**4. Update SSL Ingress Annotation, Deploy and Test**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/08-NEW-ELB-Application-LoadBalancers/08-04-ALB-Ingress-SSL>

**Add annotations related to SSL**

--- 04-ALB-Ingress-SSL.yml

    ## SSL Settings

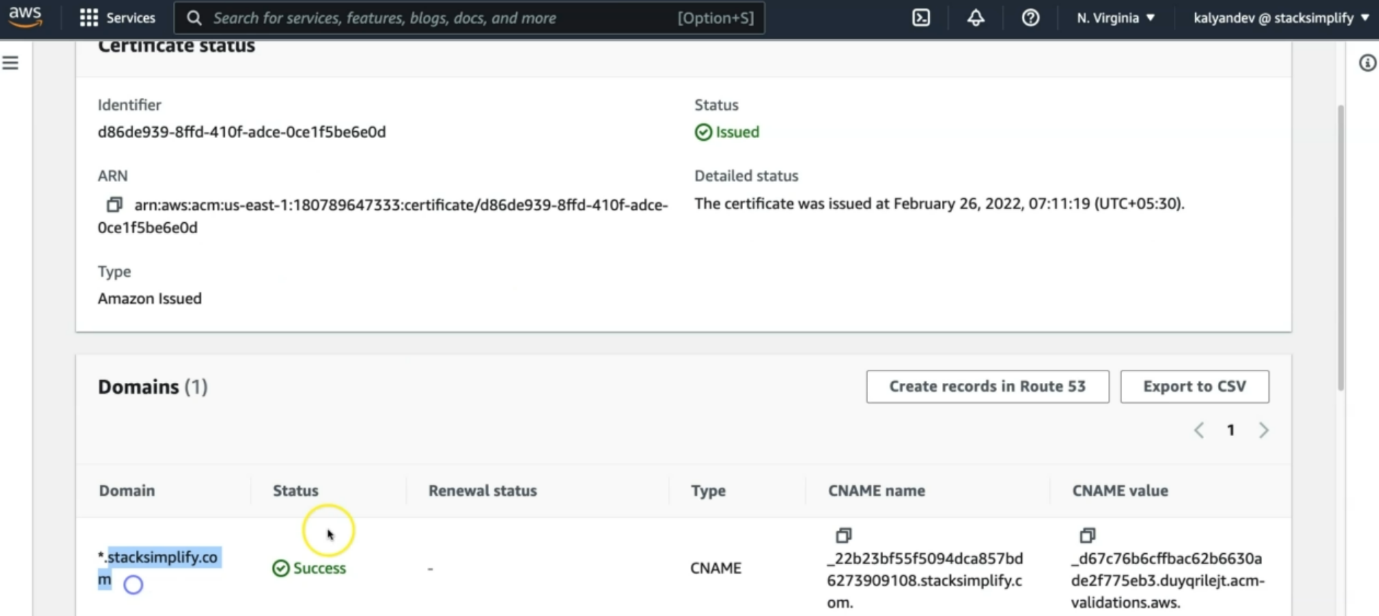
    alb.ingress.kubernetes.io/listen-ports: '[{"HTTPS":443}, {"HTTP":80}]'

    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:us-east-1:180789647333:certificate/632a3ff6-3f6d-464c-9121-b9d97481a76b

    #alb.ingress.kubernetes.io/ssl-policy: ELBSecurityPolicy-TLS-1-1-2017-01 #Optional (Picks default if not used)

--- **note** – those are the extra parameters I added for ssl certificate.

--- we have to copy the arn form aws certificate manager.



--- this is the certificate we have created and we can also see its arn.

**manifests**

--- here, I am copying the all manifests.

