

# Solution

## content

---

- Architecture overview
- S3 Bucket (Static Website Storage)
- CloudFront (content delivery network)
- ACM certificate (HTTPS)
- Route 53 (DNS configuration)
- Cross-Account Setup (Important Detail)
- Final Result

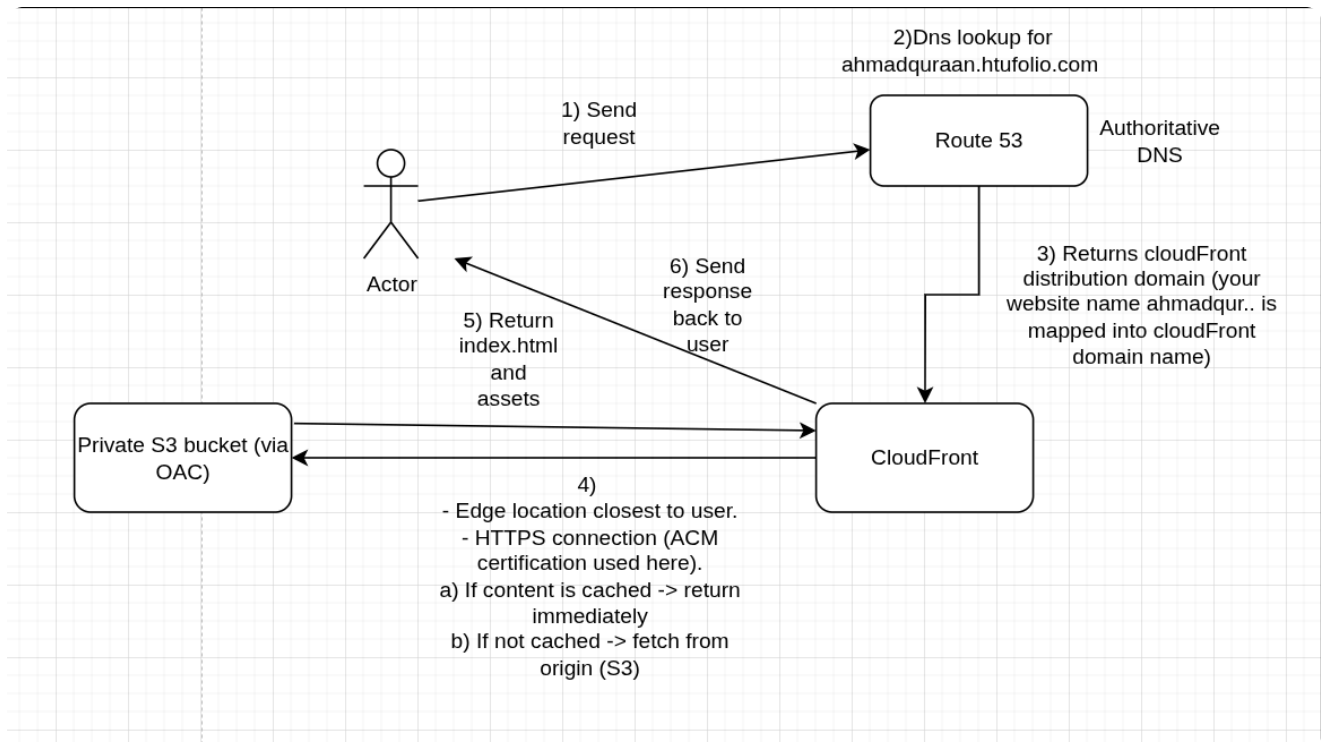
 First thing make your portfolio, the index.html, css, etc.....

