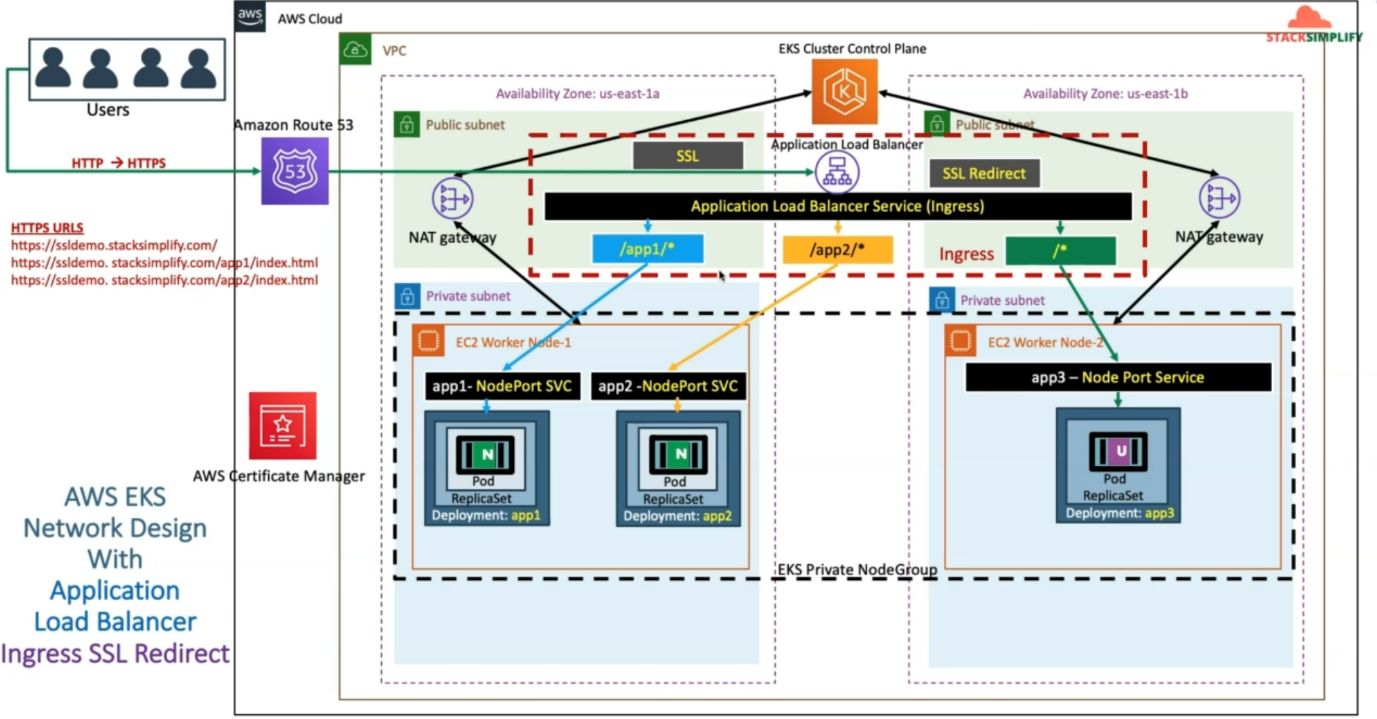
**5: Update SSL Ingress Redirection Annotation, Deploy, Test and Clean-up**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/08-NEW-ELB-Application-LoadBalancers/08-05-ALB-Ingress-SSL-Redirect>

--- **note** – in this lesion we will learn about ssl redirection use case.



--- this network diagram is from our previous class, which is ssl ingress. As part of this lesion, we will add annotations related to SSL redirect in the ingress service. Whenever we redeploy this kubernetes manifest, it is going to do a http to https redirect for all the URLS.

--- which means all the applications deployed using this ingress service will be accessed only using https. So, whenever you access the application using http, it will be redirected to https.