--- **04-ALB-Ingress-SSL.yml**

# Annotations Reference: https://kubernetes-sigs.github.io/aws-load-balancer-controller/latest/guide/ingress/annotations/

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  name: ingress-ssl-demo

  annotations:

    # Load Balancer Name

    alb.ingress.kubernetes.io/load-balancer-name: ssl-ingress

    # Ingress Core Settings

    #kubernetes.io/ingress.class: "alb" (OLD INGRESS CLASS NOTATION - STILL WORKS BUT RECOMMENDED TO USE IngressClass Resource)

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

    # Health Check Settings

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

    alb.ingress.kubernetes.io/healthcheck-port: traffic-port

    #Important Note:  Need to add health check path annotations in service level if we are planning to use multiple targets in a load balancer

    alb.ingress.kubernetes.io/healthcheck-interval-seconds: '15'

    alb.ingress.kubernetes.io/healthcheck-timeout-seconds: '5'

    alb.ingress.kubernetes.io/success-codes: '200'

    alb.ingress.kubernetes.io/healthy-threshold-count: '2'

    alb.ingress.kubernetes.io/unhealthy-threshold-count: '2'

    ## SSL Settings

    alb.ingress.kubernetes.io/listen-ports: '[{"HTTPS":443}, {"HTTP":80}]'

    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:us-east-1:180789647333:certificate/d86de939-8ffd-410f-adce-0ce1f5be6e0d

    #alb.ingress.kubernetes.io/ssl-policy: ELBSecurityPolicy-TLS-1-1-2017-01 #Optional (Picks default if not used)

spec:

  ingressClassName: my-aws-ingress-class   # Ingress Class

  defaultBackend:

    service:

      name: app3-nginx-nodeport-service

      port:

        number: 80

  rules:

    - http:

        paths:

          - path: /app1

            pathType: Prefix

            backend:

              service:

                name: app1-nginx-nodeport-service

                port:

                  number: 80

          - path: /app2

            pathType: Prefix

            backend:

              service:

                name: app2-nginx-nodeport-service

                port:

                  number: 80

# Important Note-1: In path based routing order is very important, if we are going to use  "/\*", try to use it at the end of all rules.

# 1. If  "spec.ingressClassName: my-aws-ingress-class" not specified, will reference default ingress class on this kubernetes cluster

# 2. Default Ingress class is nothing but for which ingress class we have the annotation `ingressclass.kubernetes.io/is-default-class: "true"`

--- **01-Nginx-App1-Deployment-and-NodePortService.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: app1-nginx-deployment

  labels:

    app: app1-nginx

spec:

  replicas: 1

  selector:

    matchLabels:

      app: app1-nginx

  template:

    metadata:

      labels:

        app: app1-nginx

    spec:

      containers:

        - name: app1-nginx

          image: stacksimplify/kube-nginxapp1:1.0.0

          ports:

            - containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

  name: app1-nginx-nodeport-service

  labels:

    app: app1-nginx

  annotations:

#Important Note:  Need to add health check path annotations in service level if we are planning to use multiple targets in a load balancer

    alb.ingress.kubernetes.io/healthcheck-path: /app1/index.html

spec:

  type: NodePort

  selector:

    app: app1-nginx

  ports:

    - port: 80

      targetPort: 80

--- **02-Nginx-App2-Deployment-and-NodePortService.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: app2-nginx-deployment

  labels:

    app: app2-nginx

spec:

  replicas: 1

  selector:

    matchLabels:

      app: app2-nginx

  template:

    metadata:

      labels:

        app: app2-nginx

    spec:

      containers:

        - name: app2-nginx

          image: stacksimplify/kube-nginxapp2:1.0.0

          ports:

            - containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

  name: app2-nginx-nodeport-service

  labels:

    app: app2-nginx

  annotations:

#Important Note:  Need to add health check path annotations in service level if we are planning to use multiple targets in a load balancer

    alb.ingress.kubernetes.io/healthcheck-path: /app2/index.html

spec:

  type: NodePort

  selector:

    app: app2-nginx

  ports:

    - port: 80

      targetPort: 80

--- **03-Nginx-App3-Deployment-and-NodePortService.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: app3-nginx-deployment

  labels:

    app: app3-nginx

spec:

  replicas: 1

  selector:

    matchLabels:

      app: app3-nginx

  template:

    metadata:

      labels:

        app: app3-nginx

    spec:

      containers:

        - name: app2-nginx

          image: stacksimplify/kubenginx:1.0.0

          ports:

            - containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

  name: app3-nginx-nodeport-service

  labels:

    app: app3-nginx

  annotations:

#Important Note:  Need to add health check path annotations in service level if we are planning to use multiple targets in a load balancer

    alb.ingress.kubernetes.io/healthcheck-path: /index.html

spec:

  type: NodePort

  selector:

    app: app3-nginx

  ports:

    - port: 80

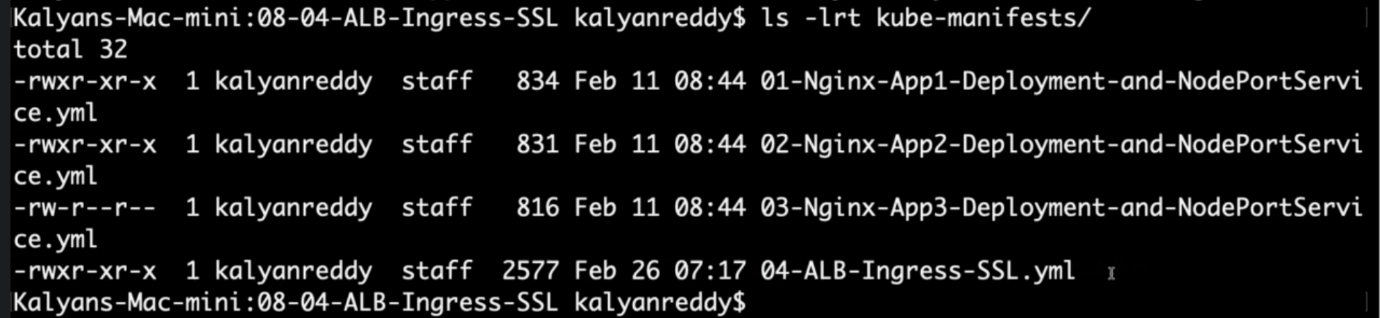
      targetPort: 80

**Deploy all manifests and test**

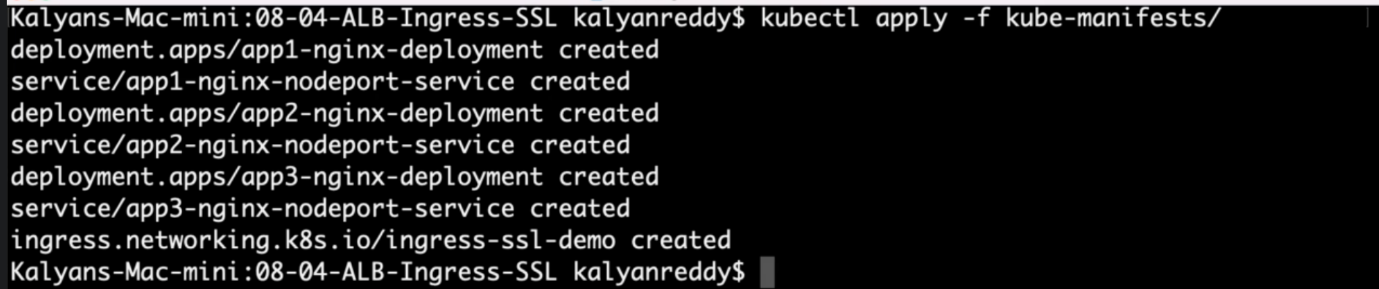
**Deploy and Verify**

**# Deploy kube-manifests**

--- **ls -lrt kube-manifests** – list the files which are inside of this folder.

