

CKA Mock Exam - 1

Deploy a pod named nginx-pod using the nginx:alpine image.

```
kubectl run nginx-pod --restart=Never --image=nginx:alpine
```

Deploy a messaging pod using the redis:alpine image with the labels set to tier=msg.

```
kubectl run messaging --restart=Never --image=redis:alpine -l tier=msg
```

Create a namespace named apx-x9984574

```
kubectl create ns apx-x9984574
```

Get the list of nodes in JSON format and store it in a file at /opt/outputs/nodes-z3444kd9.json

```
kubectl get nodes -o json > /opt/outputs/nodes-z3444kd9.json
```

Create a service messaging-service to expose the messaging application within the cluster on port 6379.

```
kubectl expose pod messaging --name=messaging-service --port=6379 -l tier=msg
```

Create a deployment named hr-web-app using the image kodekloud/webapp-color with 2 replicas

```
kubectl run hr-web-app --image=kodekloud/webapp-color --replicas=2
```

Create a static pod named static-busybox on the master node that uses the busybox image and the command sleep 1000

```
kubectl run static-busybox --restart=Never --image=busybox --dry-run -o yaml --command -- sleep 1000 > /etc/kubernetes/manifests/static-busybox.yaml
```

Create a POD in the finance namespace named temp-bus with the image redis:alpine.

```
kubectl run temp-bus --restart=Never --image=redis:alpine -n finance
```

A new application orange is deployed. There is something wrong with it. Identify and fix the issue.

```
kubectl edit pod/orange
```

```
    update the sleep command
```

Expose the hr-web-app as service hr-web-app-service application on port 30082 on the nodes on the cluster

```
kubectl expose deploy hr-web-app --name=hr-web-app-service --port=8080 --type=NodePort
```

```
kubectl edit svc hr-web-app-service
```

```
    update nodePort to: 30082
```

Use JSON PATHquery to retrieve the osImages of all the nodes and store it in a file /opt/outputs/nodes_os_x43kj56.txt

```
kubectl get nodes -o jsonpath={.items[*].status.nodeInfo.osImage} > /opt/outputs/nodes_os_x43kj56.txt
```

Create a Persistent Volume with the given specification.

```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-analytics
spec:
  storageClassName: manual
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 100Mi
```

```
hostPath:
  path: "/pv/data-analytics"
EOF
```