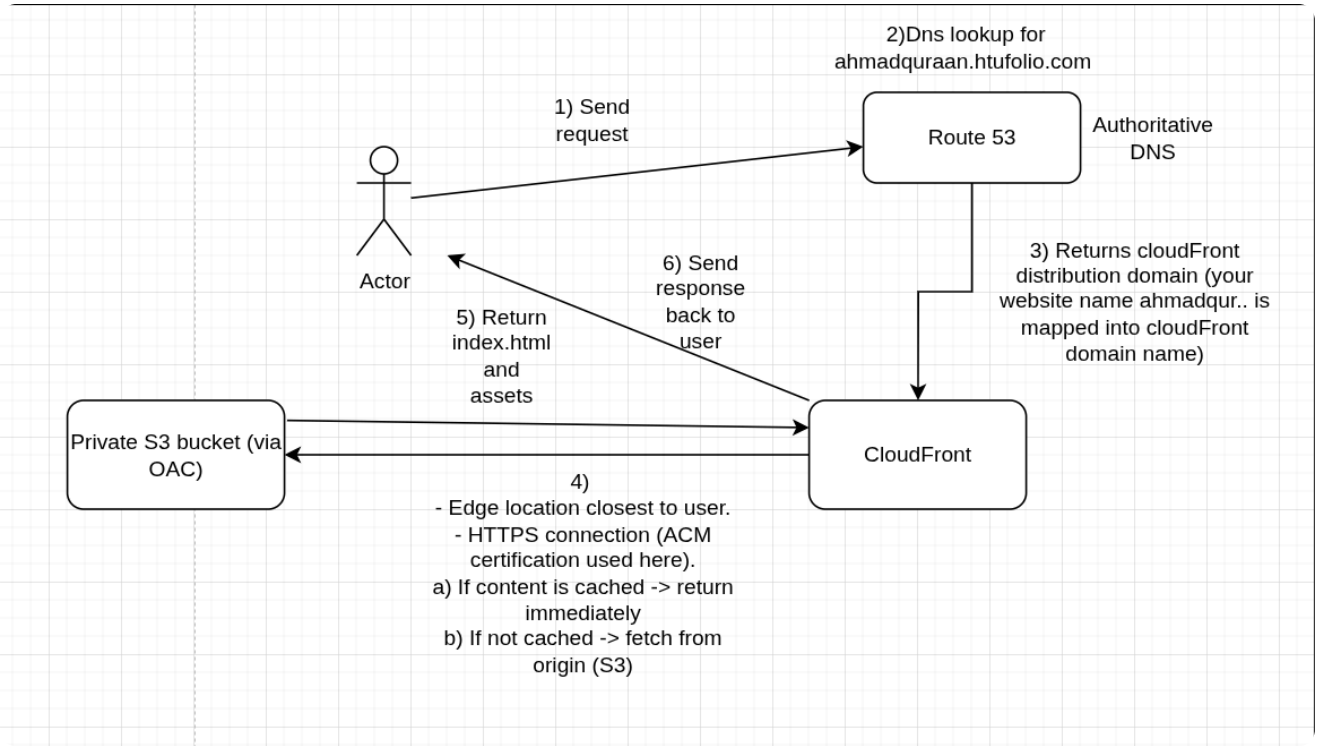
Solution

content

> **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
```



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

5

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)

5

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.

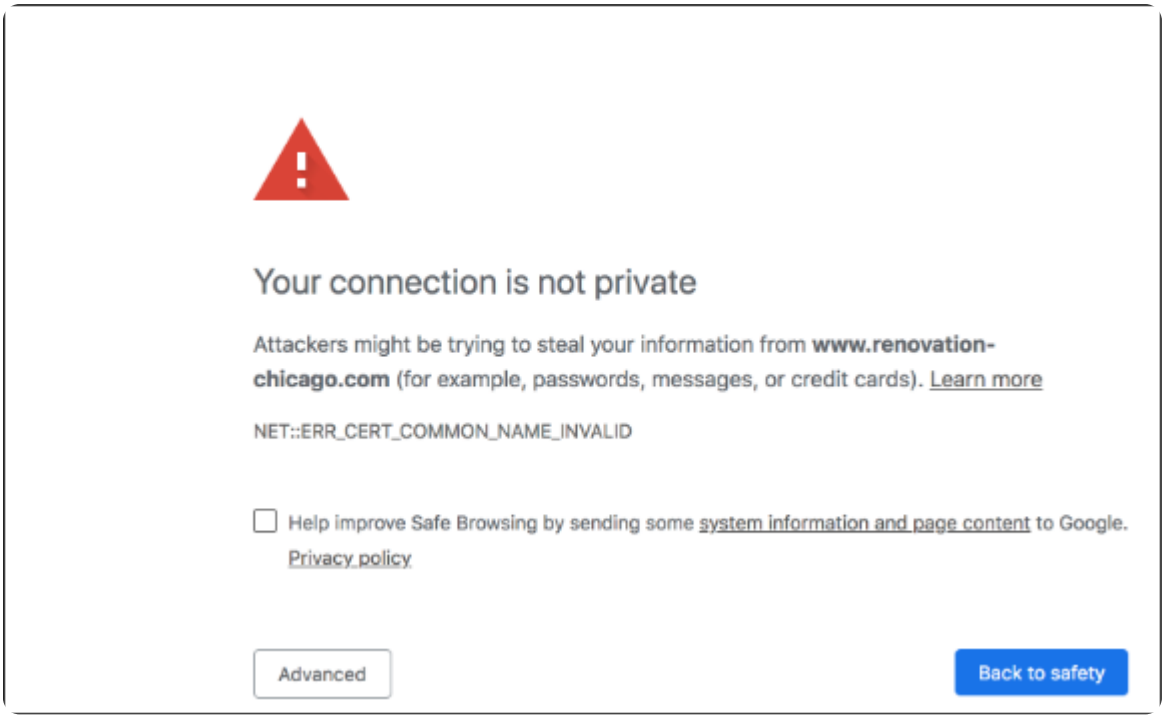
ACM certificate (HTTPS)

Sub-content

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

Why making ACM certificate?

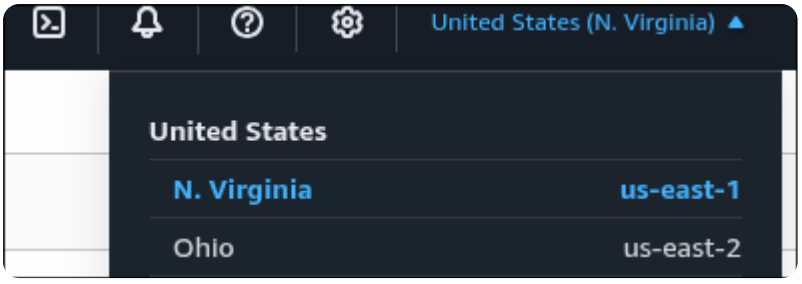
