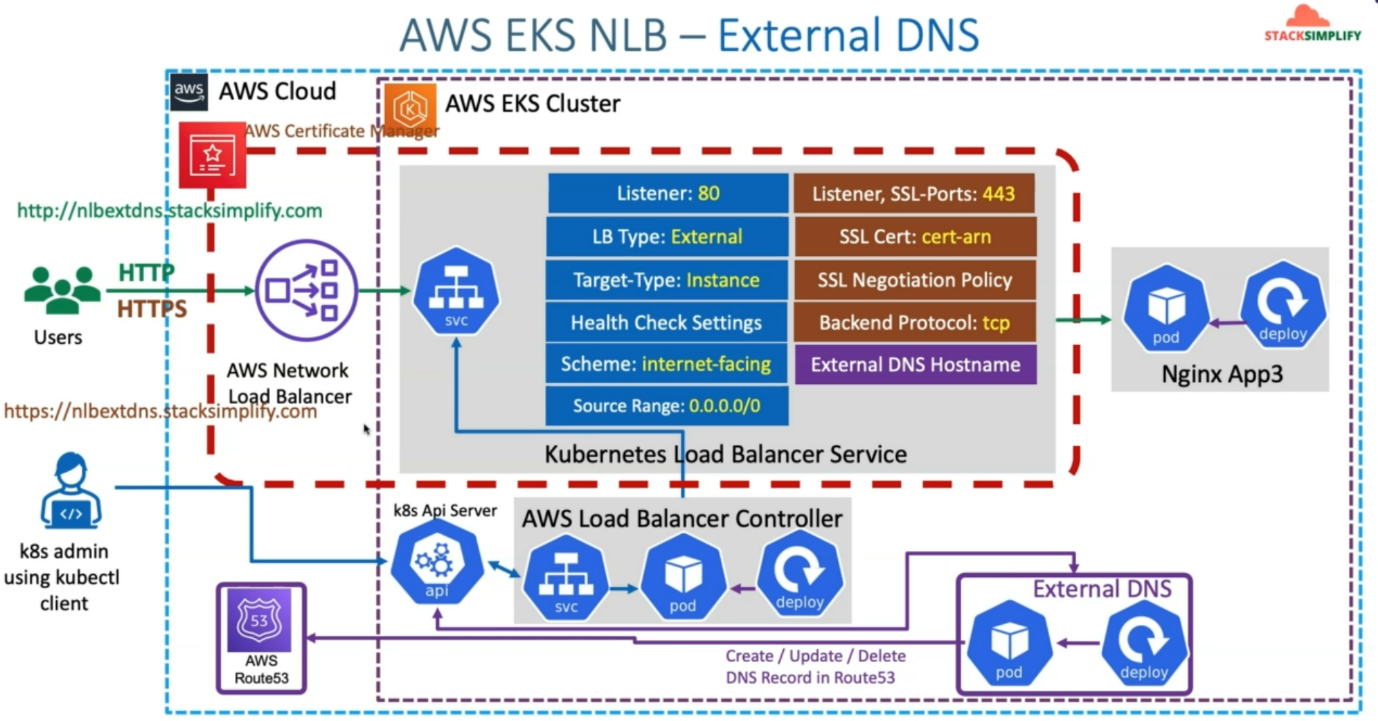
**06. NLB External DNS Demo using k8s Service**

