AWS Static Website Hosting – Project Documentation

---

1. Project Name

End-to-End Static Website Hosting using AWS S3, CloudFront, Route 53, ACM, and GoDaddy

---

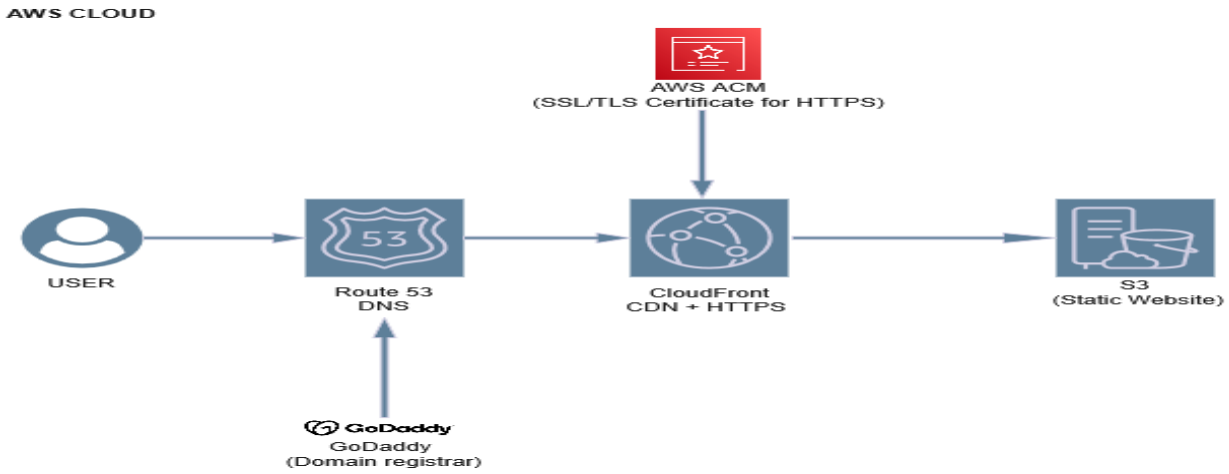
2. Introduction

This project demonstrates how to deploy and secure a static website using multiple AWS services integrated together.
The objective was to design a low-cost, highly available, and globally distributed static website architecture mapped to a custom domain.

I used **Amazon S3** to host the static files, **Amazon CloudFront** as the CDN for global performance, **AWS Certificate Manager (ACM)** for SSL/TLS encryption, and **Amazon Route 53** for DNS management.
The domain was registered with **GoDaddy** and delegated to Route 53 nameservers.

This is the same type of setup I've implemented several times in real client environments — combining cost efficiency, performance, and best-practice security.

Architecture Overview



Key Features

- End-to-end HTTPS with AWS-managed SSL certificates
- Custom domain (aniket123.shop) integrated via Route 53
- Global content caching and acceleration using CloudFront
- Cost-optimized, serverless, and easy to maintain
- Scalable solution following AWS best practices