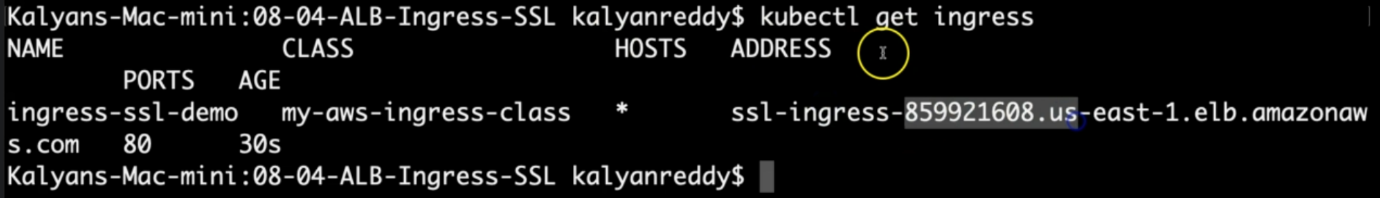


--- **kubectl apply -f kube-manifests/**

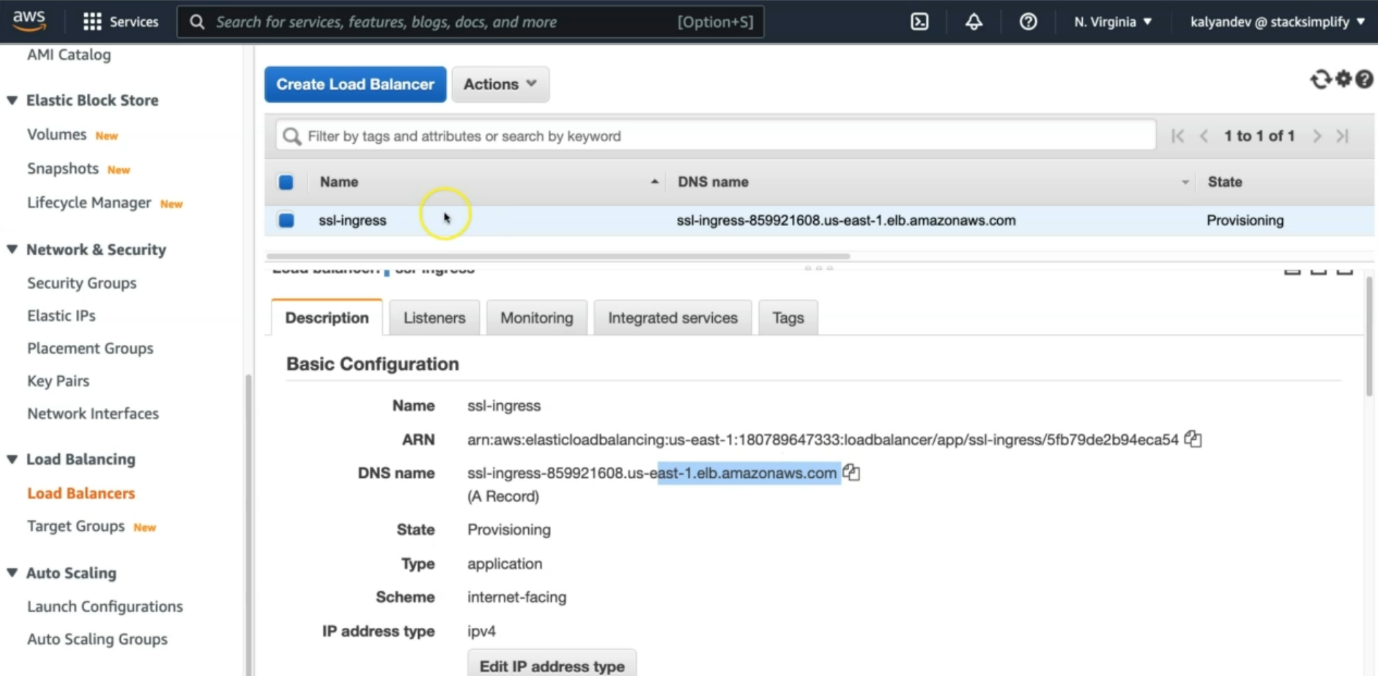


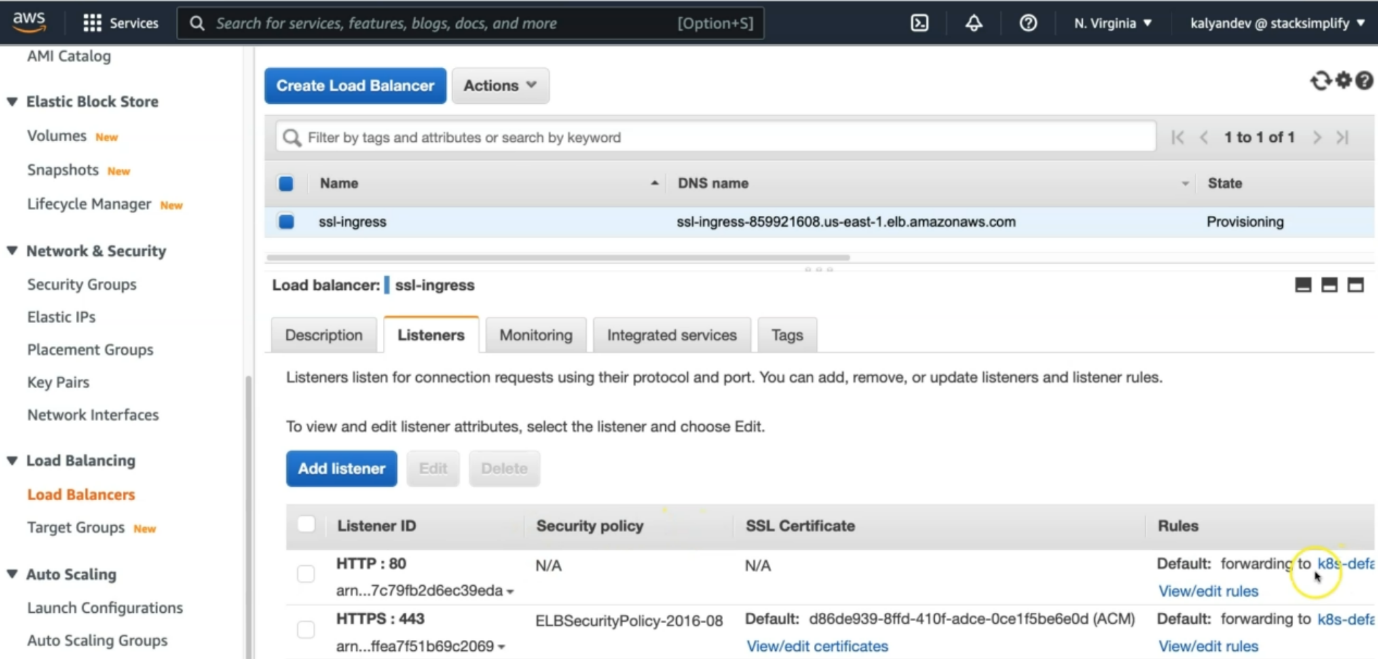
**# Verify Ingress Resource**

--- **kubectl get ingress**



--- this address should match with aws load balancer DNS name

--- click on the listeners



--- it listening to 80 and 443.

