

# SECURE STATIC WEBSITE DEPLOYMENT ON AWS WITH S3, CLOUDFRONT, AND ROUTE 53

Proof-of-Concept Documentation (DO NOT use as-is in production)

**Collaborators:** Nicolas Portilla Gomez | Joshua Shapiro | Andrew Enright

**Version:** 1.0 | **Date:** 07 08 2025

## Table of Contents

- 1. Introduction ..... 2
- 2. Architecture Overview ..... 2
- 3. Prerequisites ..... 2
- 4. Implementation Steps..... 2
  - 4.1 Create S3 Bucket ..... 2
  - 4.2 Upload Website Files..... 3
  - 4.3 Configure CloudFront..... 3
  - 4.4 Route 53 DNS & HTTPS ..... 3
- 5. Cost & Security Considerations..... 4
- 6. Outcome & Verification..... 4
- 7. Future Enhancements ..... 4
- 8. Appendix ..... 4
  - Appendix A - Route 53 Hosted Zone Configuration for inter.jimwest405.com ..... 4
  - Appendix B – Public-read bucket policy..... 5
  - Appendix C - Verification of website function..... 5
  - Appendix D - Pending Validation in AWS Certificate Manager ..... 6
  - Appendix E - Creating DNS Validation Records in Route 53..... 6
  - Appendix F - Adding CNAME Validation Records in Cloudflare ..... 6
  - Appendix G - Certificate Successfully Issued..... 6

# 1. INTRODUCTION

This documentation outlines the process of hosting a static website using AWS S3 for storage and CloudFront for content delivery. AWS was chosen for its scalability, global reach, and integration with other cloud services; readers will learn how to securely deploy and distribute a static site using these tools.

## 2. ARCHITECTURE OVERVIEW

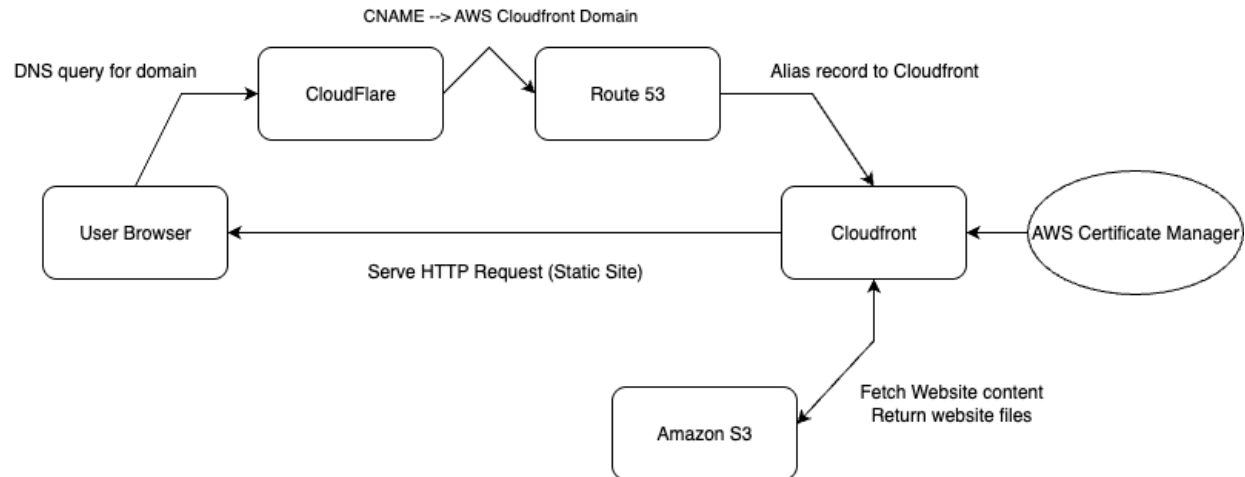


Figure 1: Architecture Diagram

## 3. PREREQUISITES

Item	Notes
AWS account (Free Tier)	IAM user with S3/CloudFront/Route 53 permissions
HTML/CSS site	At minimum <b>index.html</b>
Optional custom domain	Registered in Route 53 or external registrar (e.g. Cloudflare, GoDaddy, etc)

## 4. IMPLEMENTATION STEPS

### 4.1 Create S3 Bucket

1. Navigate to S3 -> Create bucket.
2. Bucket name: static-site-project-trevenx-interns
3. Disable "Block all public access".

