CKA Mock Exam - 3

Create a new service account with the name pvviewer. Grant this Service account access to list all PersistentVolumes in the cluster by creating an appropriate cluster role called pvviewer-role and ClusterRoleBinding called pvviewer-role-binding.

create a service account

```
kubectl create serviceaccount pvviewer

kubectl create clusterrole pvviewer-role --verb=list --resource=pv

kubectl create clusterrolebinding pvviewer-role-binding --clusterrole=pvviewer-role --serviceaccount=default:pvviewer

kubectl run pvviewer --restart=Never --image=redis --serviceaccount=pvviewer
```

List the InternalIP of all nodes of the cluster. Save the result to a file /root/node_ips

```
kubectl get nodes -o jsonpath='{.items[*].status.addresses[?(@.type=="InternalIP")].address}' > /root/node_ips
```

Create a pod called multi-pod with two containers. Container 1, name: alpha, image: nginx. Container 2: beta, image: busybox, command sleep 4800.

```
kubectl run beta --restart=Never --image=busybox --env=name=beta --dry-run -o yaml --command -- sleep 4800 > multi-pod.yaml
```

use this command to

```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: multi-pod
spec:
  containers:
  - command:
    - sleep
    - "4800"
    image: busybox
    env:
    - name: name
      value: beta
    name: beta
  - image: nginx
    env:
    - name: name
      value: alpha
    name: alpha
EOF
```

Create a Pod called non-root-pod , image: redis:alpine. runAsUser: 1000. fsGroup: 2000

```
kubectl run non-root-pod --restart=Never --image=redis:alpine --dry-run -o yaml > non-root-pod.yaml

cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: non-root-pod
spec:
  securityContext:
    runAsUser: 1000
    fsGroup: 2000
  containers:
  - image: redis:alpine
    name: non-root-pod
EOF
```

Create NetworkPolicy, by the name ingress-to-nptest that allows incoming connections to the service over port 80

```
cat << EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ingress-to-nptest
  namespace: default
spec:
  podSelector:
    matchLabels:
      run: np-test-1
  policyTypes:
  - Ingress
  ingress:
  - ports:
    - protocol: TCP
      port: 80
EOF

kubectl run test-np --restart=Never --image=busybox:1.28 --rm -i -- nc -z -v -w 2 np-test-service 80
```

use this command to test service

Taint the worker node node01 to be Unschedulable. Once done, create a pod called dev-redis, image redis:alpine to ensure workloads are not scheduled to this worker node. Finally, create a new pod called prod-redis and image redis:alpine with toleration to be scheduled on node01.

```
kubectl taint nodes node01 env_type=production:NoSchedule

kubectl run dev-redis --restart=Never --image=redis:alpine

cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: prod-redis
spec:
  containers:
  - name: prod-redis
    image: redis:alpine
  tolerations:
  - key: env_type
    value: production
    operator: Equal
    effect: NoSchedule
EOF
```

Create a pod called hr-pod in hr namespace belonging to the production environment and frontend tier .

```
kubectl create ns hr

kubectl run hr-pod -n hr --restart=Never --image=redis:alpine -l environment=production,tier=frontend
```

A kubeconfig file called super.kubeconfig has been created in /root. There is something wrong with the configuration. Troubleshoot and fix it.

```
kubectl cluster-info --kubeconfig=/root/super.kubeconfig

cp /root/.kube/config /root/super.kubeconfig
```

We have created a new deployment called nginx-deploy. scale the deployment to 3 replicas. Has the replica's increased? Troubleshoot the issue and fix it.

```
kubectl scale deploy nginx-deploy --replicas=3

grep contro1ler /etc/kubernetes/manifests/kube-controller-manager.yaml | wc -l
```

```
sed -i 's/kube-contro1ler-manager/kube-controller-manager/g' /etc/kubernetes/manifests/kube-controller-manager.yaml
```