**# Verify Apps**

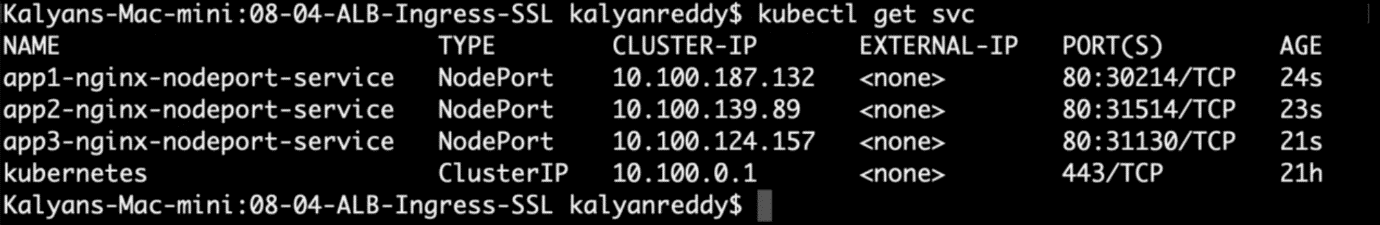
--- **kubectl get deploy**

--- **kubectl get pods**

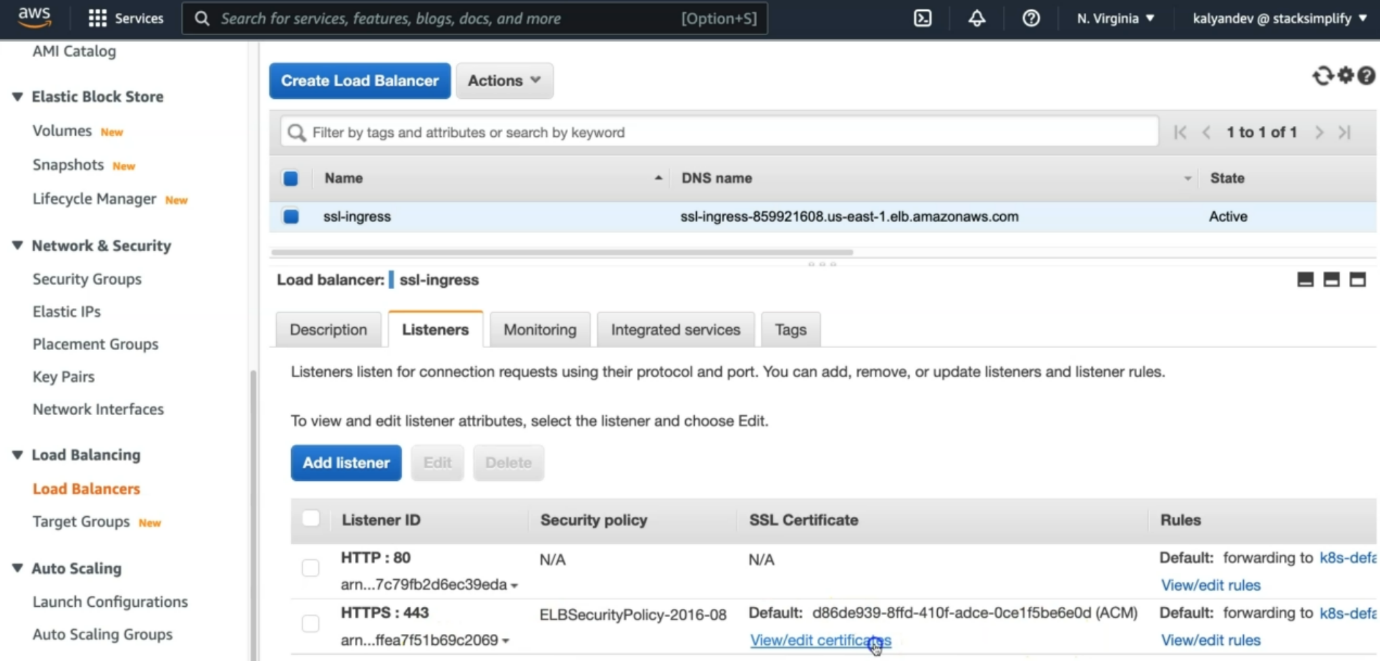


**# Verify NodePort Services**

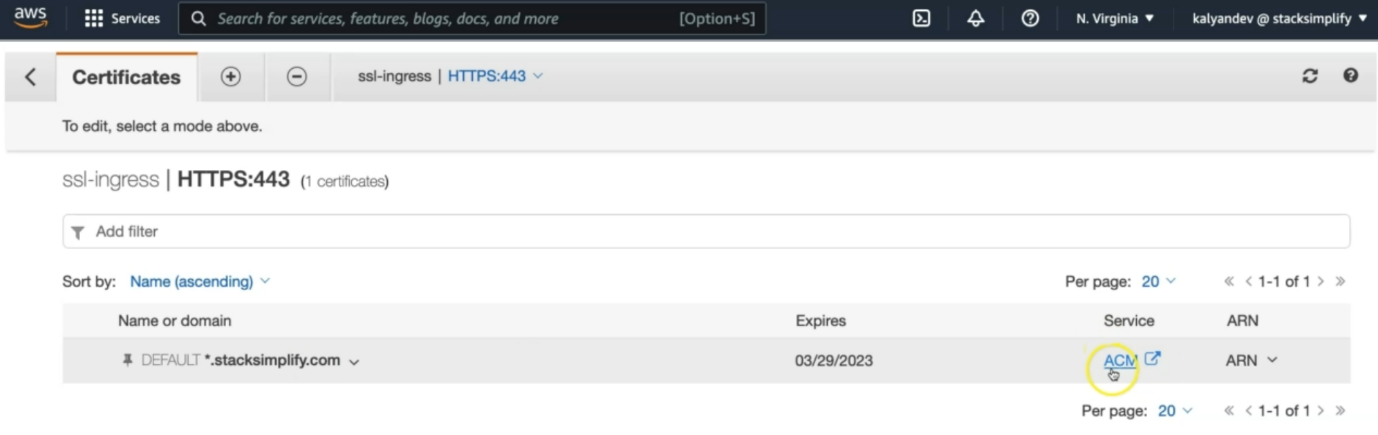
--- **kubectl get svc**



**Ssl certificate**



--- **note** – click on view/edit certificates.



