

# AWS Secure Static Website Hosting

August 7, 2025

## Project Overview

This project demonstrates hosting a secure static website on AWS using:

- Amazon S3 (Private) + CloudFront (OAI) + AWS WAF + CloudWatch Logs + SNS + CloudWatch Alarm + Cost Explorer & Budgets

## Architecture Diagram

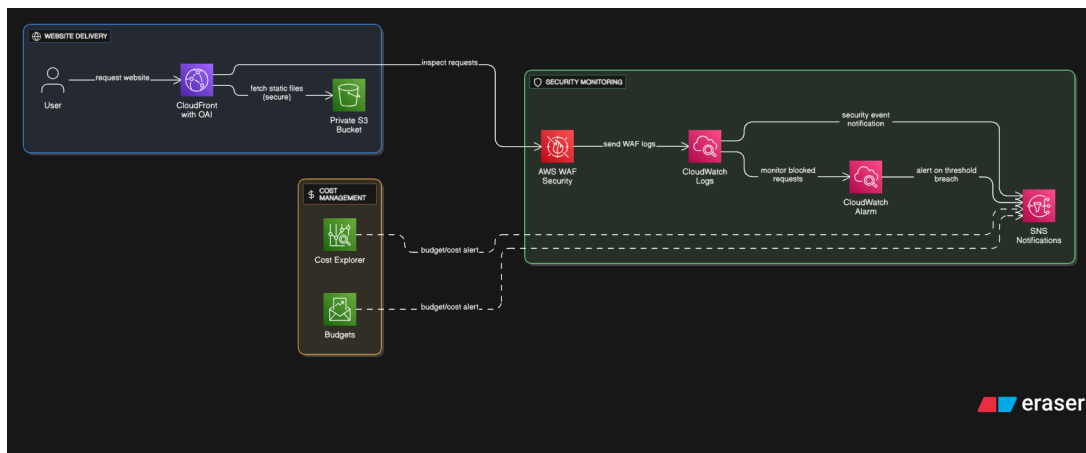


Figure 1: AWS Secure Static Website Hosting Architecture

## Implementation Steps

### Step 1: Create a Private S3 Bucket

1. Navigate to AWS Console → S3 → Create bucket
2. Bucket name: `awsfirst-project`
3. Block Public Access: Enable all options
4. Create bucket
5. Upload Website Files:
  - Upload `index.html`, CSS, JS, images
  - Files remain private (no public access)

### Step 2: Create CloudFront Distribution with OAI

1. Go to CloudFront → Create distribution
2. Origin domain: Choose your S3 bucket
3. Origin access:
  - Select Origin Access Identity (OAI)
  - Create new OAI → Update bucket policy automatically
4. Viewer protocol policy: Redirect HTTP to HTTPS
5. Default root object: `index.html`
6. Create distribution and wait until Deployed

**Test:** Visit your CloudFront domain (e.g., `https://dXXXX.cloudfront.net`). You should see your `index.html`.

### Step 3: Attach AWS WAF to CloudFront

1. Go to WAF & Shield → Web ACLs → Create web ACL
2. Region: Global (CloudFront)
3. Resources to protect: Select your CloudFront distribution
4. Add AWS Managed Rules:
  - `AWSManagedRulesCommonRuleSet` (common attacks)
  - `AWSManagedRulesKnownBadInputsRuleSet` (SQL Injection, XSS)
5. Create Web ACL

#### Step 4: Enable CloudWatch Logs for WAF

1. Navigate to WAF → Your Web ACL → Logging and metrics tab
2. Click Enable logging → CloudWatch Logs
3. Create log group: `aws-waf-logs-webacl-first`
4. Enable Store full logs

#### Step 5: Set Up Amazon SNS for Notifications

1. Go to Amazon SNS → Create topic
  - Type: Standard
  - Name: WAFNotifications
2. Create subscription → Protocol: Email → Add your email
3. Check your inbox and Confirm subscription

## Problem Resolution

### Warning

#### Issue 1: CloudFront did not serve the index.html file

**Symptoms:** Website URL showed Access Denied or blank page

**Cause:** Missing Default Root Object configuration

**Solution:** Updated CloudFront Distribution → Default Root Object to `index.html`

### Warning

#### Issue 2: CloudWatch Logs did not capture WAF logs

**Symptoms:** No logs appeared in CloudWatch initially

**Cause:** Incorrect log group naming format

**Solution:** Created log group with format `aws-waf-logs-<name>`

## Key Takeaways

- Always configure the **Default Root Object** in CloudFront
- Use the correct log group naming format (`aws-waf-logs-<name>`) for WAF
- Monitor costs with AWS Budgets to avoid unexpected charges

## Project Output

- Secure Static Website URL via CloudFront (HTTPS)
- WAF Protection with blocked requests (403 Forbidden)
- CloudWatch Logs with request details
- SNS Notifications for alerts
- CloudWatch Alarm for blocked traffic spikes
- Cost Management with AWS Budgets and Cost Explorer