---

3. Prerequisites

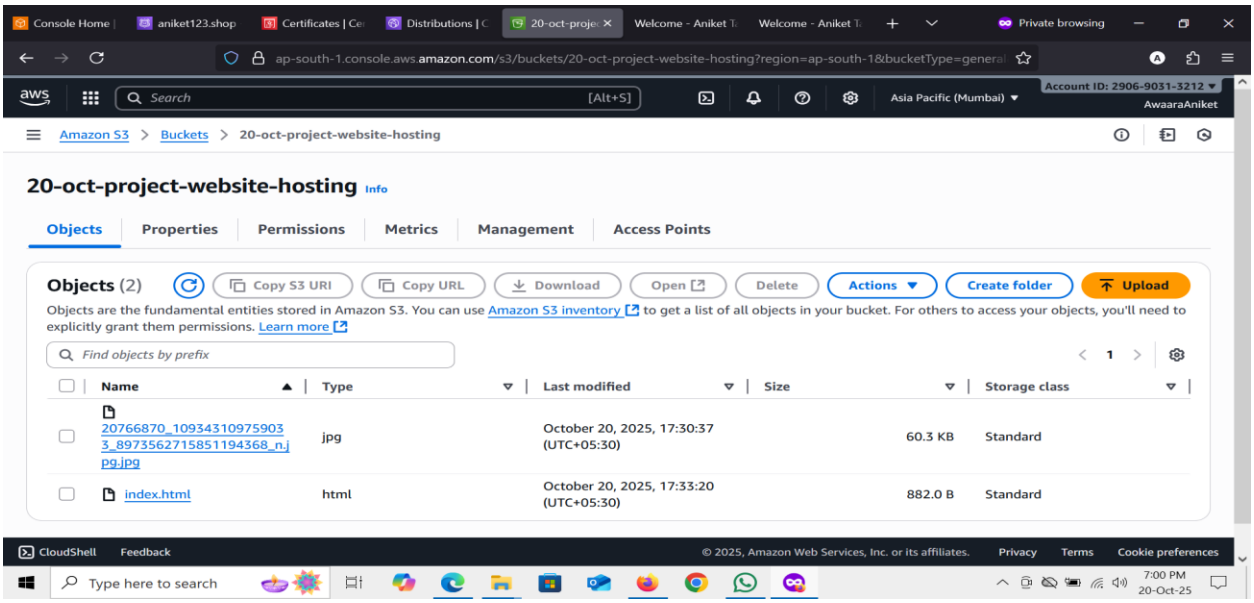
Requirement	Description
AWS Account	Full access to S3, CloudFront, Route 53, and ACM
Domain Name	Purchased from GoDaddy (aniket123.shop)
IAM Role	User with admin or power-user permissions
Browser / Tools	AWS Console, VS Code, Chrome
Knowledge	AWS networking, DNS, SSL basics

4. Project Structure

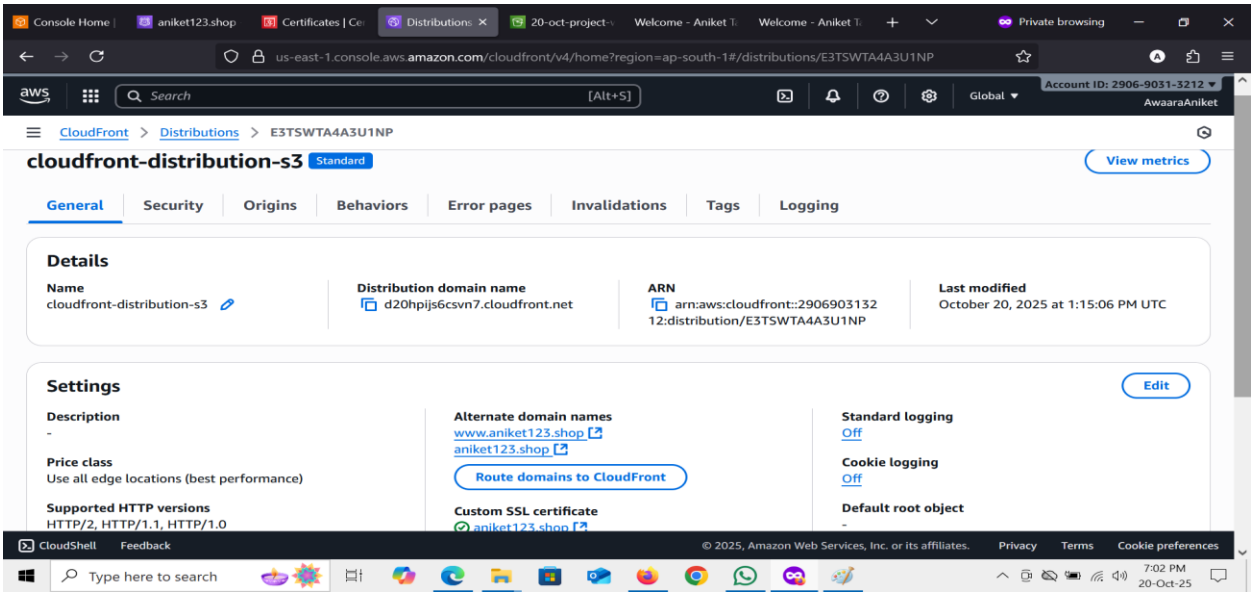
```
aws-static-website-hosting/
├── index.html                # Main static website
├── screenshots/              # AWS configuration screenshots
│   ├── s3-bucket.png
│   ├── cloudfront.png
│   ├── route53.png
│   ├── acm.png
│   └── final-website.png
└── README.md                 # Documentation for GitHub
```