--- the certificate is associated. Click on ACM. It will take us to AWS certificate manager. We have successfully associated the SSL certificate.

--- **NOTE** – we have associated the certificate using certificate related arn in ingress manifest.

**Verify Load Balancer & Target Groups**

--- Load Balancer - Listeners (Verify both 80 & 443)

--- Load Balancer - Rules (Verify both 80 & 443 listeners)

--- Target Groups - Group Details (Verify Health check path)

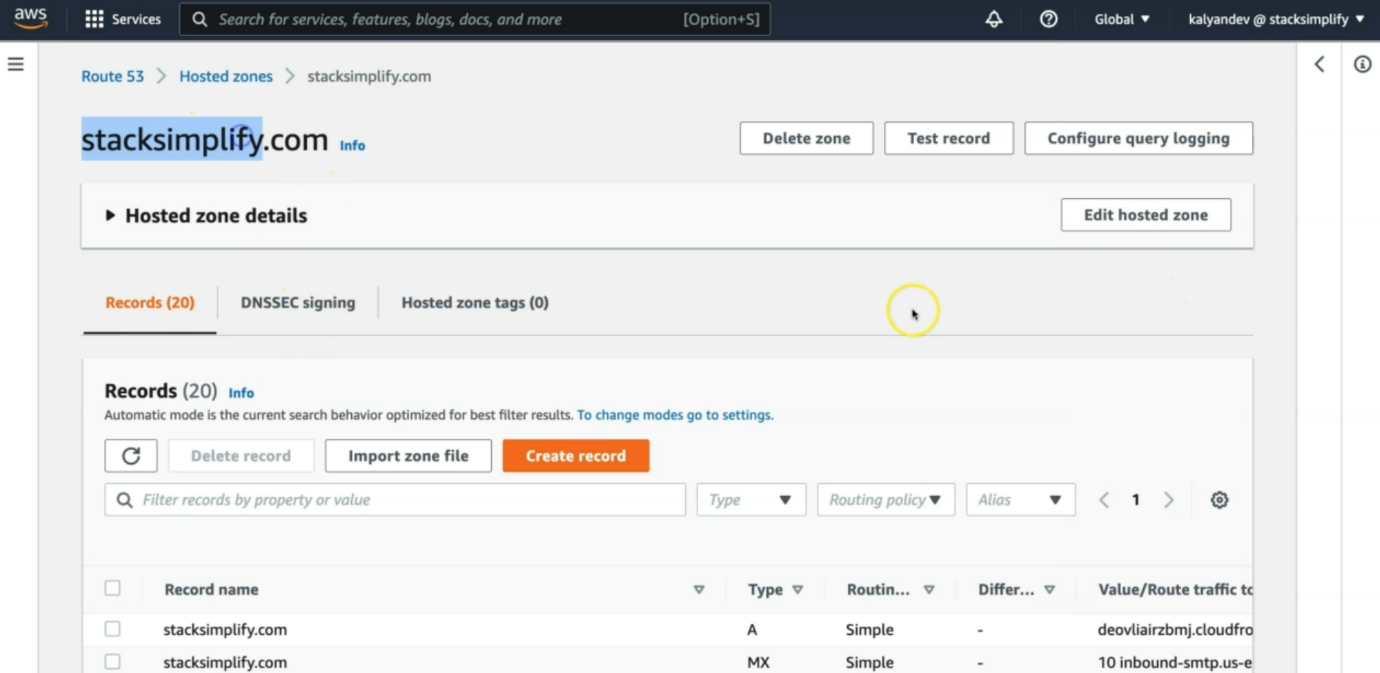
--- Target Groups - Targets (Verify all 3 targets are healthy)