**Add annotations related to SSL Redirect**

--- Redirect from HTTP to HTTPS

--- File Name: **04-ALB-Ingress-SSL-Redirect.yml**

    # SSL Redirect Setting

    alb.ingress.kubernetes.io/ssl-redirect: '443'

--- **04-ALB-Ingress-SSL-Redirect.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)

    # SSL Redirect Setting

    alb.ingress.kubernetes.io/ssl-redirect: '443'

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

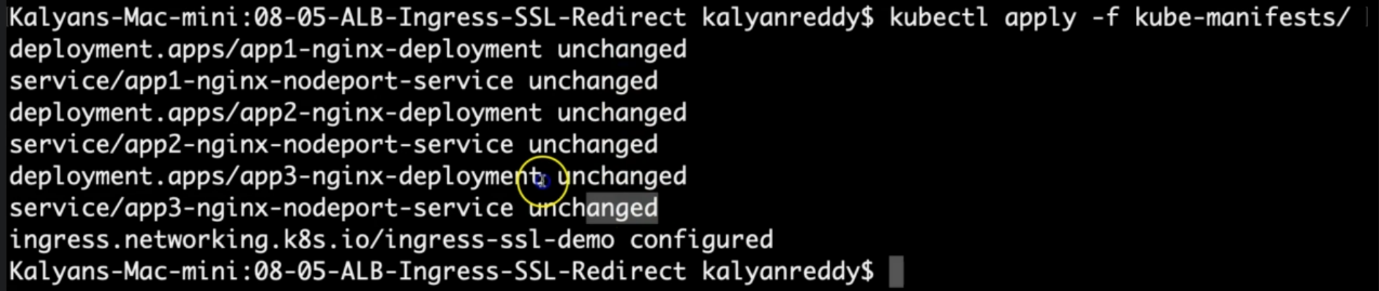
# 2. Default Ingress class is nothing but for which ingress class we have the annotation `ingressclass.kubernetes.io/is-default-class: "true"`

**Deploy all manifests and test**

**Deploy and Verify**

**# Deploy kube-manifests**

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



**# Verify Ingress Resource**

--- **kubectl get ingress**

**# Verify Apps**

--- **kubectl get deploy**

--- **kubectl get pods**

**# Verify NodePort Services**

--- **kubectl get svc**

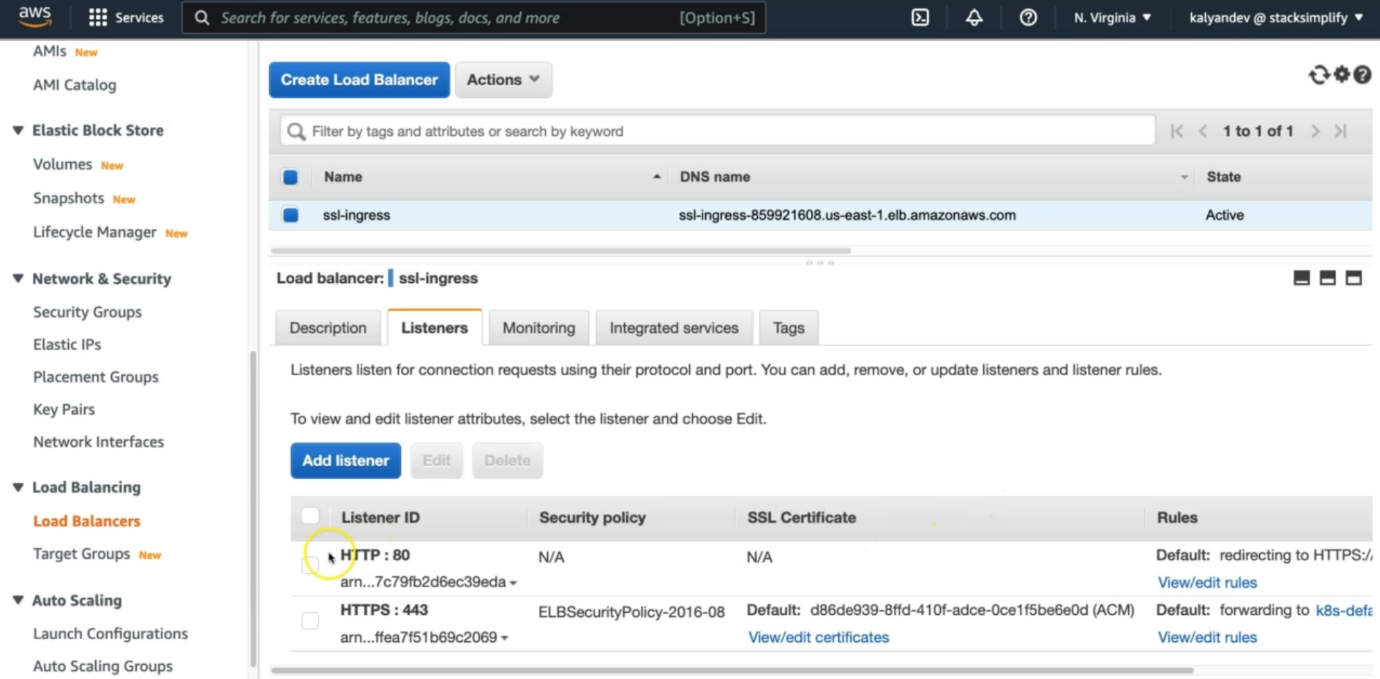
**Verify Load Balancer & Target Groups**

