**7. EKS Fargate - Deploy 3 Apps & Test, 2 On Fargate and 1 on EC2Node Group**

**Deploy Apps**

--- Pre-requisite Check: Verify if RDS DB which is required for UMS Service is UP and RUNNING.

**02-Applications/**

--- under this folder, you will find all the 3 application folders separately. In each application folder, their manifests will be present.

**01-ns-app1**

--- **01-ns-app1 is 1st application folder, now we will see the app manifests under this.**

**--- 01-namespace.yml**

apiVersion: v1

kind: Namespace

metadata:

  name: ns-app1

# Apps deployed in this namespace will run on a EC2 Managed Node Group

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

apiVersion: apps/v1

kind: Deployment

metadata:

  name: app1-nginx-deployment

  labels:

    app: app1-nginx

  namespace: ns-app1

spec:

  replicas: 2

  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

          resources:

            requests:

              memory: "128Mi"

              cpu: "500m"

            limits:

              memory: "500Mi"

              cpu: "1000m"

---

apiVersion: v1

kind: Service

metadata:

  name: app1-nginx-nodeport-service

  labels:

    app: app1-nginx

  namespace: ns-app1

  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

**--- 03-ALB-Ingress-SSL-Redirect-with-ExternalDNS.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: app1-ingress-service

  labels:

    app: app1-nginx

  namespace: ns-app1

  annotations:

    # Load Balancer Name

    alb.ingress.kubernetes.io/load-balancer-name: app1-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'

    # External DNS - For creating a Record Set in Route53

    external-dns.alpha.kubernetes.io/hostname: app1101.stacksimplify.com

spec:

  rules:

    - http:

        paths:

          - path: /app1

            pathType: Prefix

            backend:

              service:

                name: app1-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.

**02-ns-app2**

--- **02-ns-app2 I 2nd application folder, now we will see the app manifests under this.**

**--- 01-namespae.yml**

apiVersion: v1

kind: Namespace

metadata:

  name: ns-app2

# Apps deployed in this namespace will run on a Fargate fp-app2

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

apiVersion: apps/v1

kind: Deployment

metadata:

  name: app2-nginx-deployment

  labels:

    app: app2-nginx

  namespace: ns-app2

spec:

  replicas: 2

  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

          resources:

            requests:

              memory: "128Mi"

              cpu: "500m"

            limits:

              memory: "500Mi"

              cpu: "1000m"

---

apiVersion: v1

kind: Service

metadata:

  name: app2-nginx-nodeport-service

  labels:

    app: app2-nginx

  namespace: ns-app2

  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

    # For Fargate

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

spec:

  type: NodePort

  selector:

    app: app2-nginx

  ports:

    - port: 80

      targetPort: 80

**--- 03-ALB-Ingress-SSL-Redirect-with-ExternalDNS.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: app2-ingress-service

  labels:

    app: app2-nginx

  namespace: ns-app2

  annotations:

    # Load Balancer Name

    alb.ingress.kubernetes.io/load-balancer-name: app2-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'

    # External DNS - For creating a Record Set in Route53

    external-dns.alpha.kubernetes.io/hostname: app2201.stacksimplify.com

    # For Fargate

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

spec:

  rules:

    - http:

        paths:

          - 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.

**03-ns-ums**

--- 03**-ns-app3 Is 3rd application folder, now we will see the app manifests under this.**

**--- 01-namespace.yml**

apiVersion: v1

kind: Namespace

metadata:

  name: ns-ums

# Apps deployed in this namespace will run on a Fargate fp-ums

**--- 02-MySQL-externalName-Service.yml**

apiVersion: v1

kind: Service

metadata:

  name: mysql

  labels:

    runon: fargate

  namespace: ns-ums

spec:

  type: ExternalName

  externalName: usermgmtdb.cxojydmxwly6.us-east-1.rds.amazonaws.com #exteranl db endpoint.