**Add DNS in Route53**

--- Go to Services -> Route 53

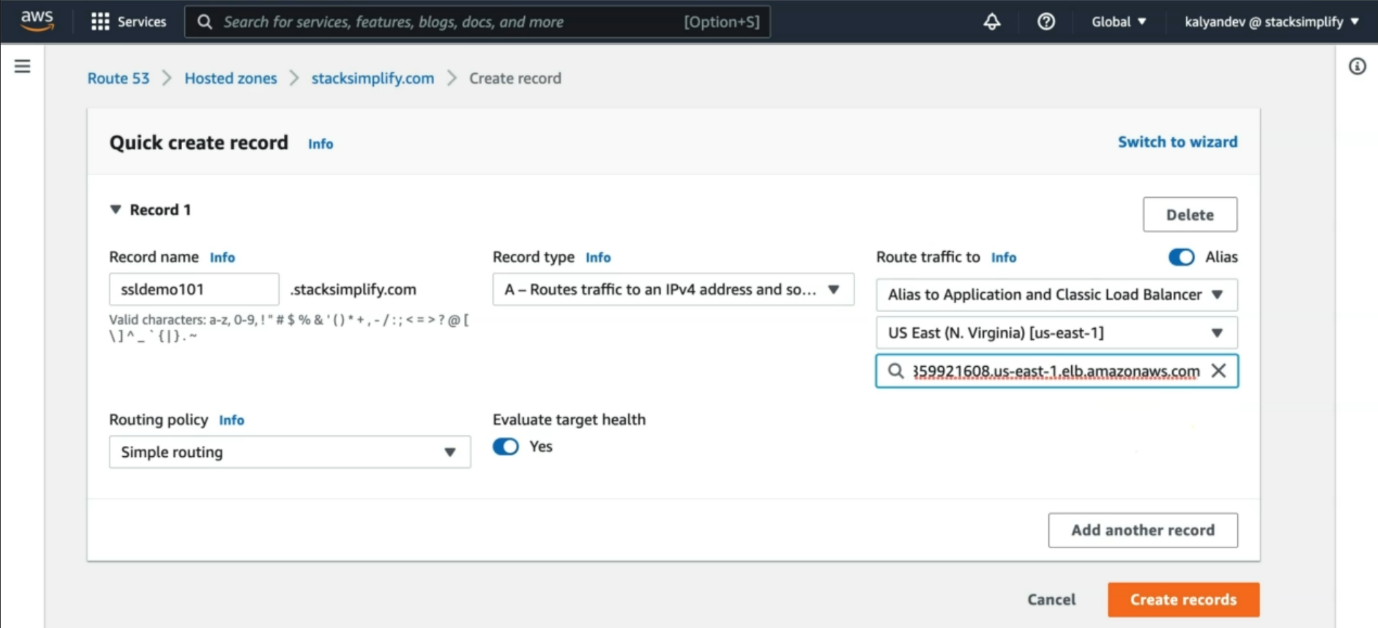
--- Go to Hosted Zones

Click on yourdomain.com (in my case stacksimplify.com)



--- Create a Record Set

1. Name: ssldemo101.stacksimplify.com
2. Alias: yes
3. Alias Target: Copy our ALB DNS Name here (Sample: ssl-ingress-551932098.us-east-1.elb.amazonaws.com)
4. Click on Create



--- click on create records.

**Access Application using newly registered DNS Name**

--- Access Application

--- Important Note: Instead of stacksimplify.com you need to replace with your registered Route53 domain (Refer pre-requisite Step-02)

**# HTTP URLs**

--- <http://ssldemo101.stacksimplify.com/app1/index.html>

--- <http://ssldemo101.stacksimplify.com/app2/index.html>

--- <http://ssldemo101.stacksimplify.com/>

**# HTTPS URLs**

--- <https://ssldemo101.stacksimplify.com/app1/index.html>

--- <https://ssldemo101.stacksimplify.com/app2/index.html>

--- <https://ssldemo101.stacksimplify.com/>

**Annotation Reference**

--- AWS Load Balancer Controller Annotation Reference - <https://kubernetes-sigs.github.io/aws-load-balancer-controller/v2.4/guide/ingress/annotations/>