**5. Create, Deploy Ingress Rules and verify and clean-up**

--- **Reference** - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/08-NEW-ELB-Application-LoadBalancers/08-02-ALB-Ingress-Basics>

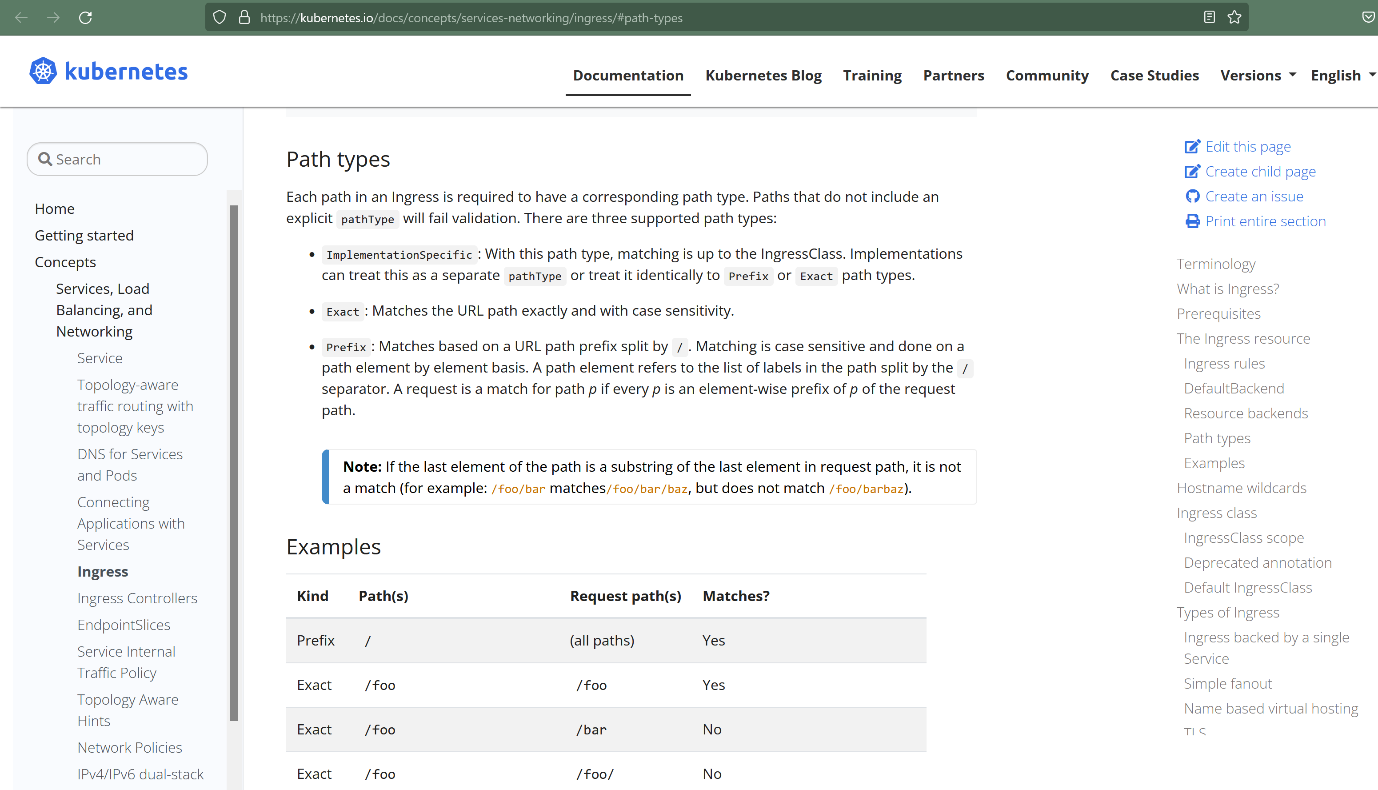
--- **note** – here we are going to define ingress manifest with ingress rule, before going to understand path types details.

**Review Ingress kube-manifest with Ingress Rules**

--- Discuss about Ingress Path Types - <https://kubernetes.io/docs/concepts/services-networking/ingress/#path-types>

--- we have 3 types of the paths in ingress perspective.

1. Implementationspecific
2. Exact
3. prefix



--- Better Path Matching With Path Types - <https://kubernetes.io/blog/2020/04/02/improvements-to-the-ingress-api-in-kubernetes-1.18/#better-path-matching-with-path-types>

--- Sample Ingress Rule - <https://kubernetes.io/docs/concepts/services-networking/ingress/#the-ingress-resource>

--- **ImplementationSpecific (default):** With this path type, matching is up to the controller implementing the IngressClass. Implementations can treat this as a separate pathType or treat it identically to the Prefix or Exact path types.