**Aws perspective design**



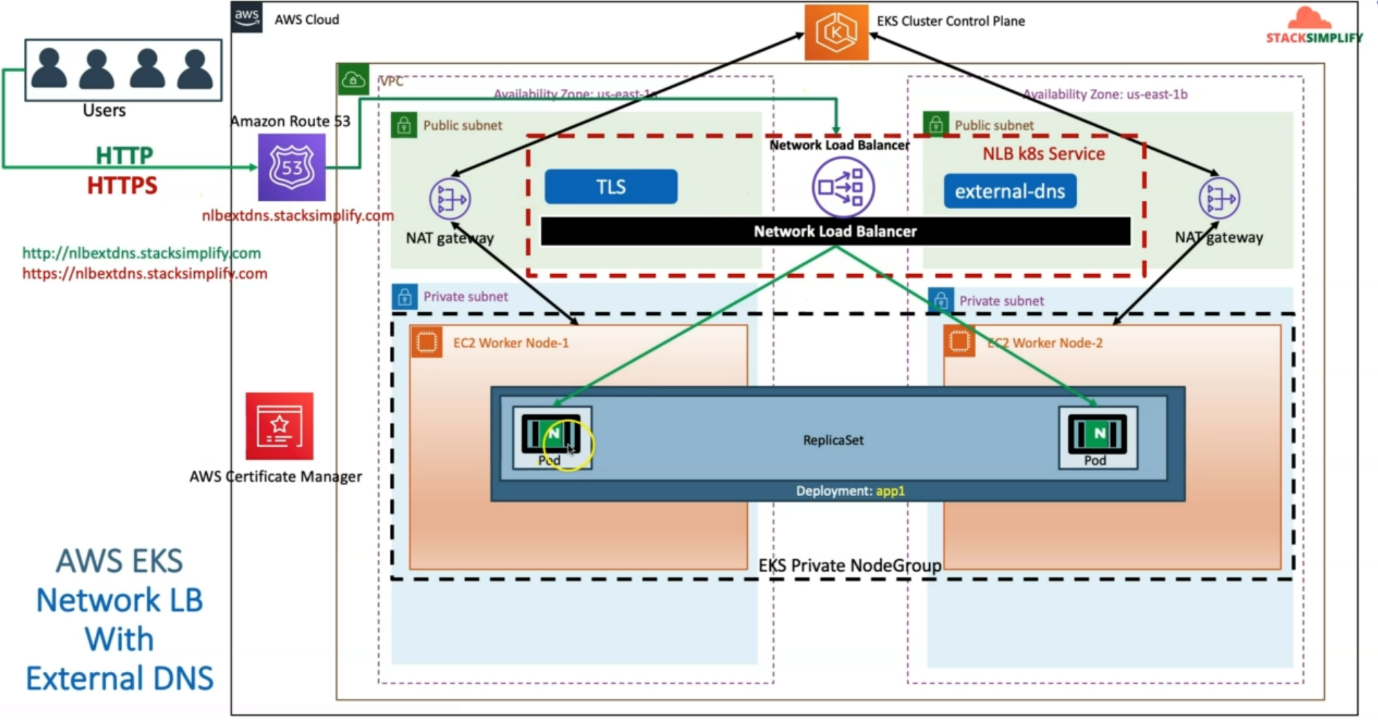
--- all these things we have implemented in previous demo. So, now we are going to add external DNS hostname related annotation to our Kubernetes service.

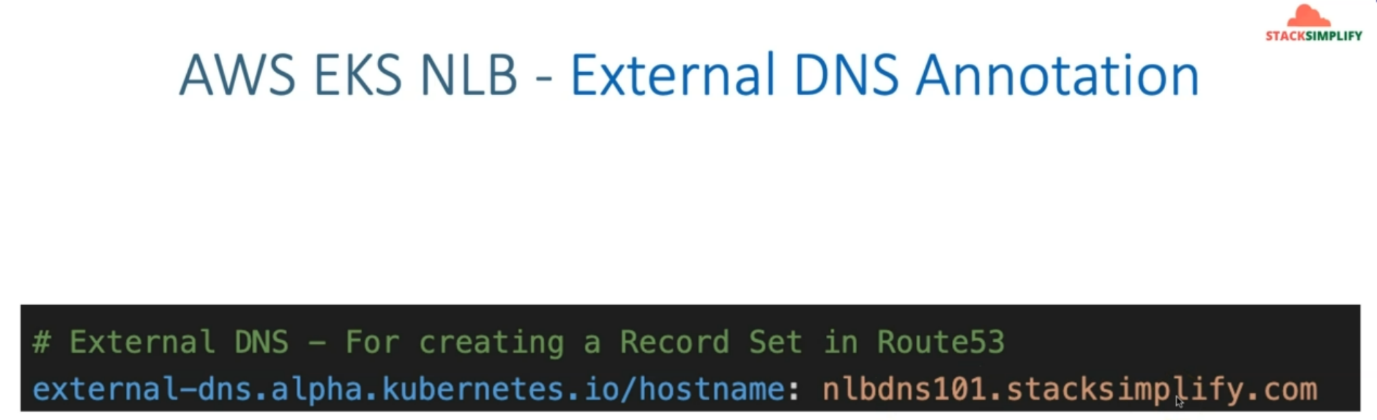
--- we already deployed our external DNS related pod on our EKS cluster. Whenever we are implementing the ALB ingress related use cases.