**--- 03-UserManagementMicroservice-Deployment-Service.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: usermgmt-microservice

  labels:

    app: usermgmt-restapp

    runon: fargate

  namespace: ns-ums

spec:

  replicas: 2

  selector:

    matchLabels:

      app: usermgmt-restapp

  template:

    metadata:

      labels:

        app: usermgmt-restapp

        runon: fargate

    spec:

      initContainers:

        - name: init-db

          image: busybox:1.31

          command: ['sh', '-c', 'echo -e "Checking for the availability of MySQL Server deployment"; while ! nc -z mysql 3306; do sleep 1; printf "-"; done; echo -e "  >> MySQL DB Server has started";']

      containers:

        - name: usermgmt-restapp

          image: stacksimplify/kube-usermanagement-microservice:1.0.0

          resources:

            requests:

              memory: "128Mi"

              cpu: "500m"

            limits:

              memory: "500Mi"

              cpu: "1000m"

          ports:

            - containerPort: 8095

          env:

            - name: DB\_HOSTNAME

              value: "mysql"

            - name: DB\_PORT

              value: "3306"

            - name: DB\_NAME

              value: "usermgmt"

            - name: DB\_USERNAME

              value: "dbadmin"        # RDS DB Username is dbadmin

            - name: DB\_PASSWORD

              valueFrom:

                secretKeyRef:

                  name: mysql-db-password

                  key: db-password

          livenessProbe:

            exec:

              command:

                - /bin/sh

                - -c

                - nc -z localhost 8095

            initialDelaySeconds: 60

            periodSeconds: 10

          readinessProbe:

            httpGet:

              path: /usermgmt/health-status

              port: 8095

            initialDelaySeconds: 60

            periodSeconds: 10

**--- 04-Kubernetes-Secrets.yml**

apiVersion: v1

kind: Secret

metadata:

  name: mysql-db-password

  labels:

    runon: fargate

  namespace: ns-ums

type: Opaque

data:

  db-password: ZGJwYXNzd29yZDEx

**--- 05-UserManagement-NodePort-Service.yml**

apiVersion: v1

kind: Service

metadata:

  name: usermgmt-restapp-nodeport-service

  labels:

    app: usermgmt-restapp

    runon: fargate

  namespace: ns-ums

  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: /usermgmt/health-status

spec:

  type: NodePort

  selector:

    app: usermgmt-restapp

  ports:

    - port: 8095

      targetPort: 8095

**--- 07-ALB-Ingress-SSL-Redirect-with-ExternalDNS.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: ums-ingress-service

  labels:

    app: usermgmt-restapp

    runon: fargate

  namespace: ns-ums

  annotations:

    # Load Balancer Name

    alb.ingress.kubernetes.io/load-balancer-name: ums-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'

    # External DNS - For creating a Record Set in Route53

    external-dns.alpha.kubernetes.io/hostname: ums1101.stacksimplify.com

    # For Fargate

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

spec:

  rules:

    - http:

        paths:

          - path: /

            pathType: Prefix

            backend:

              service:

                name: usermgmt-restapp-nodeport-service

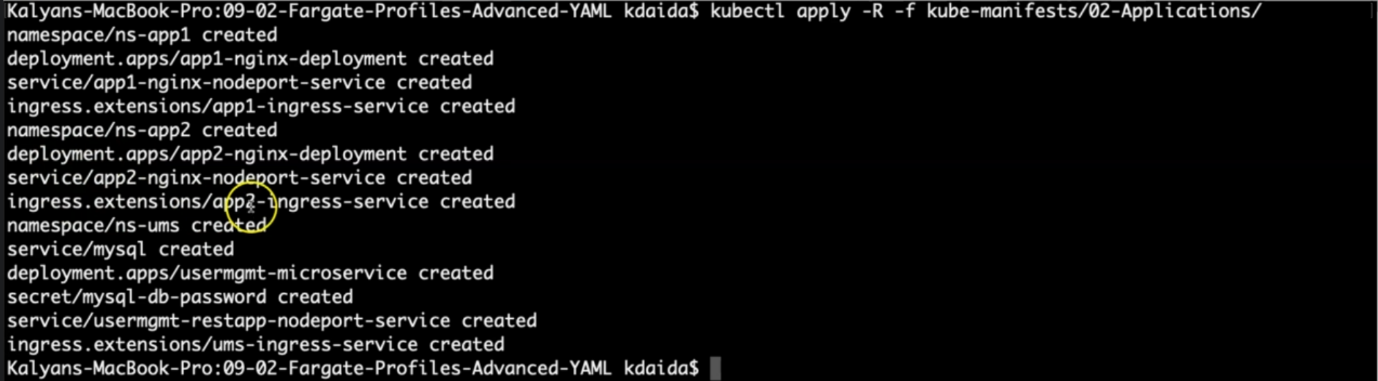
                port:

                  number: 8095

# 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.

**# Deploy Apps**

--- **kubectl apply -R -f kube-manifests/02-Applications/**

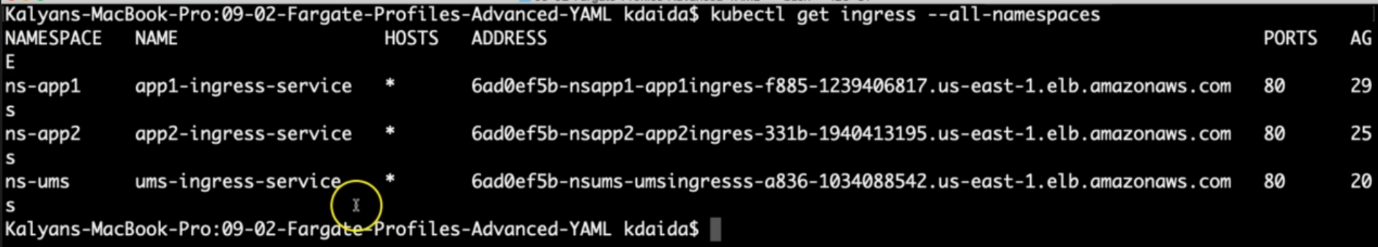


**Verify deployed Apps**

--- Verify using kubectl

**# Verify Ingress**

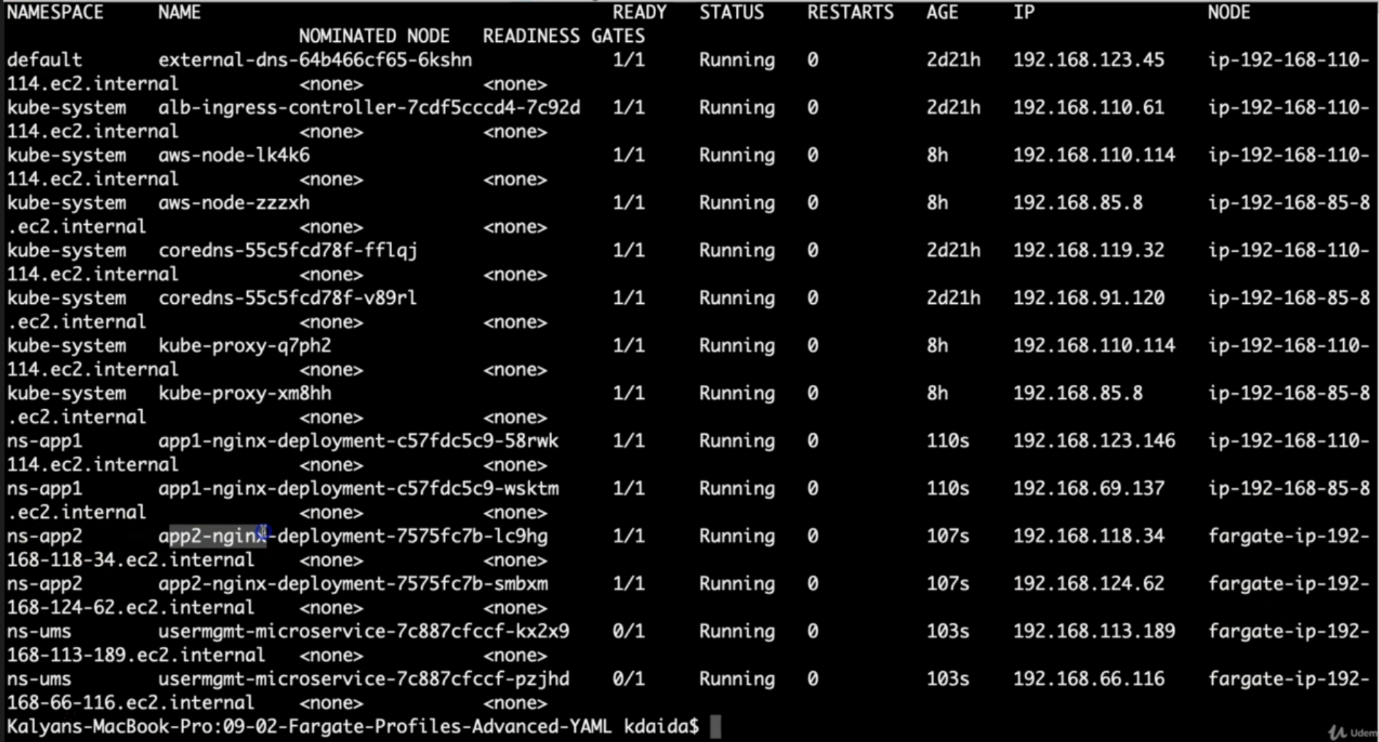
--- **kubectl get ingress --all-namespaces**