--- **Exact**: Matches the URL path exactly and with case sensitivity.

--- **Prefix**: Matches based on a URL path prefix split by /. Matching is case sensitive and done on a path element by element basis.

--- **File Location**: **02-kube-manifests-rules\02-ALB-Ingress-Basic.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-nginxapp1

  labels:

    app: app1-nginx

  annotations:

    # Load Balancer Name

    alb.ingress.kubernetes.io/load-balancer-name: app1ingressrules

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

    # Ingress Core Settings

    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

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

    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'

spec:

  ingressClassName: ic-external-lb # Ingress Class

  rules:

    - http:

        paths:

          - path: /

            pathType: Prefix

            backend:

              service:

                name: app1-nginx-nodeport-service

                port:

                  number: 80

# 1. If  "spec.ingressClassName: ic-external-lb" 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 kube-manifests and Verify**

**# Change Directory**

--- **cd 08-02-ALB-Ingress-Basics**

**# Deploy kube-manifests**

--- **kubectl apply -f 02-kube-manifests-rules/**

**# Verify k8s Deployment and Pods**

--- **kubectl get deploy**

--- **kubectl get pods**

**# Verify Ingress (Make a note of Address field)**

--- **kubectl get ingress**

**Obsevation:**

1. Verify the ADDRESS value, we should see something like "app1ingressrules-154912460.us-east-1.elb.amazonaws.com"

**# Describe Ingress Controller**

--- **kubectl describe ingress ingress-nginxapp1**

**Observation:**

1. Review Default Backend and Rules

**# List Services**

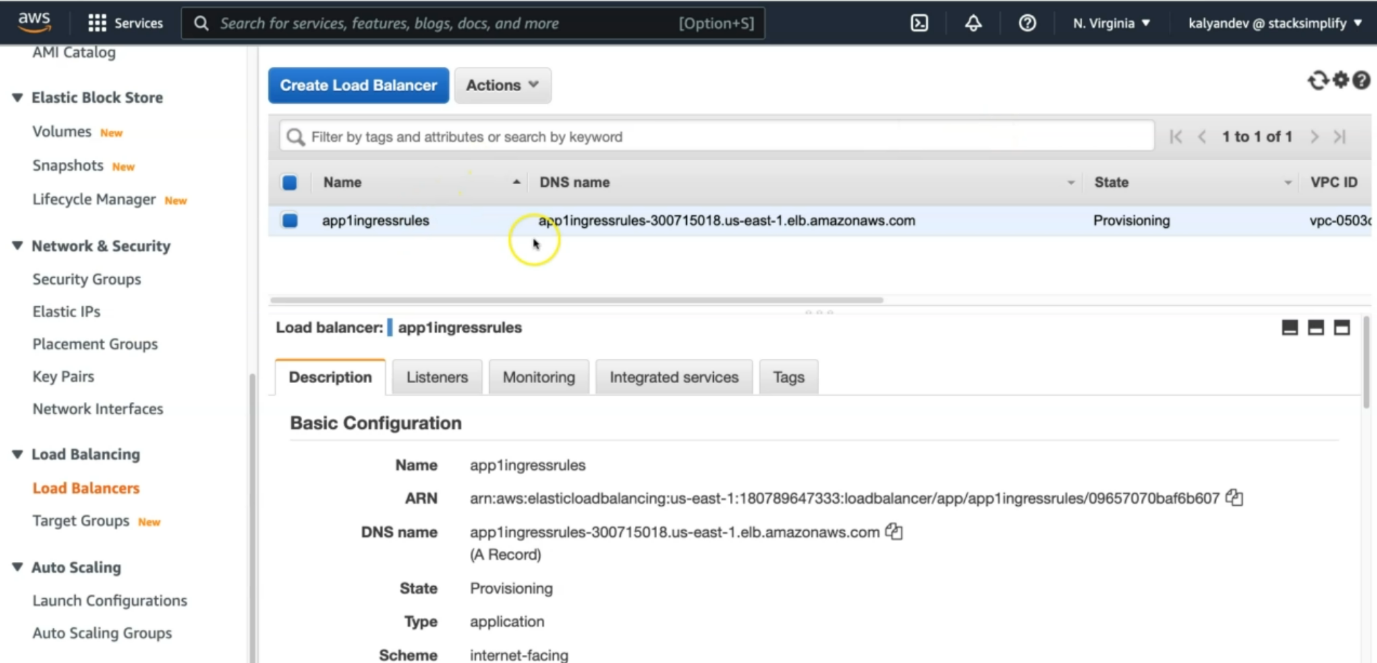
--- **kubectl get svc**

**# Verify Application Load Balancer using**

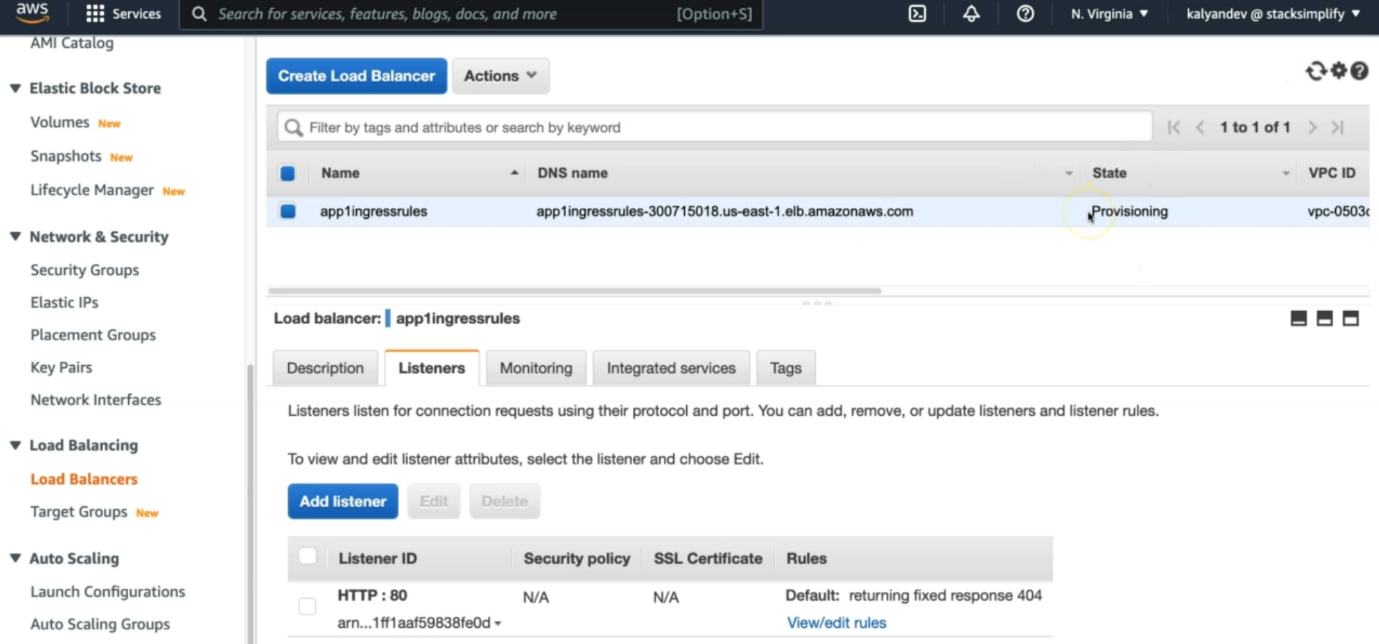
Goto AWS Mgmt Console -> Services -> EC2 -> Load Balancers