- Now if we tried to put the record (cloud front with ahmadquraan.htufolio) in the hosted zone in Route 53, we will face a big issue which is that our domain is not secure (HTTP), and such window will appear.

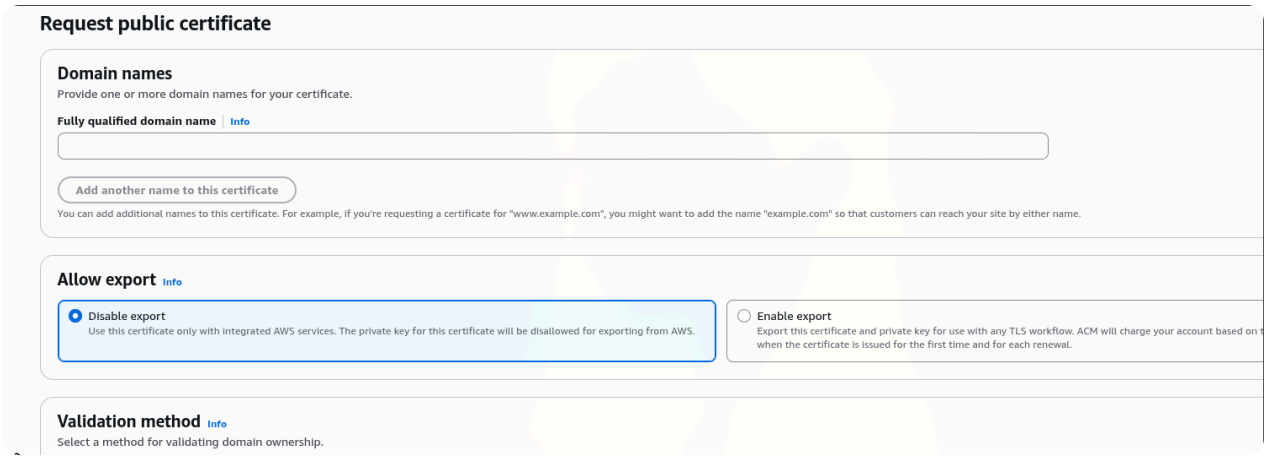


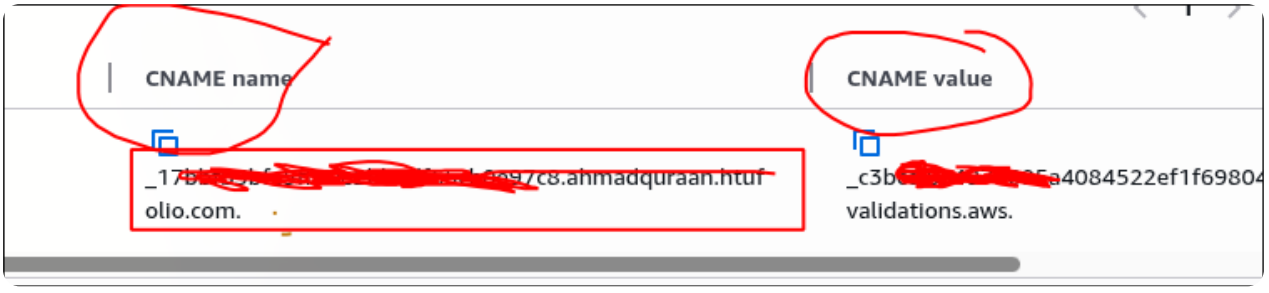
- and to make it secure we need to make a certification using AWS Certificate manager.

How to make ACM certificate ?

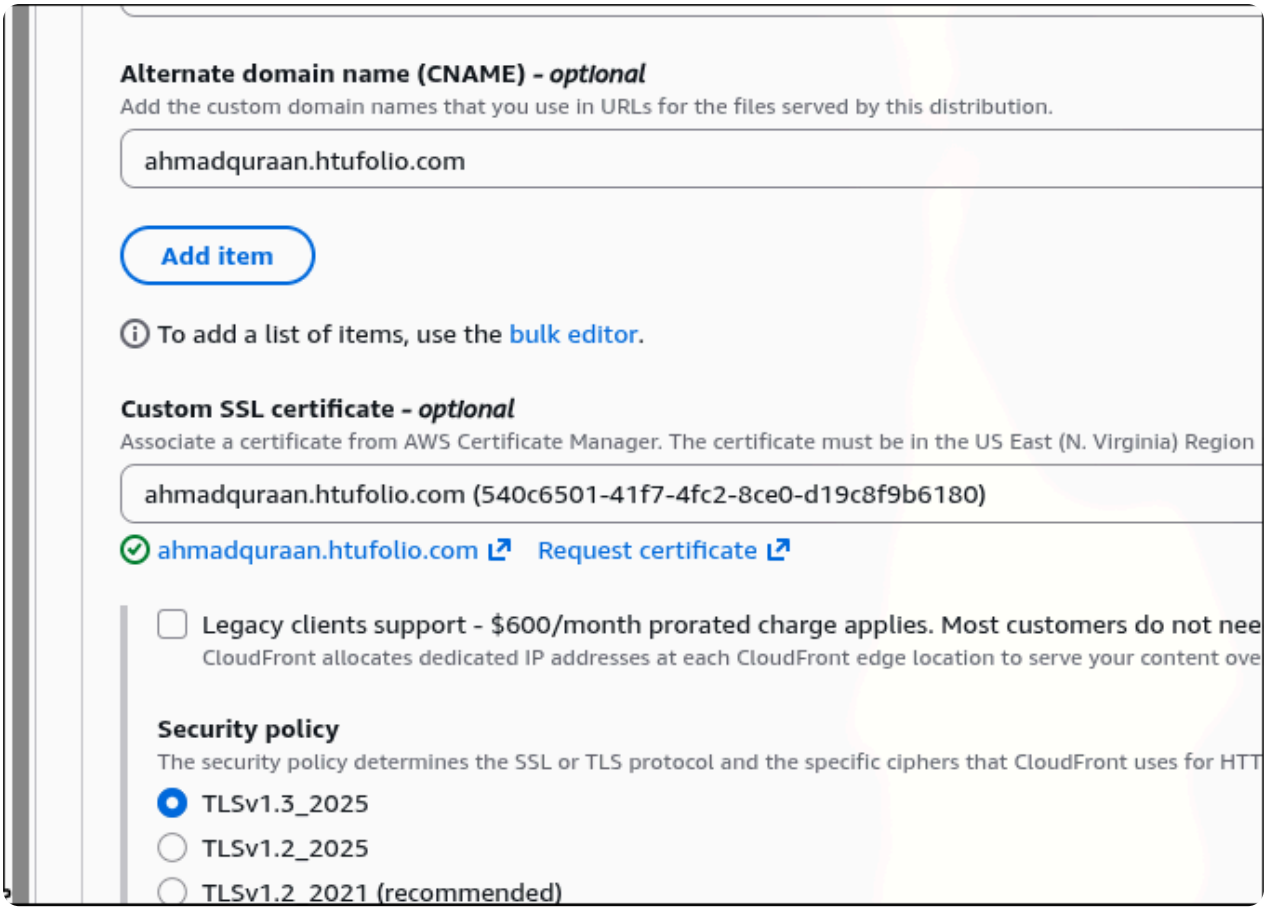
```