## Architecture overview

---

```
User (Browser)
  ↓ HTTPS
Custom Domain (ahmadquraan.htufolio.com)
  ↓ Route 53 (DNS)
CloudFront Distribution
  ↓ Origin Access Control (OAC)
Private S3 Bucket
  ↓
index.html + static assets
```

CSS



- Key points:
  - The S3 bucket is **private**
  - All traffic goes through **CloudFront**
  - HTTPS is enabled using **ACM**
  - DNS is managed using **Route 53**



## S3 Bucket (Static Website Storage)

1. Make a S3 bucket.
2. Upload the content on it `index.html` ,...
3. Ensured `index.html` is located at the **root of the bucket**
4. Ensure it's private (don't uncheck the default)



## CloudFront (content delivery network)

CloudFront provides performance, security, and HTTPS support.

- Create a new distribution and in Origin --> put the S3 domain name.
- Origin path --> write "index.html" --> **Default root object:** `index.html`
- Leave every setting as it is .
- OAC (Origin access control) --> Just allow the cloudFront to access the S3 bucket, public access is blocked.



## ACM certificate (HTTPS)

---

### Sub-content

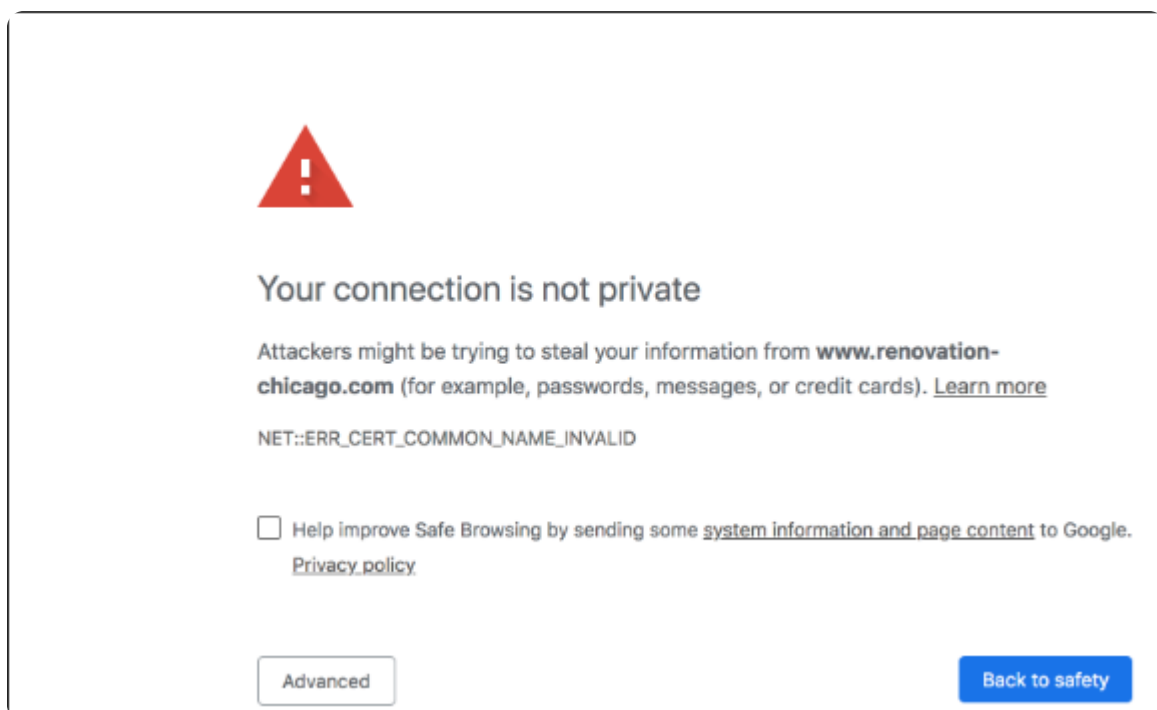
---

- Why making ACM certificate?
- How to make ACM certificate ?
- How ACM and CloudFront integrate ?

### Why making ACM certificate?

---

- Now if we tried to put the record (cloud front with ahmadquraan.htufolio) in the hosted zone in Route 53, we will face a big issue which is that our domain is not secure (HTTP), and such window will appear.