5. Screenshots

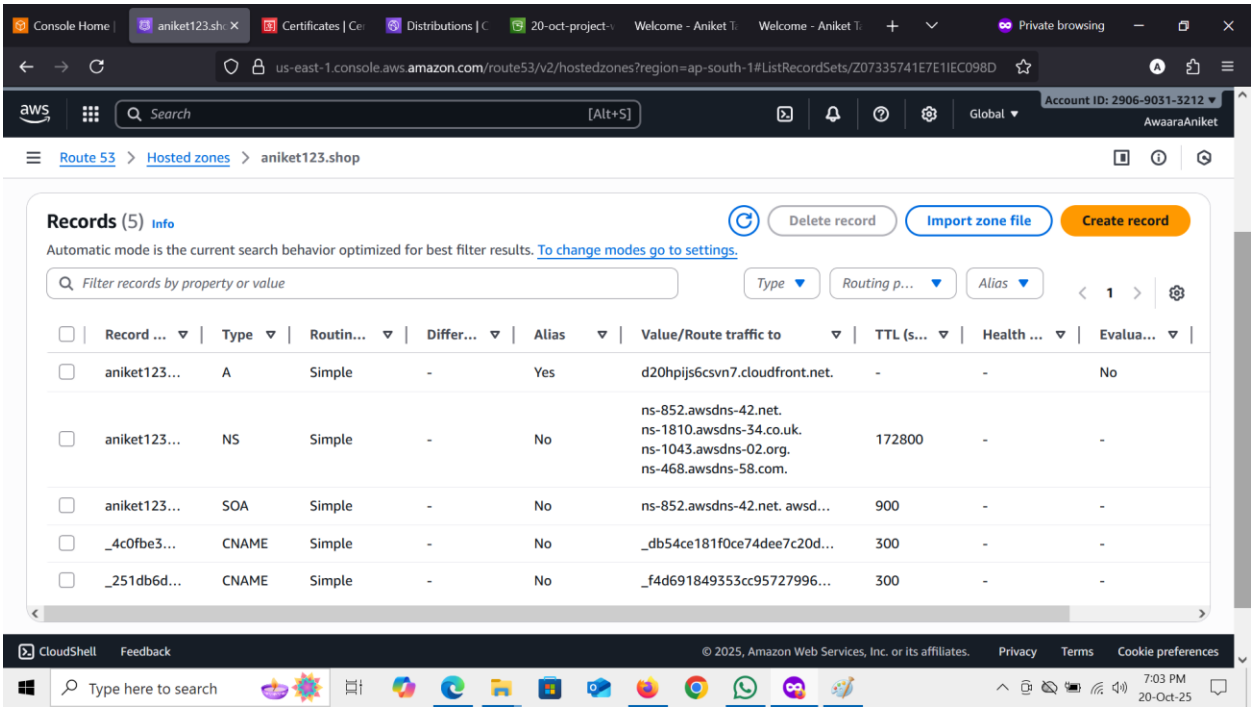
S3 Bucket - Static Website Hosting



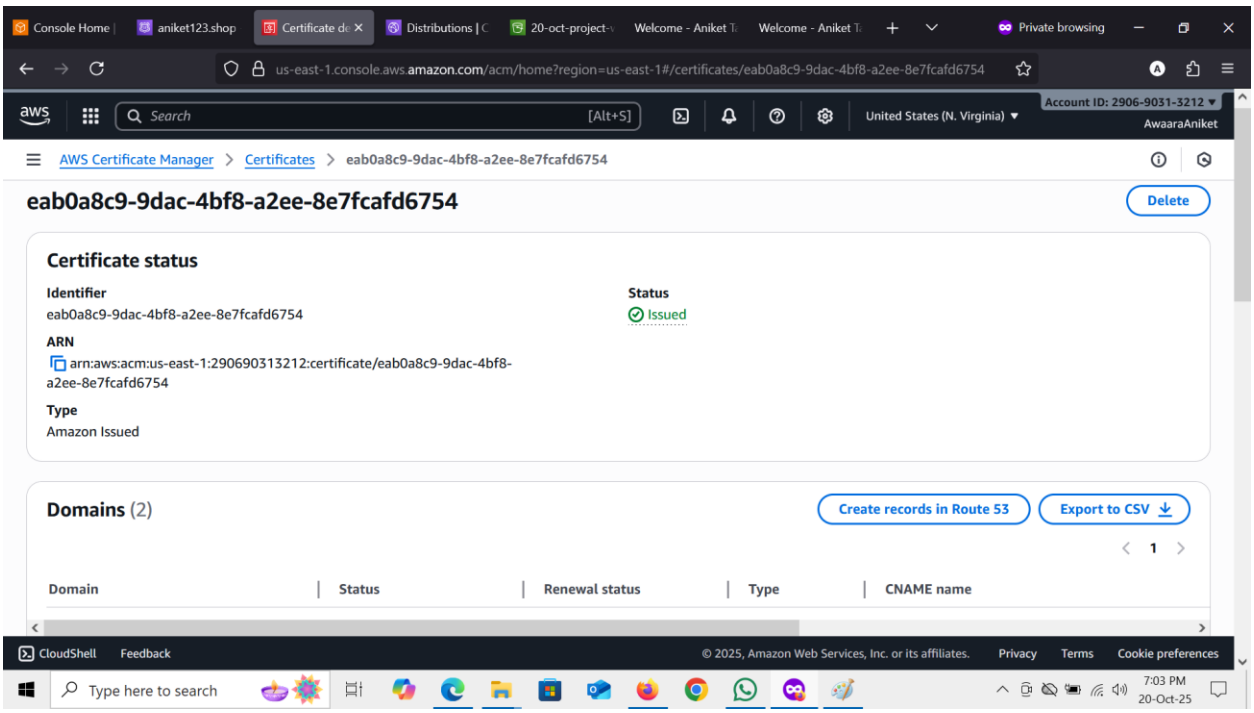
CloudFront Distribution - Deployed



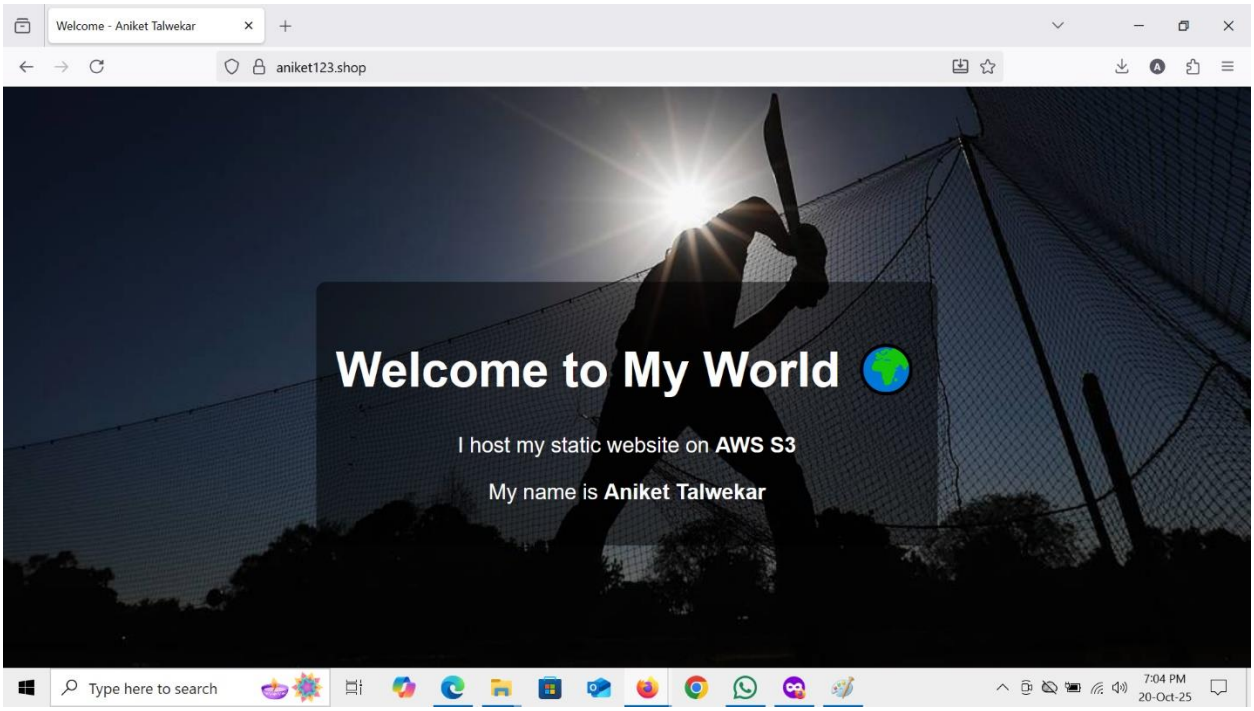
Route 53 Hosted Zone - DNS Records



SSL Certificate – Issued



Final Live Website



6. Explanation

Step 1 – S3 Configuration

Created a dedicated S3 bucket named 20-oct-project-website-hosting and enabled **Static Website Hosting**.
Uploaded the index.html file and adjusted the bucket policy to allow public read access.
I ensured versioning and logging were disabled for cost control but can be enabled in production for compliance.

Step 2 – CloudFront Distribution

Configured a new **CloudFront distribution** using the S3 website endpoint as the origin.
Set Viewer Protocol Policy to **Redirect HTTP to HTTPS** for secure access.
Added Alternate Domain Names (aniket123.shop, www.aniket123.shop) and linked a custom SSL certificate from ACM.
Configured cache behavior and disabled query-string forwarding for optimized caching.

Step 3 – SSL Certificate (ACM)

Requested a **public certificate** from **AWS Certificate Manager** in **us-east-1 (N. Virginia)**, since CloudFront requires certificates from this region.
Validated ownership using DNS by adding CNAME records in Route 53.
Once validated, ACM automatically issued the certificate without downtime or manual intervention.

