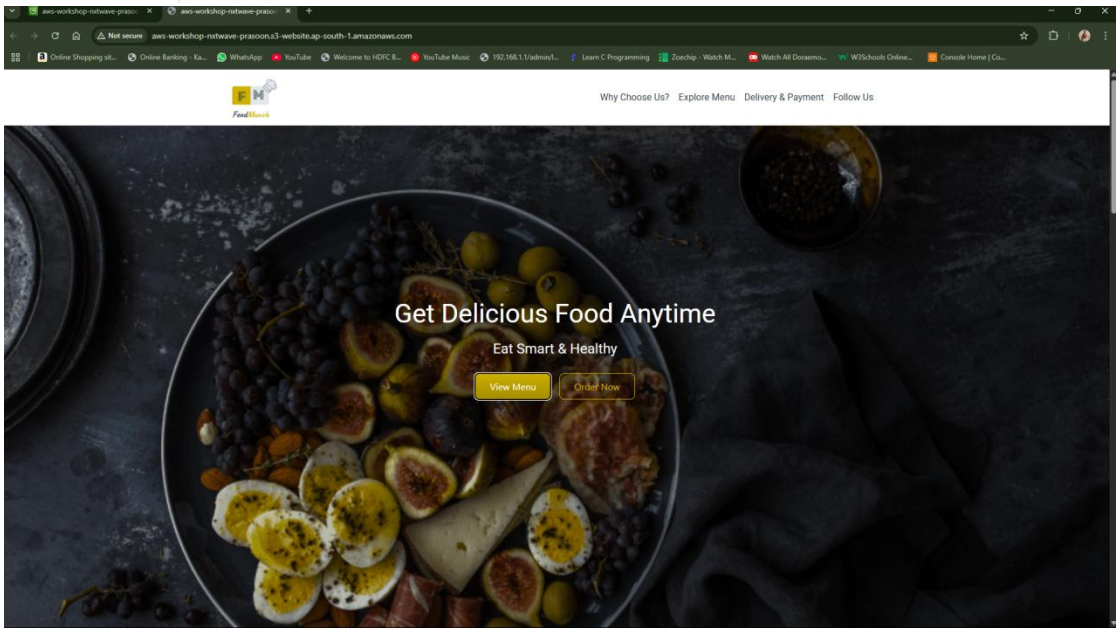
The website should load quickly and securely

6. Results:

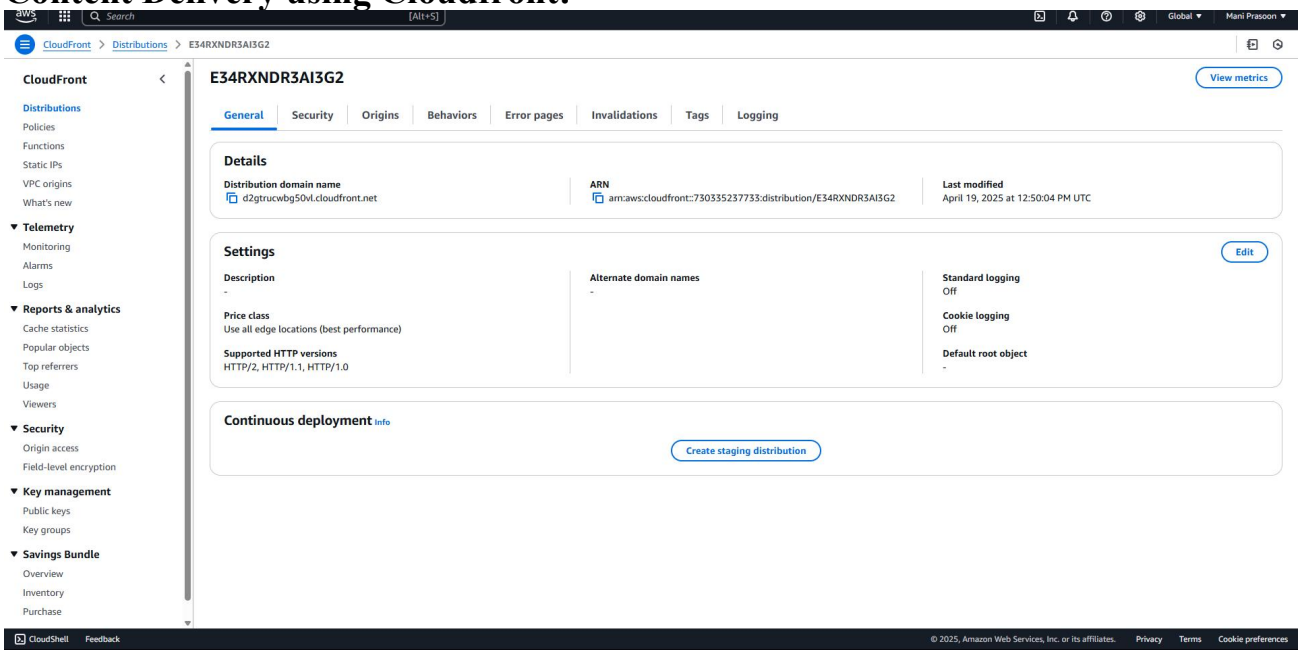
Hosting website using Amazon S3:

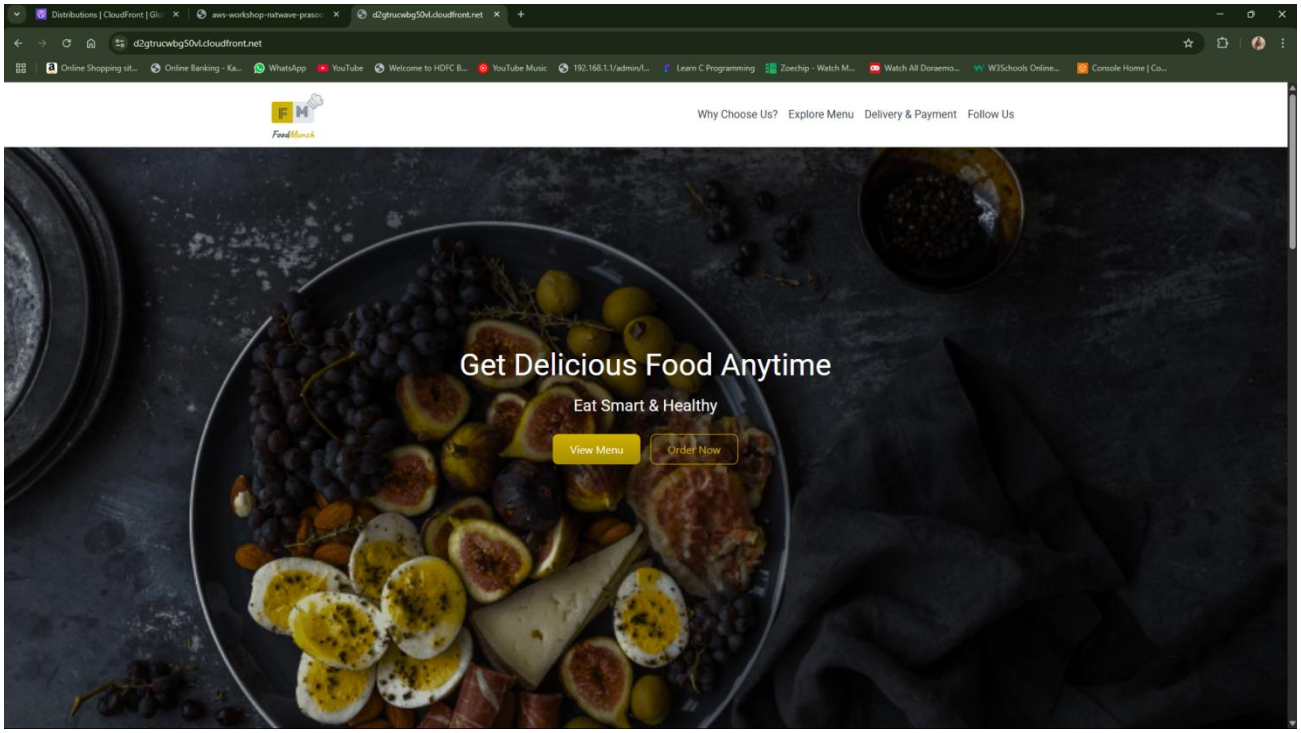

Search [Alt+S]
Asia Pacific (Mumbai)
Mani Prasoon
Amazon S3 > Buckets > aws-workshop-nxtwave-prasoon
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
aws-workshop-nxtwave-prasoon Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (3)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
images/ Folder - - -
index.css css March 16, 2025, 11:28:40 (UTC+05:30) 4.5 KB Standard
index.html html March 16, 2025, 11:28:41 (UTC+05:30) 17.7 KB Standard


Search [Alt+S]
Asia Pacific (Mumbai)
Mani Prasoon
Amazon S3 > Buckets > aws-workshop-nxtwave-prasoon
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Transfer acceleration
Disabled
Object Lock
Edit
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. Learn more
Object Lock
Disabled
Requester pays
Edit
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more
Requester pays
Disabled
Static website hosting
Edit
Use this bucket to host a website or redirect requests. Learn more
We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps
Create Amplify app
S3 static website hosting
Enabled
Hosting type
Bucket hosting
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://aws-workshop-nxtwave-prasoon.s3-website.ap-south-1.amazonaws.com


Not secure
aws-workshop-nxtwave-prasoon.s3-website.ap-south-1.amazonaws.com
FoodMunch
Why Choose Us?
Explore Menu
Delivery & Payment
Follow Us
Get Delicious Food Anytime
Eat Smart & Healthy
View Menu
Order Now

Content Delivery using Cloudfront:





7. Challenges Faced

1.S3 Public Access Restrictions

AWS blocks public access by default; configuring bucket policies correctly can be tricky, especially when using CloudFront with OAI or OAC.

2.CloudFront Propagation Delay

Changes to the distribution (like behavior updates or certificate attachments) can take 10–30 minutes to deploy globally.

8.Conclusion

Hosting a static website using Amazon S3 and accelerating content delivery with CloudFront offers a powerful, scalable, and cost-efficient solution. By combining the simplicity of S3's static hosting with the speed and global reach of CloudFront, the setup delivers fast, secure, and highly available web experiences.

Through proper configuration of security, caching, SSL, and DNS, the project demonstrates how modern cloud-native architectures can replace traditional web hosting with serverless, maintenance-free infrastructure. Despite a few setup complexities, the end result is a production-grade static website that's optimized for performance, resilience, and global access.

9.References

- **Amazon S3 Documentation**
https://docs.aws.amazon.com/s3/index.html

- **Amazon S3 Static Website Hosting**
https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html

- **Amazon CloudFront Documentation**
https://docs.aws.amazon.com/cloudfront/index.html

- **Using CloudFront with Amazon S3**
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/DownloadDistS3AndCustomOrigin.html