#### 4. Enable static website hosting -> index document = index.html.

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ **Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ **Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Figure 2: Bucket Public Access Settings

## 4.2 Upload Website Files

1. Click Upload -> add index.html and assets.
2. Verify public read via bucket policy (See Appendix B)

## 4.3 Configure CloudFront

1. Create distribution -> Origin = S3 website endpoint.
2. Default root object = index.html.
3. Viewer protocol policy = "Redirect HTTP -> HTTPS".

**static-website-project** Standard View metrics

[General](#) | [Security](#) | [Origins](#) | [Behaviors](#) | [Error pages](#) | [Invalidations](#) | [Tags](#) | [Logging](#)

**Details**

<b>Name</b> static-website-project <a href="#">↗</a>	<b>Distribution domain name</b> <a href="#">↗</a> d3hh6nea5y7w9x.cloudfront.net	<b>ARN</b> <a href="#">↗</a> arn:aws:cloudfront::748732839:080:distribution/E2A63DPJWC6FL	<b>Last modified</b> <a href="#">↗</a> Deploying
---	--	--	---

**Settings** Edit

<b>Description</b> -	<b>Alternate domain names</b> - <a href="#">Add domain</a>	<b>Standard logging</b> <a href="#">Off</a>
<b>Price class</b> Use all edge locations (best performance)		<b>Cookie logging</b> <a href="#">Off</a>
<b>Supported HTTP versions</b> HTTP/2, HTTP/1.1, HTTP/1.0		<b>Default root object</b> -

Figure 3: CloudFront Distribution Settings

## 4.4 Route 53 DNS & HTTPS

1. Request TLS cert in ACM (us-east-1) for intern.jimwest405.com
2. Validate via DNS CNAME
3. Edit CloudFront distribution -> Alternate domain names = inter.jimwest405.com -> attach cert
4. In route 53 hosted zone create a record (alias) to the CloudFront distribution.

## 5. COST & SECURITY CONSIDERATIONS

Service	Free tier?	Monthly cost (low traffic)
S3 storage	Yes, 5 GB	< \$0.05
CloudFront	1 TB out	\$0-\$2
Route 53 hosted zone	N/A	\$0.50

- Enable S3 versioning + lifecycle rules for potential rollbacks.
- Consider AWS WAF for production sites.

## 6. OUTCOME & VERIFICATION

- Browsing to <https://intern.jimwest405.com/> returns expected content. (See Appendix C)
- SSL padlock present (tested via <https://www.ssllabs.com/ssltest/>).

## 7. FUTURE ENHANCEMENTS

- CI/CD with GitHub Actions to auto-sync S3.
- Add CloudFront Functions for HTTP security headers.
- Configure S3 access logs to Athena/OpenSearch for analysis.

