Project Report:

Scalable Static Website Deployment using AWS S3, Cloudflare, and GitHub Actions

---

1. Introduction

With the growing need for fast, scalable, and secure websites, deploying static sites using cloud infrastructure has become increasingly popular. This project demonstrates the end-to-end setup of a scalable static website hosted on **AWS S3**, secured and accelerated using **Cloudflare**, and automatically deployed using **GitHub Actions** for CI/CD. The project ensures global accessibility, reduced latency, and automated deployment with minimal manual intervention.

---

2. Abstract

This project aims to build and deploy a static website using a cost-effective and scalable architecture. By leveraging **Amazon S3** as the storage and hosting solution, **Cloudflare** for **DNS** (*Domain Name System*), **CDN** (*Content Delivery Network*) and **HTTPS** (*Hypertext Transfer Protocol Secure*) encryption, and **GitHub Actions** for automating deployment pipelines, the project provides a highly available and secure web application infrastructure. The static website is updated and deployed automatically upon every commit to the main branch, ensuring continuous integration and seamless content updates.

---

3. Tools Used

- **Amazon Web Services (AWS S3):** For static website hosting.
- **Cloudflare:** For DNS management, HTTPS (SSL/TLS), and CDN acceleration.
- **GitHub Actions:** For CI/CD automation.
- **GitHub:** For version control and code hosting.
- **HTML/CSS:** For creating the static website.
- **YAML:** For writing the GitHub Actions workflow file.

---

4. Steps Involved in Building the Project

Step 1: Create and Configure AWS S3 Bucket

- Create a new S3 bucket.
- Enabled static website hosting and specify index.html as the default document.
- Configure the bucket policy to allow public read access using a predefined JSON policy.

Step 2: Develop Static Website

- Design basic HTML (index.html) and optional CSS (styles.css) files.
- Validated layout and responsiveness locally before pushing to GitHub.

Step 3: Set Up GitHub Repository and Actions

- Create a GitHub repository and push the project files.
- Configure a GitHub Actions workflow (.yml file) to deploy the site to S3 upon each push to the main branch.
- Verify successful builds via the GitHub Actions dashboard.

Step 4: Configure Cloudflare for CDN (Content Delivery Network) & **HTTPS** (Hypertext Transfer Protocol Secure)

- Add the custom domain to Cloudflare.
- Update DNS records (CNAME) to point to the S3 website endpoint.
- Enable SSL under Cloudflare's SSL/TLS tab and set it to "Full" or "Full (Strict)" to support HTTPS.

---

5. Conclusion

This project successfully demonstrates how to host a static website that is **highly available, globally distributed, secure**, and **automatically deploy** with every code change. Using a combination of AWS S3, Cloudflare, and GitHub Actions provides a powerful, scalable, and low-maintenance solution for static web hosting. This approach ensures faster content delivery, seamless updates, and secure connections — ideal for personal projects, portfolios, and lightweight web applications.

GitHub Repository Link: Click-Here
Live Website: Click-Here

SUBMITTED BY

K. PRANAY SESA SAI