search certificate manager --> request # Make sure you are in us-east-1
```

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

- After that:
- 

- 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.
- 

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."

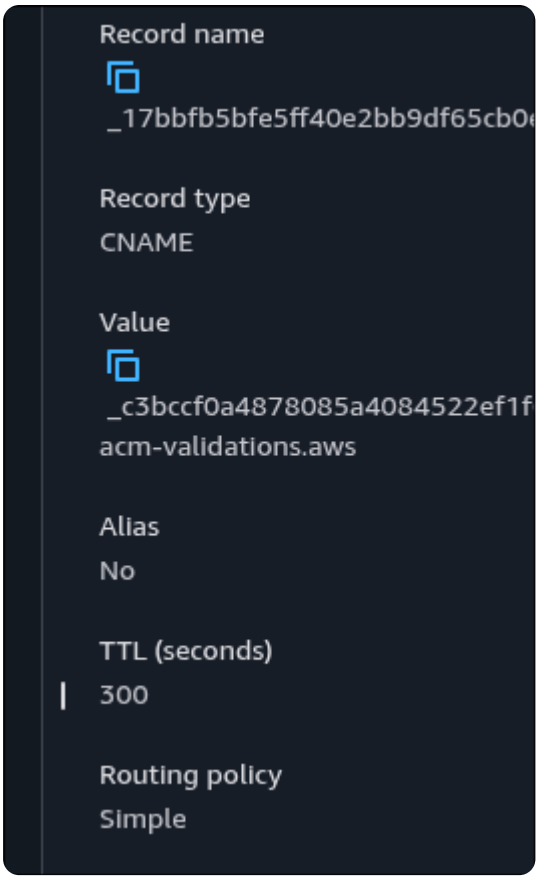
- So CloudFront:
  - Stores the certificate
  - Uses it during TLS handshakes
  - Presents it to browsers
- For that we attach the cloudFront with the certificate after make it and verify it throw making the ACM record.

