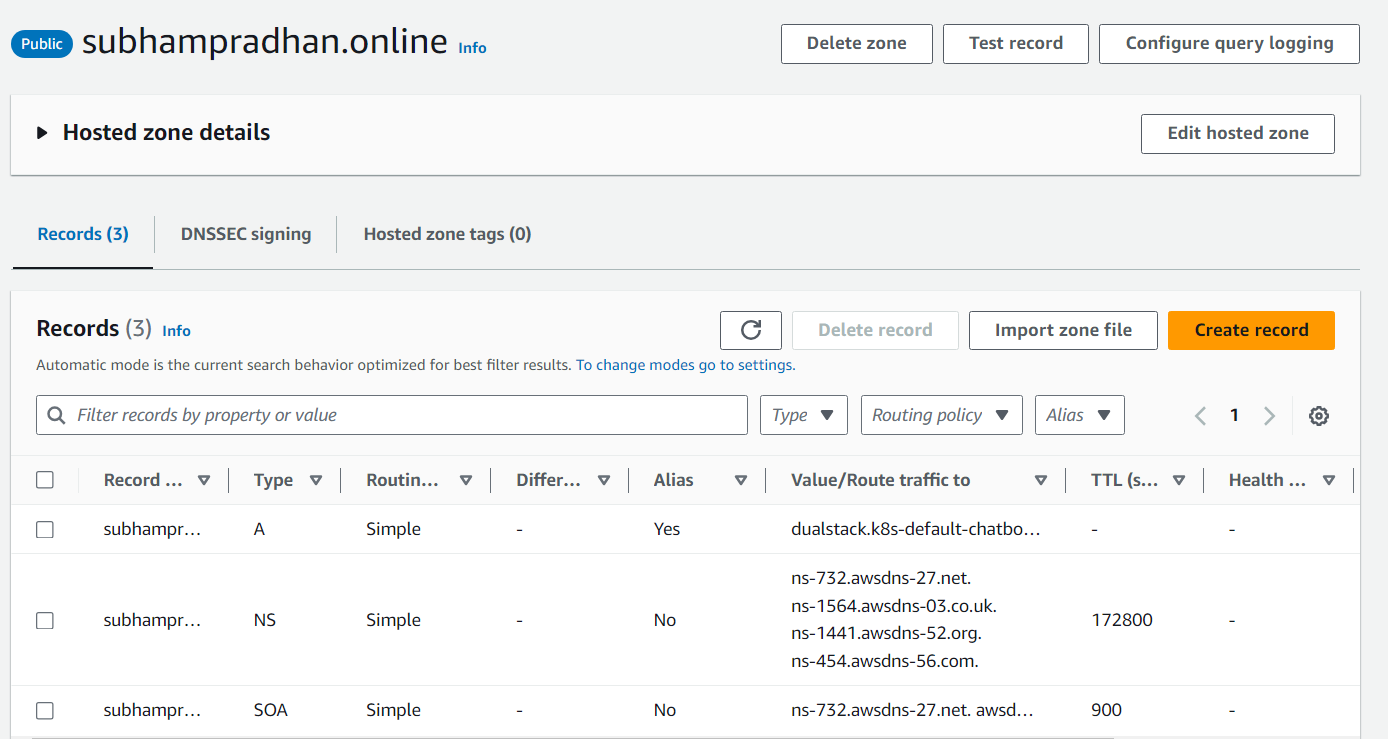
**Adding Domain name to the Application and SSL/TLS to it**

Adding a simple A record in the route 53 after creating a public hosted zone. 

Changing the ingress file in order to add the host.

apiVersion: apps/v1

kind: Deployment

metadata:

name: chatbot-deployment

labels:

app: django-app-deployment

spec:

replicas: 1

selector:

matchLabels:

app: django-app

template:

metadata:

labels:

app: django-app

spec:

containers:

- name: chatbot-cont

imagePullPolicy: Always

image: 991486635617.dkr.ecr.us-east-1.amazonaws.com/chatobott-img:latest

ports:

- containerPort: 8000

---

apiVersion: v1

kind: Service

metadata:

name: chatbot-service

spec:

selector:

app: django-app

ports:

- protocol: TCP

port: 80

targetPort: 8000

---

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: chatbot-ingress

annotations:

kubernetes.io/ingress.class: alb

alb.ingress.kubernetes.io/scheme: internet-facing

alb.ingress.kubernetes.io/target-type: ip

alb.ingress.kubernetes.io/healthcheck-protocol: HTTP

alb.ingress.kubernetes.io/healthcheck-path: "/"

spec:

rules:

- host: "subhampradhan.online"

http:

paths:

- path: /

pathType: Prefix

backend:

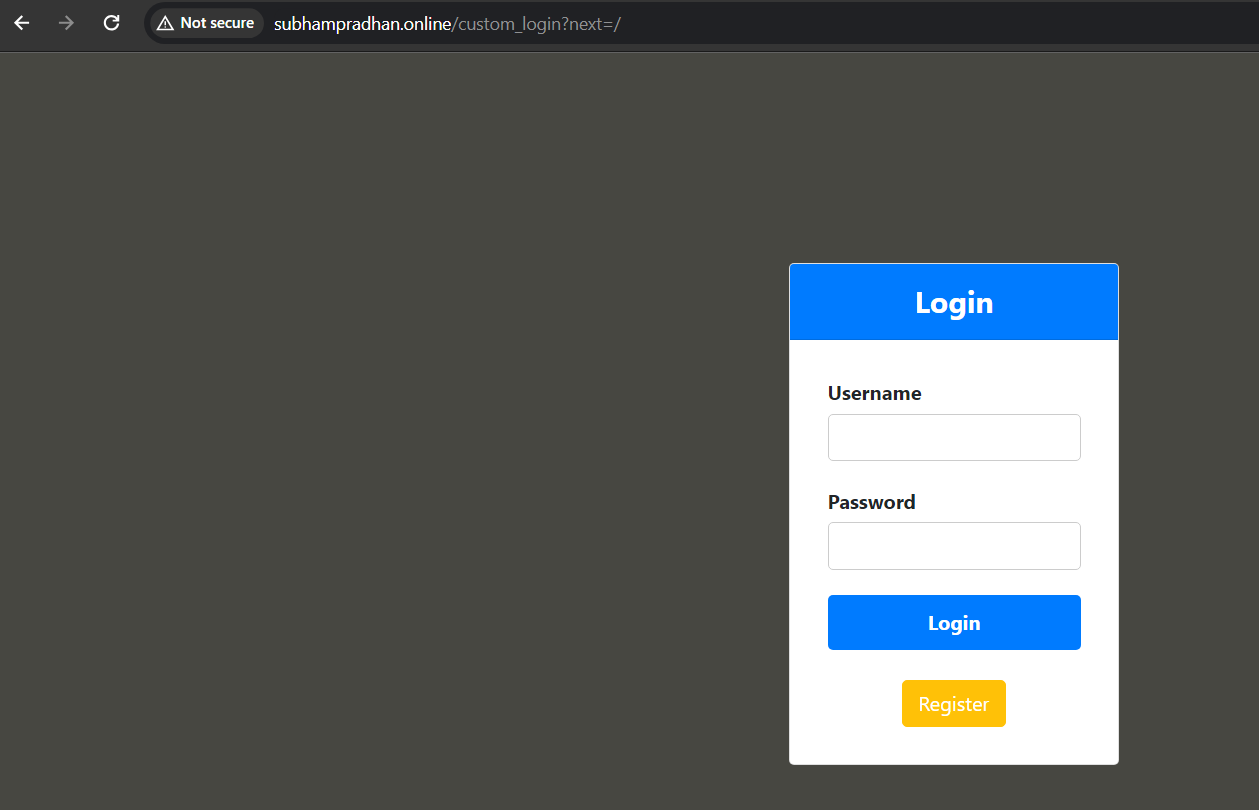
service:

name: chatbot-service

port:

number: 80

**After all these settings the domain started working fine.**

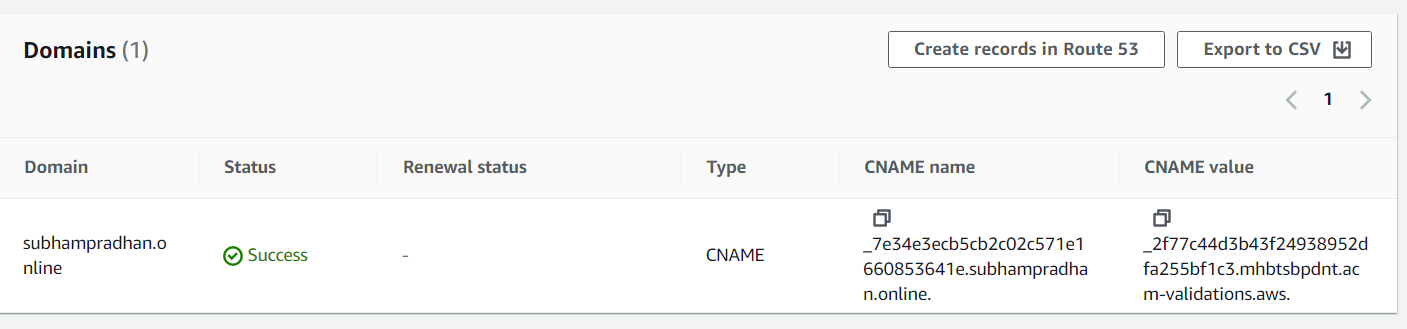
****

**And now it’s time to secure it.**

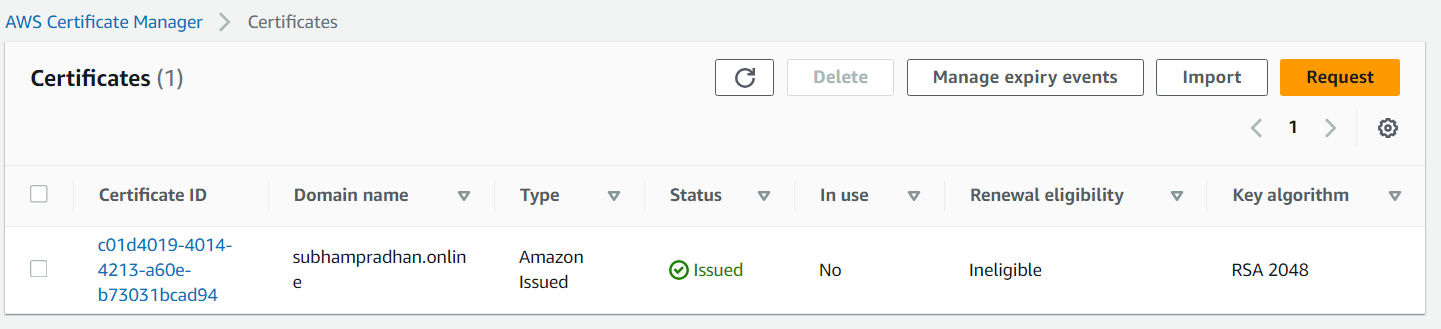
Getting a certificate issued from ACM (AWS Certificate Manager)

Requested the certificate via DNS record.

After requesting for the certificate added the CNAME record generated by the certificate issue request in the Route53 records.



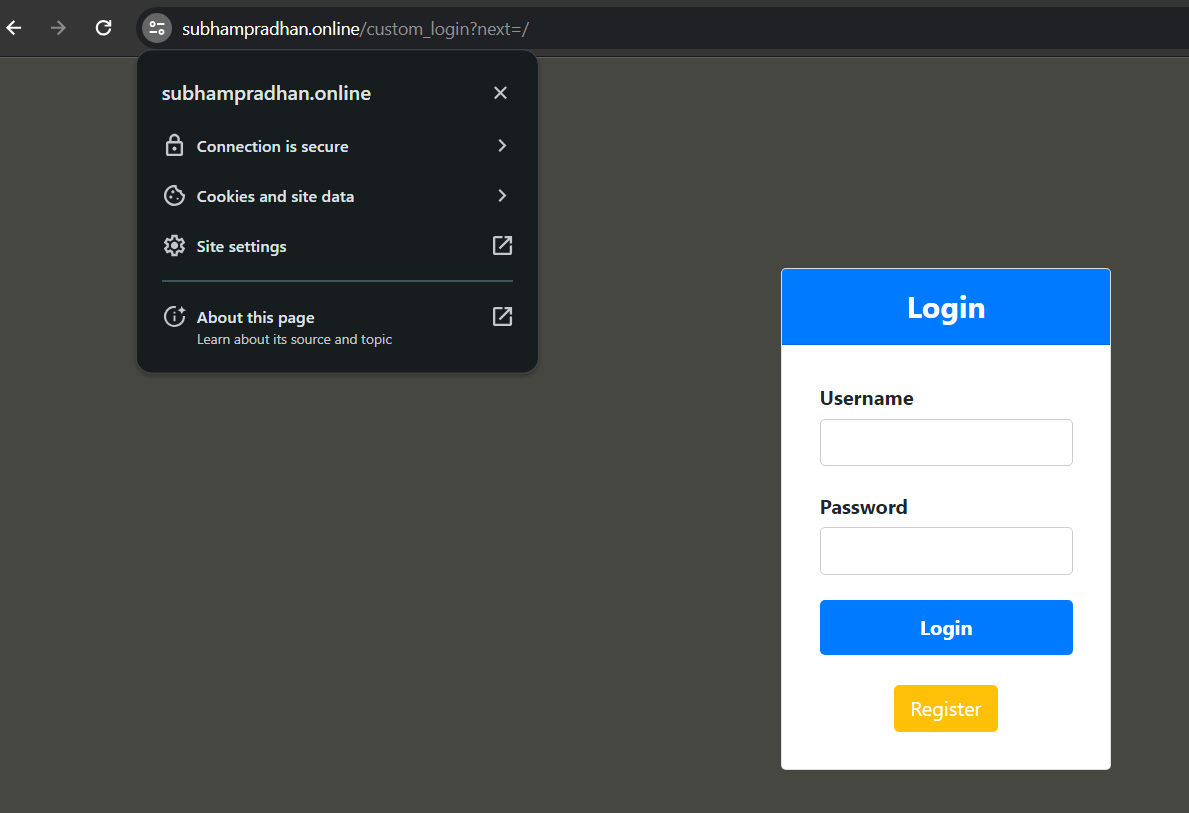
And now the certificate has been issued successfully.



Adding the ARN in the ingress file of the K8s application.



After doing the above changes in the ingress file and apply it, the application now works perfectly on HTTPS.



**And so, with this the SSL has been added and the domain names has been added to an application running inside pods of Kubernetes.**

**Note:** For redirecting the traffic from http to https…i.e if the customer hits the domain with http in front then also he should be redirected to https…for this functionality we can use the loadbalancer listerners.  
In listener annotation open both 80 and 443 and then in edit the listner 80 to redirect the traffic from 80 to 443…and you are done. (**redirect the URL**) option