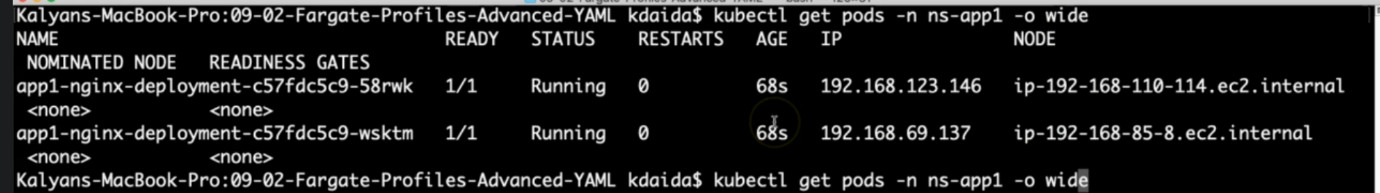
--- **note** - in all name spaces, the ingress got created or not.

**# Verify Pods**

--- **kubectl get pods --all-namespaces -o wide**



--- **kubectl get pods -n ns-app1 -o wide**

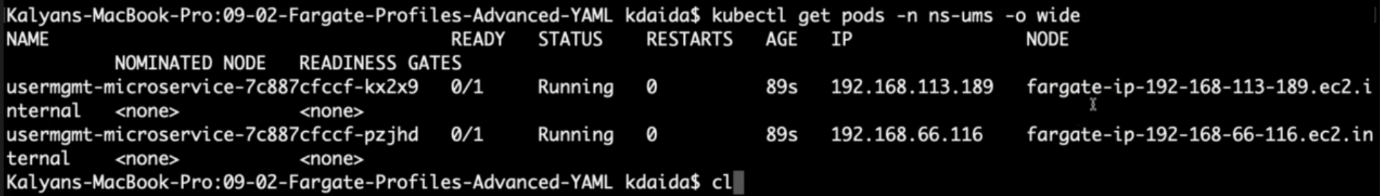


--- **kubectl get pods -n ns-app2 -o wide**



--- **note** – it is showing fargate nodes.