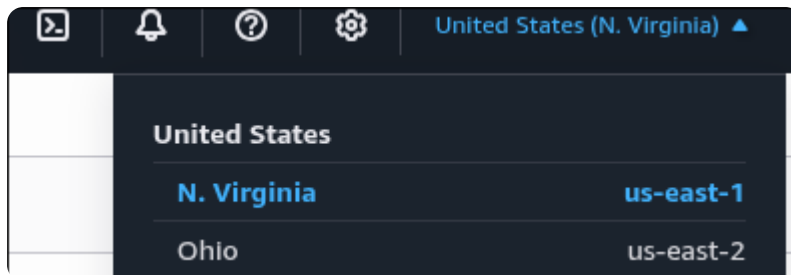


- and to make it secure we need to make a certification using **AWS Certificate manager**.

# How to make ACM certificate ?

```
search certificate manager --> request # Make sure you are in us-east-1
```

- Certificate was created in **us-east-1** (required for CloudFront). --> doesn't accept any other certificate.



- After that:

**Request public certificate**

**Domain names**  
Provide one or more domain names for your certificate.

**Fully qualified domain name** [Info](#)

[Add another name to this certificate](#)

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

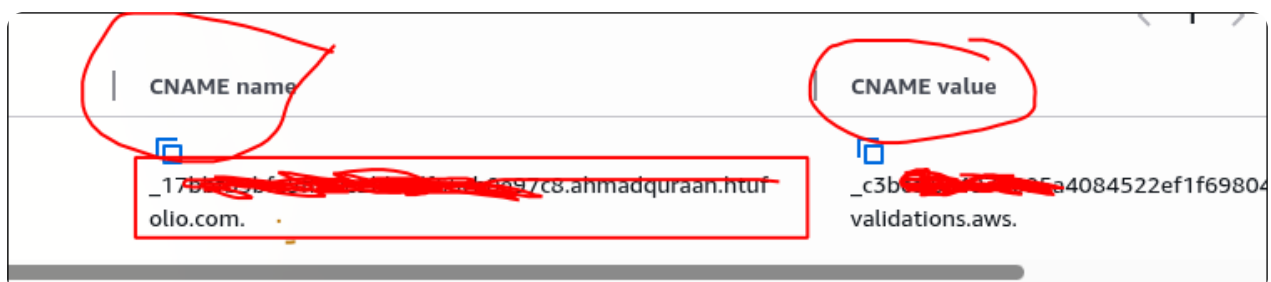
**Allow export** [Info](#)

☒ **Disable export**  
Use this certificate only with integrated AWS services. The private key for this certificate will be disallowed for exporting from AWS.

☐ **Enable export**  
Export this certificate and private key for use with any TLS workflow. ACM will charge your account based on the number of times you export the certificate when the certificate is issued for the first time and for each renewal.

**Validation method** [Info](#)  
Select a method for validating domain ownership.

- In the domain name just put your domain name (the one you wanna make a record on the hosted zone) ex: **ahmadquraan.htufolio.com**.
- Now two values will appear in the certification, CNAME name and value.



- Now the certification will remain under "validation" until you make a specific DNS record for it in the hosted zone using those values--> **Create this CNAME record with this exact value**. The ACM record **Must be created in the SAME hosted zone as the domain** you're requesting the certificate for.
- So go to the hosted zone (Yazan IAM account) then make a new record there with the **CNAME name as it's name and CNAME value as it's value**
  - The reason is **You need the ACM DNS record to prove that you own the domain**.
  - CloudFront will not trust or use a certificate unless ACM has verified that ownership.

- The key principle: *Who are you to claim this domain?* so because anybody can generate a certificate for domains they don't own, AWS will ask you to provide the DNS record to approve that you own it .
- Now after making the certification and the ACM record in the hosted zone, the certification will change it's status into "Issued".
- After that you need to get back to the cloudFront and give it the certification that you wanna use. [CloudFront -> General -> Edit](#)
  - The CNAME -> IS your domain name.
  - SSL certificate -> your certification.

