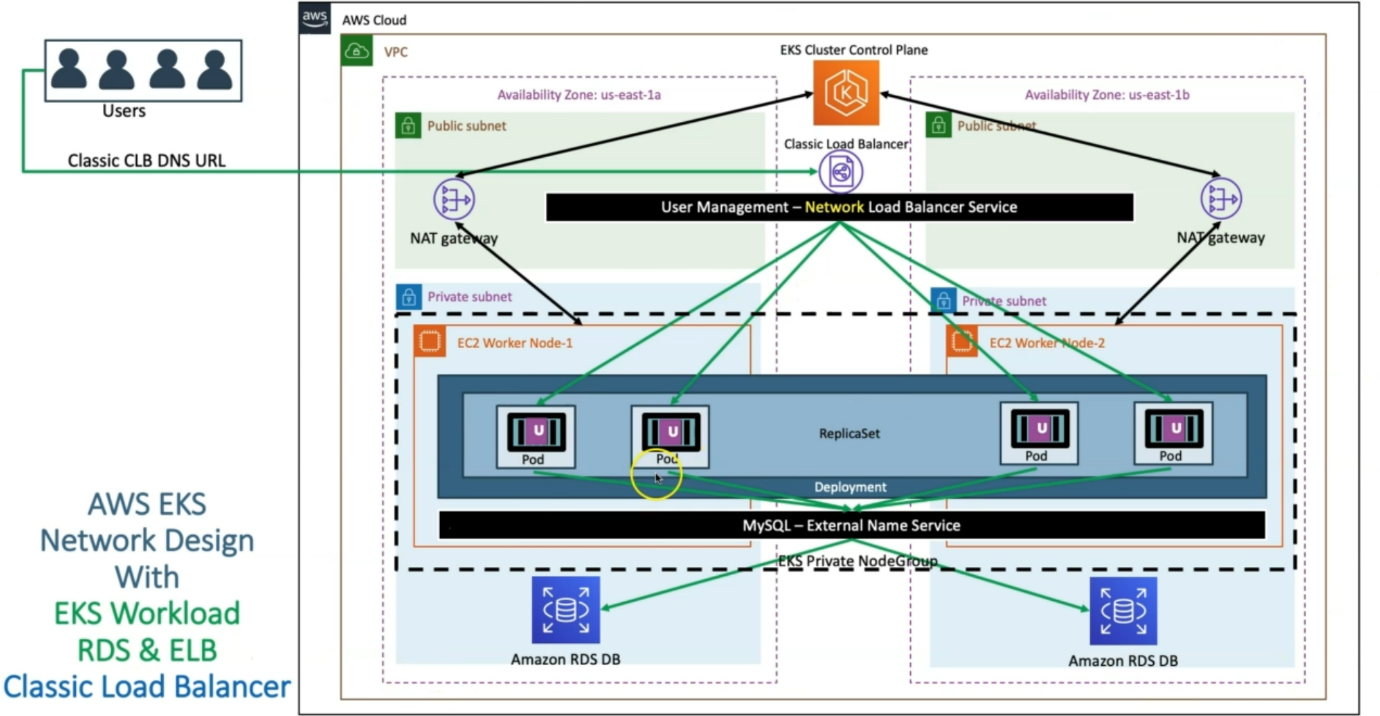
**4. EKS with Network Load Balancers Demo**

--- **note** – in this lecture, we are going to create network load balancer.



--- **note** – the entire architecture is going to be same and only change here is network load balancer instead of classic load balancer.

--- what is the network load balancer…?

The network load balancer is nothing but a tcp load balancer. Generally, tcp load balancer is provided for providing high availability to an application which does not need http or smtp server. Whoever wants tcp load balancer, we will provide network load balancer.

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

--- **04-NetworkLoadBalancer.yml**

apiVersion: v1

kind: Service

mtadata:

  name: nlb-usermgmt-restapp

  labels:

    app: usermgmt-restapp

  annotations:

    service.beta.kubernetes.io/aws-load-balancer-type: nlb    # To create Network Load Balancer

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 NLB 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.

**Access Application**

**# Access Application**

--- **http://<NLB-DNS-NAME>/usermgmt/health-status**

**Clean Up**

**# Delete all Objects created**

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

**# Verify current Kubernetes Objects**

--- **kubectl get all**