1. Verify Listeners and Rules inside a listener

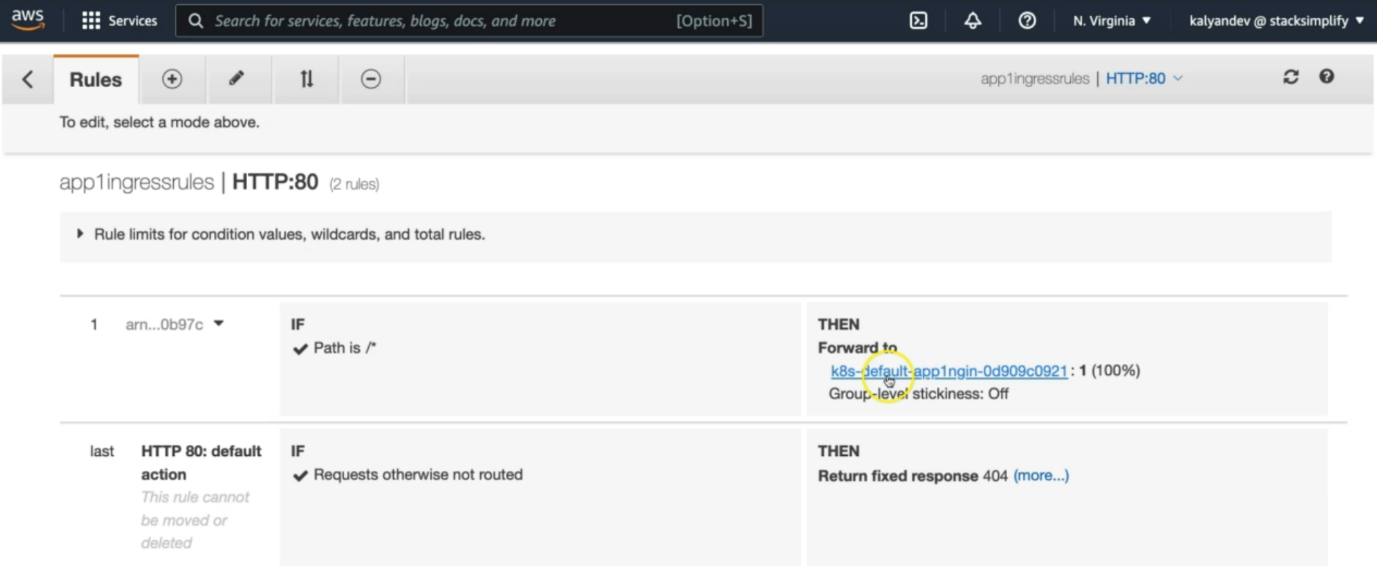
2. Verify Target Groups



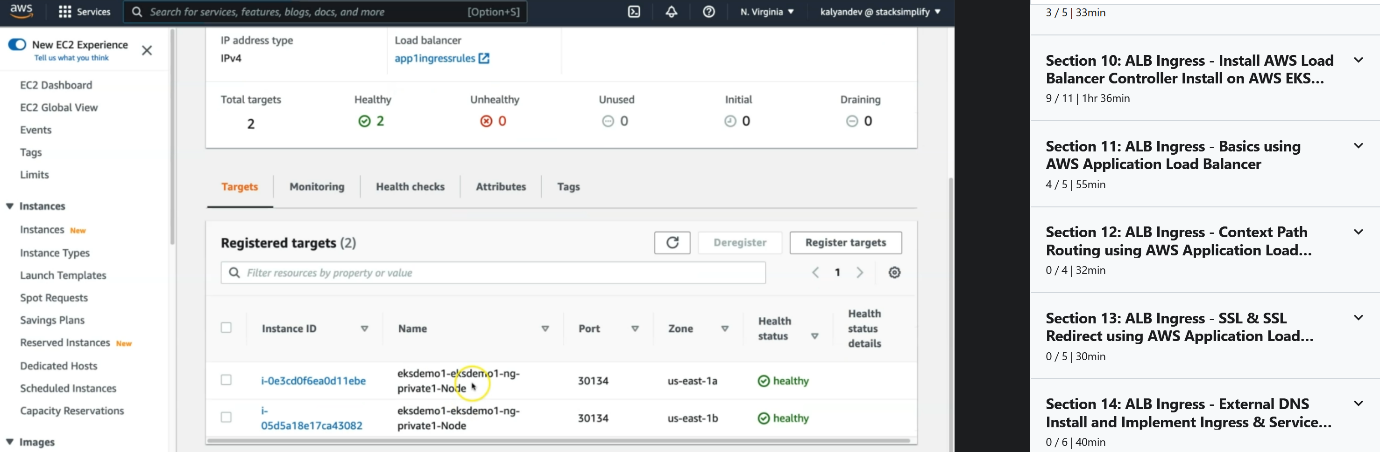
--- click on listeners.



--- it is listening on HTTP:80, click on view/edit rules.



--- **note** – there is no backend for us, the path is /\*, the traffic will go to the app1nignx target group. Inside of that target group we have our respective node port service-related information.



**# Access App using Browser**

--- **kubectl get ingress**

--- **http://<ALB-DNS-URL>**

--- **http://<ALB-DNS-URL>/app1/index.html**

or

--- **http://<INGRESS-ADDRESS-FIELD>**

--- **http://<INGRESS-ADDRESS-FIELD>/app1/index.html**

**# Sample from my environment (for reference only)**

--- **http://app1ingressrules-154912460.us-east-1.elb.amazonaws.com**

--- **http://app1ingressrules-154912460.us-east-1.elb.amazonaws.com/app1/index.html**

**# Verify AWS Load Balancer Controller logs**

--- **kubectl get po -n kube-system**

--- **kubectl logs -f aws-load-balancer-controller-794b7844dd-8hk7n -n kube-system**

**Clean Up**

**# Delete Kubernetes Resources**

--- **kubectl delete -f 02-kube-manifests-rules/**

**# Verify if Ingress Deleted successfully**

--- **kubectl get ingress**

--- **Important Note:** It is going to cost us heavily if we leave ALB load balancer idle without deleting it properly

**# Verify Application Load Balancer DELETED**

--- Goto AWS Mgmt Console -> Services -> EC2 -> Load Balancers