**Alternate domain name (CNAME) - optional**  
Add the custom domain names that you use in URLs for the files served by this distribution.

ahmadquraan.htufolio.com

[Add item](#)

*i* To add a list of items, use the [bulk editor](#).

**Custom SSL certificate - optional**  
Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region

ahmadquraan.htufolio.com (540c6501-41f7-4fc2-8ce0-d19c8f9b6180)

☒ [ahmadquraan.htufolio.com](#) [Request certificate](#)

☐ Legacy clients support - \$600/month prorated charge applies. Most customers do not need this. CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your content over IPv4.

**Security policy**  
The security policy determines the SSL or TLS protocol and the specific ciphers that CloudFront uses for HTTP connections.

☒ TLSv1.3\_2025

☐ TLSv1.2\_2025

☐ TLSv1.2\_2021 (recommended)

## How ACM and CloudFront integrate ?

---

ACM (Certificate Authority role)

CloudFront (Web server / CDN role)

## ACM (Certificate Authority role)

---

- ACM's job is only to:
  - Verify domain ownership
  - Issue SSL/TLS certificates
  - Renew them automatically

- ACM **does not**:
  - Serve websites
  - Handle user traffic
  - Terminate HTTPS connections
- Once ACM issues the certificate, its job is basically done.

## CloudFront (Web server / CDN role)

---

- CloudFront's job is to:
  - Accept browser connections
  - Terminate HTTPS (TLS handshake)
  - Serve content from S3
- But CloudFront **cannot guess** which certificate you want.
- So you must tell it:

**!!** "When someone visits this domain, use THIS certificate."

- So CloudFront:
  - Stores the certificate
  - Uses it during TLS handshakes
  - Presents it to browsers
- For that we attach the cloudFront with the certificate after make it and verify it throw making the ACM record.




## Route 53 (DNS configuration)

---

- Now in **htufolio.com**, 2 records have been created:
  1. ACM record:
    - Type: CNAME
    - These are the values we should make the ACM record with:

```
CNAME name: _17bbfb5bfe5ff40e2bb9df65cb0e97c8.ahmadquraan.htufolio.com
CNAME value: _c3bccf0a4878085a4084522ef1f69804.jkddzztszm.acm-validations
```


Record name

\_17bbfb5bfe5ff40e2bb9df65cb0e

Record type

CNAME

Value

\_c3bccf0a4878085a4084522ef1f  
acm-validations.aws

Alias

No

TTL (seconds)

300

Routing policy

Simple

2. Website record (Main record):

- type: CNAME
- Name: ahmadquraan
- Value: dxxxxxxxxxxxxx.cloudfront.net

Record name [Info](#)

ahmadquraan .htufolio.com

Keep blank to create a record for the root domain.

Record type [Info](#)

CNAME – Routes traffic to another domain ... ▼

☐ Alias

Value [Info](#)

dy6vwh8t8hli9.cloudfront.net

Enter multiple values on separate lines.

*anyone searched for that domain will mapped to cloudfront*

## Cross-Account Setup (Important Detail)


---

- CloudFront, S3, and ACM are in [my personal AWS account](#).
- Route 53 hosted zone is in [Yazan AWS account](#).
- Because of this:
  - **CNAME** was used (not Alias).
  - Alias records only work when Route 53 and CloudFront are in the [same account](#).



## Final Result

---

- Website is accessible at:  
<https://ahmadquraan.htufolio.com>
- HTTPS works correctly (lock icon .
- S3 bucket is private.
- CloudFront serves the site securely and efficiently.
- DNS resolves correctly using Route 53.



## One-Line Conclusion

---

**!!** This project deploys a secure static portfolio on AWS using a private S3 bucket, CloudFront with Origin Access Control, ACM for HTTPS, and Route 53 for custom domain DNS management.