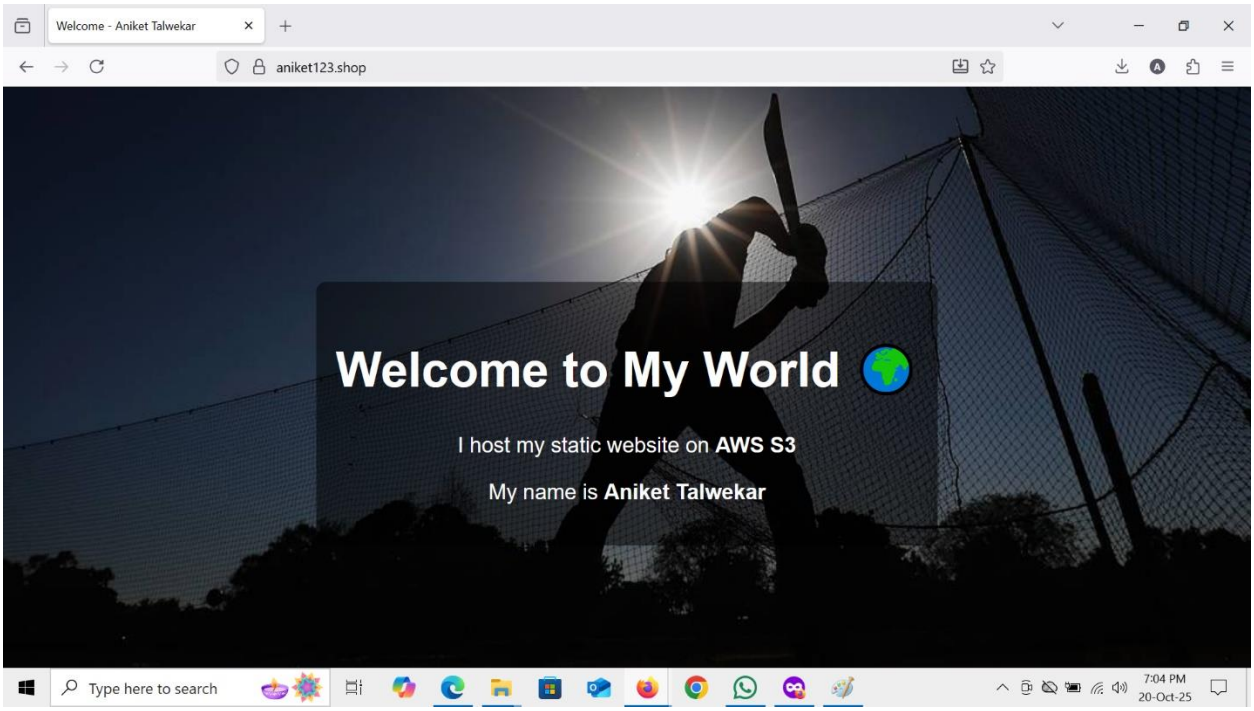
Step 4 – Route 53 and Domain Integration

Created a **hosted zone** for aniket123.shop in Route 53.
Added an **A (Alias)** record pointing to the CloudFront distribution, and a **CNAME** for www**.**
In GoDaddy, updated the domain's nameservers to the four Route 53 nameservers.
After propagation (30 minutes), the domain began resolving through AWS DNS.

Step 5 – Testing and Validation

Tested both URLs:

https://aniket123.shop
https://www.aniket123.shop



Both responded successfully with valid HTTPS, served from CloudFront edge locations.
Also verified the SSL certificate chain and TTL resolution using dig and nslookup tools.

Troubleshooting

During setup, handled:

- **AccessDenied** errors due to missing bucket policy
- **504 Gateway Timeout** during CloudFront propagation
- **DNS delays** while waiting for global resolution
  All were fixed through IAM review, cache invalidation, and propagation checks.

7. Cost Overview

Service	Description	Monthly Cost
S3	Static website hosting	$0.02 per GB
CloudFront	CDN delivery (Free Tier up to 1TB)	Free
Route 53	Hosted zone and DNS management	$0.50/month
ACM	SSL/TLS certificate	Free
GoDaddy	Domain name	₹499/year (~$6)

Total Estimated Cost: Less than $1/month (AWS) + yearly domain renewal.

This design follows the AWS **Free Tier** model and is suitable for personal or small-scale business websites.

8. Conclusion

This project was a good example of how different AWS services can be combined to create a complete, production-ready static website setup.
I used my AWS experience to design this solution in a simple yet scalable way — where each component has a clear role.

Using **S3** for static hosting and **CloudFront** for distribution gave me a fast and secure website with minimal cost.
Configuring **AWS Certificate Manager** and integrating it with CloudFront ensured full HTTPS support without any manual certificate renewal.
With **Route 53**, I managed the DNS routing efficiently, and by connecting it with **GoDaddy**, I made sure the domain resolution was clean and stable.

I also focused on practical troubleshooting — handling issues like access permissions in S3, CloudFront caching problems, and SSL validation delays.
These are the kind of real-world issues we face while working in AWS environments, so it felt like managing an actual production workload.

Overall, this project reflects how I approach cloud implementations — keeping things simple, cost-effective, and secure, following AWS best practices.
It's a small project, but it shows the complete flow from **infrastructure setup to deployment and optimization**, something I often do in real projects.