## 8. APPENDIX

### Appendix A - Route 53 Hosted Zone Configuration for intern.jimwest405.com

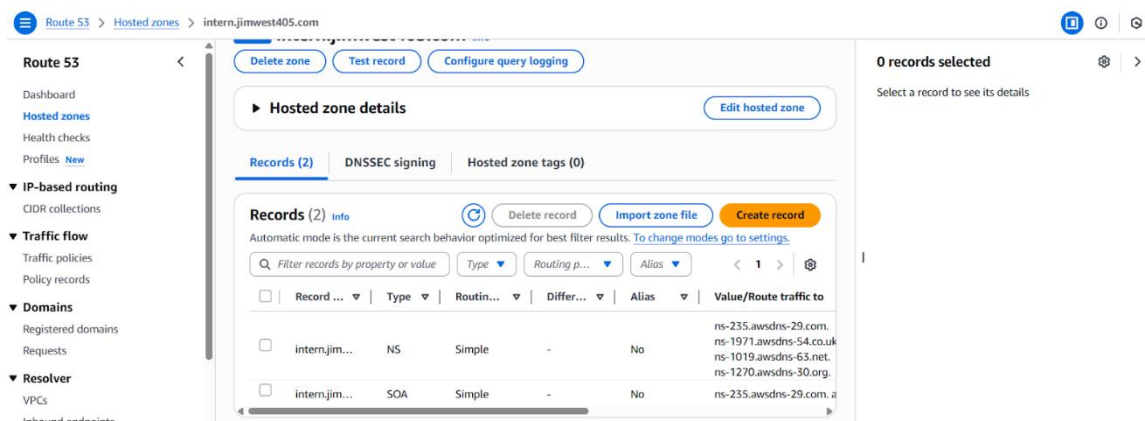


Figure 4: Route 53 DNS Records

## Appendix B – Public-read bucket policy

### Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Edit](#)[Delete](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontOrPublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::static-site-project-trevenx-interns/*"
    }
  ]
}
```

[Copy](#)

Figure 5: Bucket Policy Configuration

## Appendix C - Verification of website function

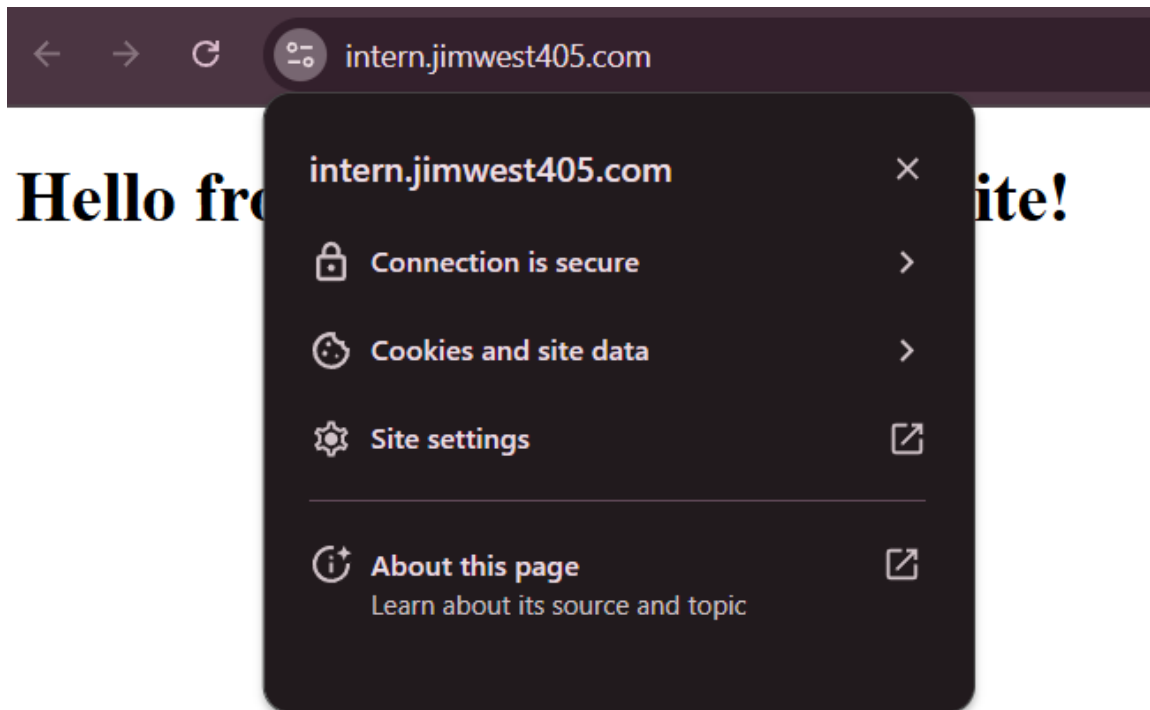