--- this external DNS pod will be watching this k8s API server for any ingress service resources, which contains external DNS hostname annotation, if it finds, then automatically, it is going to go ahead and then pick that annotation and create or update or delete the DNS record in AWS Route53.

--- **note** - here you will not get any ssl certification error because you’re assigning ssl certificates.

**Network design kubernetes perspective**





--- **note** – the above annotation should be added.

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/19-ELB-Network-LoadBalancers-with-LBC/19-03-LBC-NLB-ExternalDNS>

**Introduction**

--- Implement External DNS Annotation in NLB Kubernetes Service Manifest

**Review External DNS Annotations**

--- File Name: **kube-manifests\02-LBC-NLB-LoadBalancer-Service.yml**

**# External DNS - For creating a Record Set in Route53**

**external-dns.alpha.kubernetes.io/hostname: nlbdns101.stacksimplify.com**

apiVersion: v1

kind: Service

metadata:

  name: extdns-lbc-network-lb

  annotations:

    # Traffic Routing

    service.beta.kubernetes.io/aws-load-balancer-name: extdns-lbc-network-lb

    service.beta.kubernetes.io/aws-load-balancer-type: external

    service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: instance

    #service.beta.kubernetes.io/aws-load-balancer-subnets: subnet-xxxx, mySubnet ## Subnets are auto-discovered if this annotation is not specified, see Subnet Discovery for further details.

    # Health Check Settings

    service.beta.kubernetes.io/aws-load-balancer-healthcheck-protocol: http

    service.beta.kubernetes.io/aws-load-balancer-healthcheck-port: traffic-port

    service.beta.kubernetes.io/aws-load-balancer-healthcheck-path: /index.html

    service.beta.kubernetes.io/aws-load-balancer-healthcheck-healthy-threshold: "3"

    service.beta.kubernetes.io/aws-load-balancer-healthcheck-unhealthy-threshold: "3"

    service.beta.kubernetes.io/aws-load-balancer-healthcheck-interval: "10" # The controller currently ignores the timeout configuration due to the limitations on the AWS NLB. The default timeout for TCP is 10s and HTTP is 6s.

    # Access Control

    service.beta.kubernetes.io/load-balancer-source-ranges: 0.0.0.0/0

    service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"

    # AWS Resource Tags

    service.beta.kubernetes.io/aws-load-balancer-additional-resource-tags: Environment=dev,Team=test

    # TLS

    service.beta.kubernetes.io/aws-load-balancer-ssl-cert: arn:aws:acm:us-east-1:180789647333:certificate/d86de939-8ffd-410f-adce-0ce1f5be6e0d

    service.beta.kubernetes.io/aws-load-balancer-ssl-ports: 443, # Specify this annotation if you need both TLS and non-TLS listeners on the same load balancer

    service.beta.kubernetes.io/aws-load-balancer-ssl-negotiation-policy: ELBSecurityPolicy-TLS13-1-2-2021-06

    service.beta.kubernetes.io/aws-load-balancer-backend-protocol: tcp

    # External DNS - For creating a Record Set in Route53

    external-dns.alpha.kubernetes.io/hostname: nlbdns101.stacksimplify.com

spec:

  type: LoadBalancer

  selector:

    app: app3-nginx

  ports:

    - name: http

      port: 80

      targetPort: 80

    - name: https

      port: 443

      targetPort: 80

--- **01-Nginx-App3-Deployment.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

**Deploy all kube-manifests**

**# Verify if External DNS Pod exists and Running**

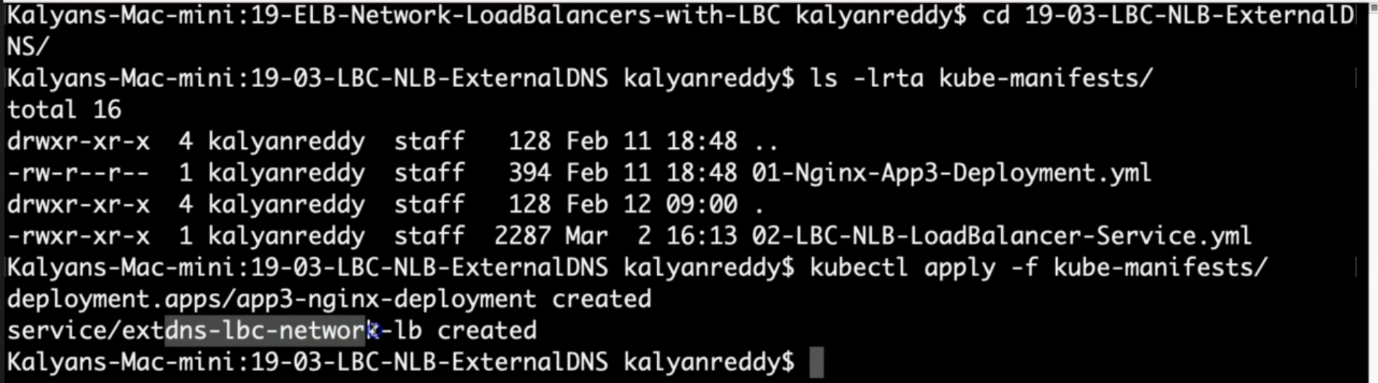
--- **kubectl get pods**

**Observation:**

external-dns pod should be running, make sure that your external dns is running otherwise you cannot do the below deployment.

**# Deploy kube-manifests**

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



**# Verify Pods**

--- **kubectl get pods**



**# Verify Services**

--- **kubectl get svc**

Observation:

1. Verify the network lb DNS name

**# Verify AWS Load Balancer Controller pod logs**

--- **kubectl -n kube-system get pods**

--- **kubectl -n kube-system logs -f <aws-load-balancer-controller-POD-NAME>**

**# Verify using AWS Mgmt Console**

Go to Services -> EC2 -> Load Balancing -> Load Balancers

1. Verify Description Tab - DNS Name matching output of "kubectl get svc" External IP

2. Verify Listeners Tab

Observation: Should see two listeners Port 80 and 443

Go to Services -> EC2 -> Load Balancing -> Target Groups

1. Verify Registered targets

2. Verify Health Check path

Observation: Should see two target groups. 1 Target group for 1 listener

**# Verify External DNS logs**

--- **kubectl logs -f $(kubectl get po | egrep -o 'external-dns[A-Za-z0-9-]+')**

**# Perform nslookup Test**

--- **nslookup nlbdns101.stacksimplify.com**

**# Access Application**

**# Test HTTP URL**

---- <http://nlbdns101.stacksimplify.com>

**# Test HTTPS URL**

--- <https://nlbdns101.stacksimplify.com>

**Clean-Up**

**# Delete or Undeploy kube-manifests**

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

**# Verify if NLB deleted**

In AWS Mgmt Console,

Go to Services -> EC2 -> Load Balancing -> Load Balancers

**References**

--- Network Load Balancer - <https://docs.aws.amazon.com/eks/latest/userguide/network-load-balancing.html>

--- NLB Service - <https://kubernetes-sigs.github.io/aws-load-balancer-controller/v2.4/guide/service/nlb/>

--- NLB Service Annotations - <https://kubernetes-sigs.github.io/aws-load-balancer-controller/v2.4/guide/service/annotations/>