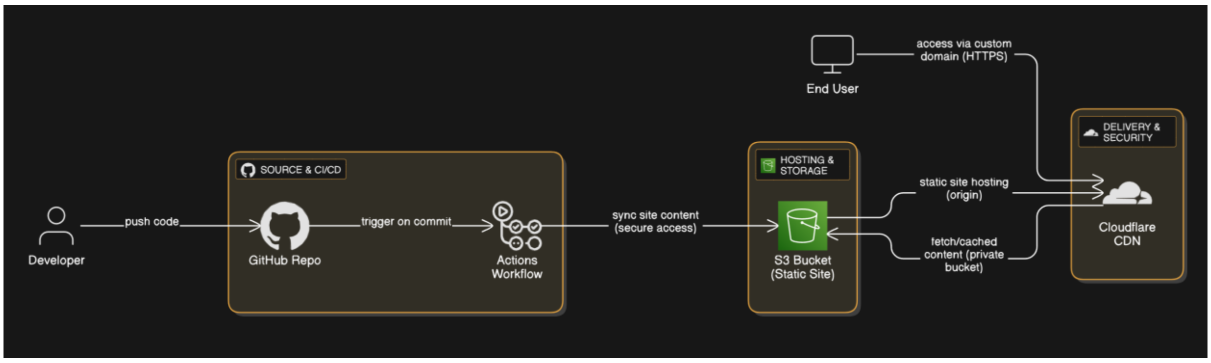
Scalable Static Website with S3 + CloudFront + GitHub Actions

Workflow:



1. Objective

- The primary goal of this project is to host a static website with scalability, global distribution, and HTTPS support using free-tier services. The website will automatically deploy whenever changes are pushed to GitHub.

Key objectives:

- Host a static HTML/CSS website on AWS S3.
- Enable a global CDN and HTTPS via CloudFront.
- Automate deployments using GitHub Actions CI/CD workflow.
- Optimize caching, versioning, and scalability.

2. Tools and Technologies

- AWS S3 (Free Tier) - Hosting static website files (HTML, CSS, JS, images).
- CloudFront (Free) - Global CDN, SSL certificate, and DNS management.

- GitHub Actions - Automate CI/CD deployment pipeline.
- HTML/CSS - Website front-end content.

3. System Architecture

GitHub Repository => GitHub Actions (CI/CD) => AWS S3 Bucket => CloudFront CDN & DNS => End Users (HTTPS)

Explanation:
1. Developers push code to GitHub.
2. GitHub Actions workflow triggers on `main` branch push.
3. Workflow syncs website files to S3
4. CloudFront serves the website globally via CDN and HTTPS.

4. Implementation Steps

Step 1: Create Static Website
Developed HTML, CSS, JAVASCRIPT and image assets.
Tested locally to verify responsiveness and styling.

Step 2: Set up AWS S3 Bucket and IAM Policies
Created a public S3 bucket with static website hosting enabled.
Configured bucket policy for public read access.
Enabled Bucket Versioning to manage updates.

Create a IAM Policies for Github Access to S3 Bucket

Step 3: Configure CloudFront
Added custom domain to CloudFront.
Configured DNS CNAME to point to S3 bucket.
Enabled SSL/TLS (Full/Strict) and caching rules.

Step 4: Create GitHub Actions Workflow
Workflow triggers on push to `main`.
Uses aws-actions/configure-aws-credentials for authentication.
Syncs only the website folder to S3.

Step 5: Test Deployment
Committed changes to GitHub.
GitHub Actions automatically deployed files to S3.
Accessed website via CloudFront domain (HTTPS).

5. Challenges and Solutions

AccessDenied errors: Corrected bucket policy.
Bucket ACL issues: Removed --acl public-read.
CloudFront caching delays: Enabled cache-busting with versioning.
HTTPS not working: Enabled SSL/TLS Full mode and updated DNS settings.

Requirements:

- Github Repo
- Github Workflows
- Website Files
- S3 Bucket with Policies
- IAM User and Policies
- CloudFront