Figure 6: Secure HTTPS Connection Confirmation

## Appendix D - Pending Validation in AWS Certificate Manager

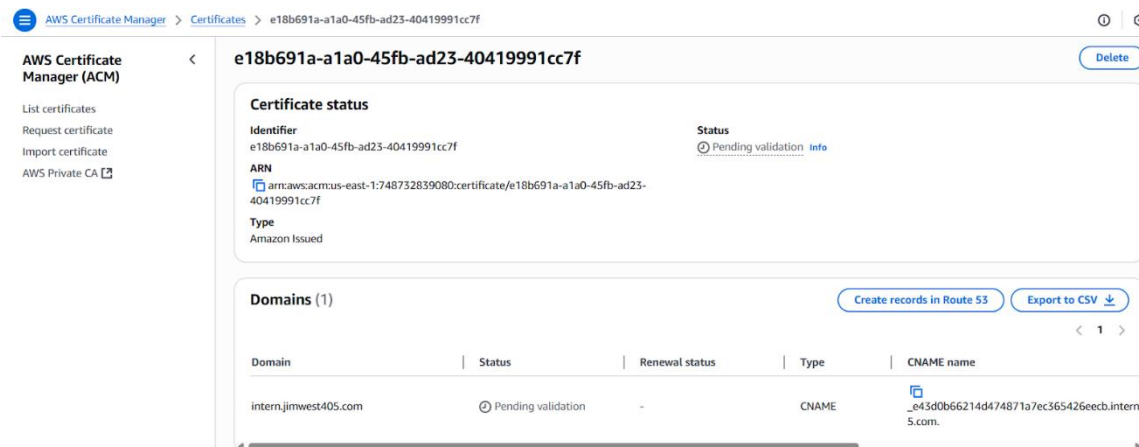


Figure 7: Certificate Creation in AWS Certificate Manager (ACM)

## Appendix E - Creating DNS Validation Records in Route 53

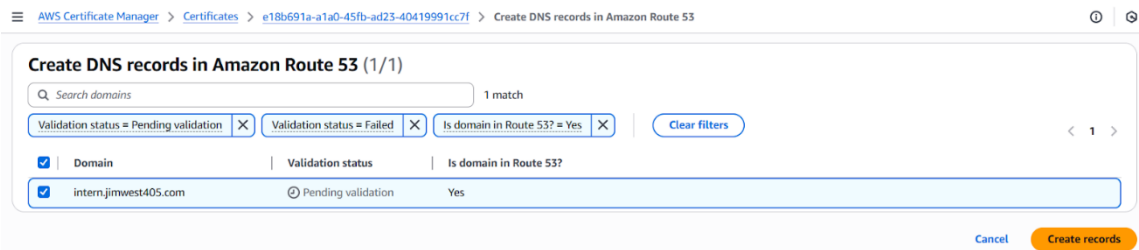


Figure 8: DNS Record Creation in Route 53

## Appendix F - Adding CNAME Validation Records in Cloudflare



Figure 9: Cloudflare DNS CNAME Record Configuration

## Appendix G - Certificate Successfully Issued

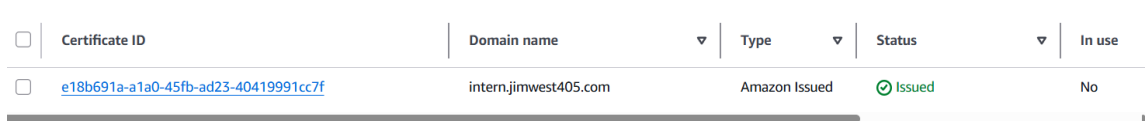


Figure 10: Certificate Successfully Issued in ACM