--- **kubectl get pods -n ns-ums -o wide**

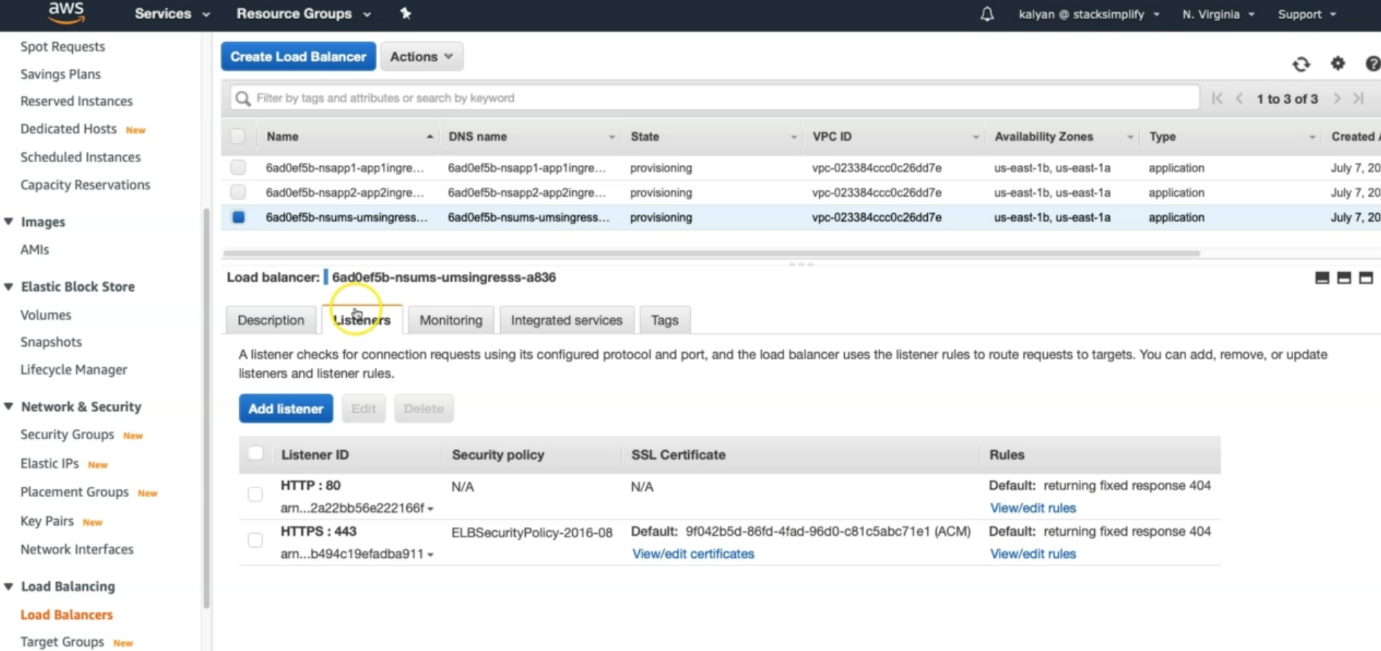


**# Verify Fargate Nodes**

--- **kubectl get nodes -o wide**

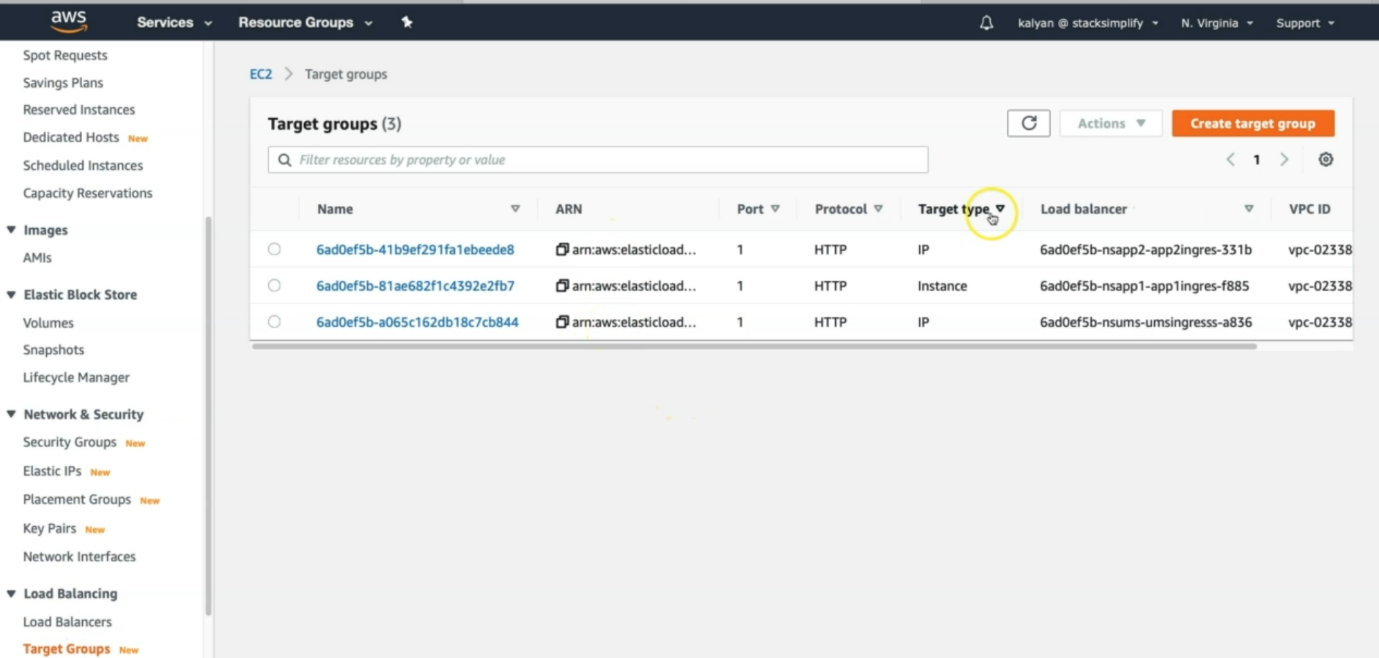
**Verify ALB & Target Groups**

--- Verify ALB Listeners, Rules.



--- Verify Target Groups

1. App1: Should use Target Type as instance
2. App2, UMS: Should use Target Type as ip



**Access Applications**

--- App1: http://app1.kubeoncloud.com/app1/index.html

--- App2: http://app2.kubeoncloud.com/app2/index.html

--- UMS Health Status Page: http://ums.kubeoncloud.com/usermgmt/health-status

--- UMS List Users: http://ums.kubeoncloud.com/usermgmt/users

**Delete Apps**

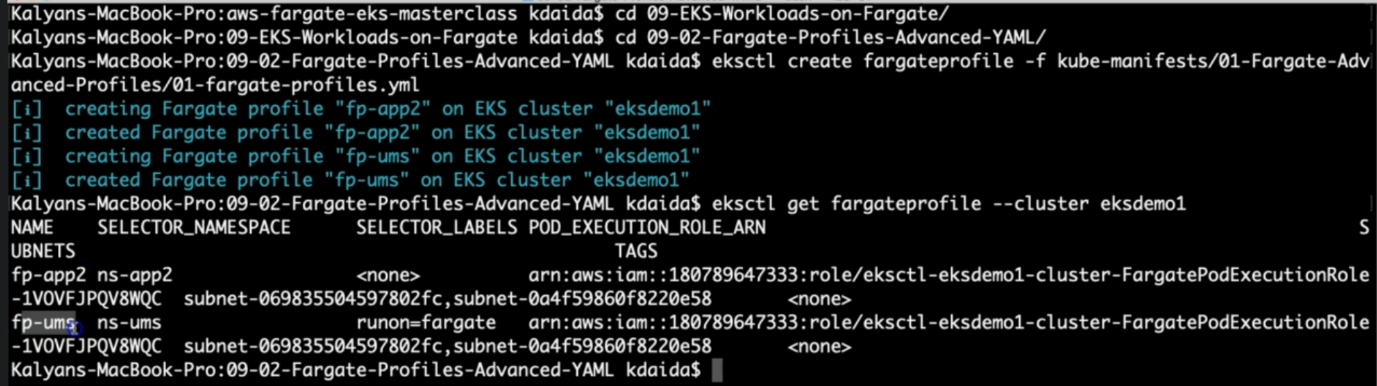
**# delete Apps**

--- **kubectl delete -R -f kube-manifests/02-Applications/**

**Delete Fargate Profile**

**# Get list of Fargate Profiles in a cluster**

--- **eksctl get fargateprofile --cluster eksdemo1**



**# Delete Fargate Profile**

--- **eksctl delete fargateprofile --cluster <cluster-name> --name <Fargate-Profile-Name> --wait**

--- **eksctl delete fargateprofile --cluster eksdemo1 --name fp-app2 --wait**

--- **eksctl delete fargateprofile --cluster eksdemo1 --name fp-ums --wait**

**Reference github Issue for ALB Ingress with Cross Namespaces**

--- kubernetes/kubernetes#17088 - <https://github.com/kubernetes/kubernetes/issues/17088>