**3: EKS with Classic Load Balancers Demo**

--- **note** – here we are going to deploy mysql external name service and our user management deployment. Also, we are going to write a classic load balancer manifest service. We will deploy all these three things in private subnets.

--- we will access our application using classic load balancer end point.

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/07-ELB-Classic-and-Network-LoadBalancers/07-02-Classic-LoadBalancer-CLB>

**Create AWS Classic Load Balancer Kubernetes Manifest & Deploy**

--- **04-ClassicLoadBalancer.yml**

apiVersion: v1

kind: Service

metadata:

  name: clb-usermgmt-restapp

  labels:

    app: usermgmt-restapp

spec:

  type: LoadBalancer # Regular k8s Service manifest with type as LoadBalancer

  selector:

    app: usermgmt-restapp

  ports:

  - port: 80

    targetPort: 8095

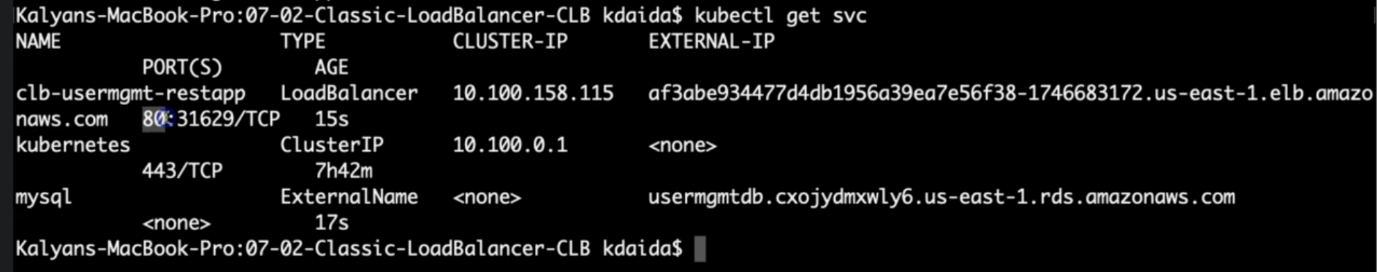
**Deploy all Manifest**

**# Deploy all manifests**

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

**# List Services (Verify newly created CLB Service)**

--- **kubectl get svc**



**# Verify Pods**

--- **kubectl get pods**

**Verify the deployment**

--- Verify if new CLB got created

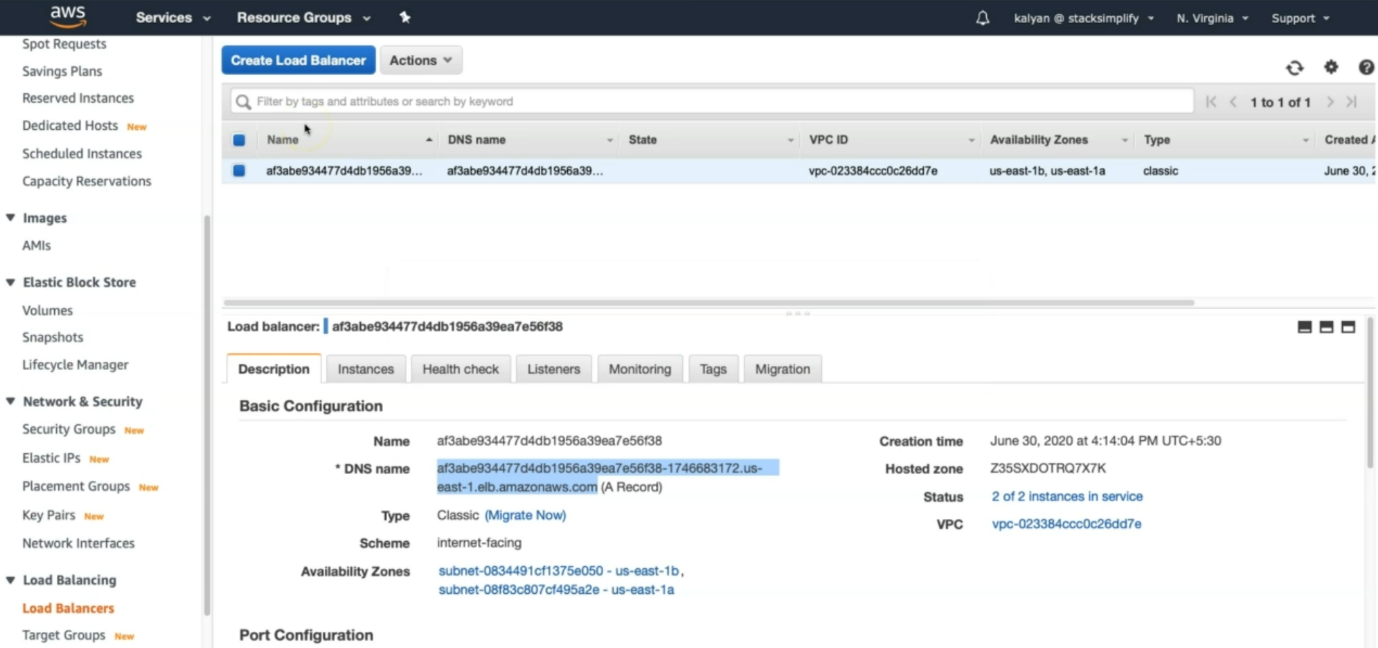
Go to Services -> EC2 -> Load Balancing -> Load Balancers

CLB should be created

Copy DNS Name (Example: a85ae6e4030aa4513bd200f08f1eb9cc-7f13b3acc1bcaaa2.elb.us-east-1.amazonaws.com)

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

Verify the health status, we should see active.



--- **note** – use the end point to access the application over the internet.

**Access Application**

**# Access Application**

--- http://<CLB-DNS-NAME>/usermgmt/health-status

**Clean Up**

**# Delete all Objects created**

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

**# Verify current Kubernetes Objects**

--- **kubectl get all**