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

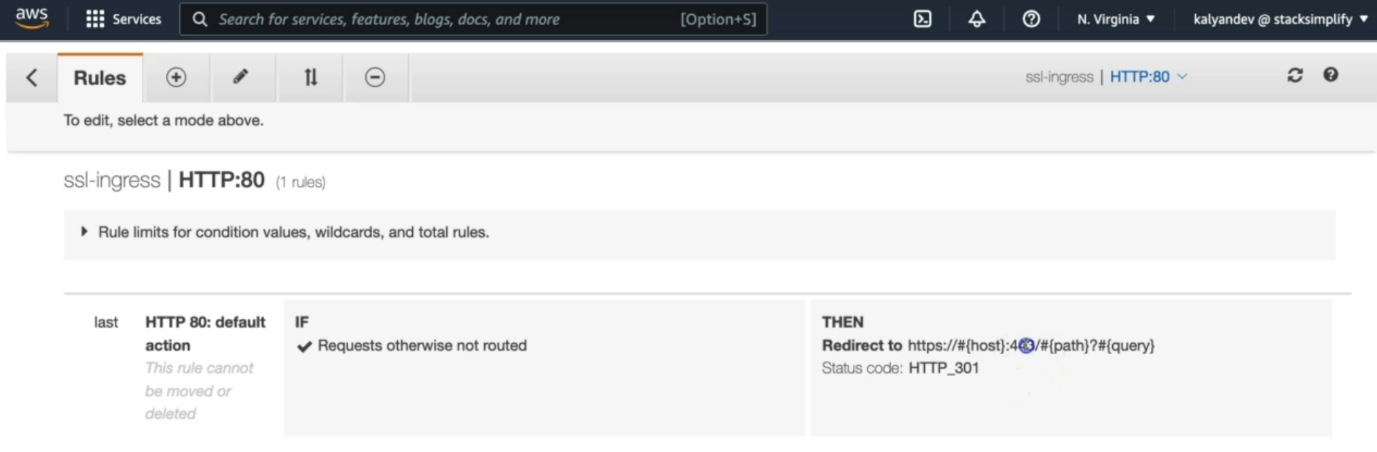
--- Load Balancer - Rules (Verify both 80 & 443 listeners)

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

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



--- **note** – inside of HTTP:80, you should not find any node port service because it if we access the application using https, it will redirect to https. Click on view/edit rules.



--- one thing, it is saying. That is redirected to https:443. This is because of whatever annotations we added in the ingress manifest.

**Access Application using newly registered DNS Name**

--- Access Application

**# HTTP URLs (Should Redirect to HTTPS)**

--- <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/>

**Clean Up**

**# Delete Manifests**

--- **kubectl delete -f kube-manifests/**

**## Delete Route53 Record Set**

--- **Delete Route53 Record we created (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/>