Route 53 (DNS configuration)

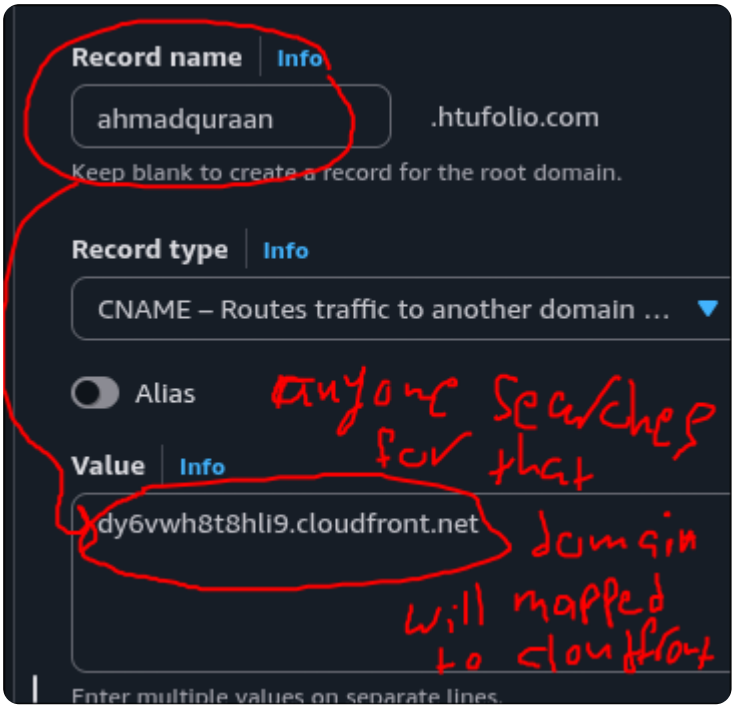
- Now in **htufolio.com**, 2 records have been created:

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
```



2. Website record (Main record):
    - type: CNAME
    - Name: ahmadquraan
    - Value: dxxxxxxxxxxxx.cloudfront.net



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**.

ω

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.

ω

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.