

DUMPSBOSS.COM

Certified Kubernetes Administrator (CKA)

Linux Foundation CKA

Version Demo

Total Demo Questions: 10

Total Premium Questions: 129

Buy Premium PDF

<https://dumpsboss.com>

support@dumpsboss.com

dumpsboss.com

QUESTION NO: 1 - (SIMULATION)

Create 5 nginx pods in which two of them is labeled env=prod and three of them is labeled env=dev

ANSWER: See Explanation Below For Answer**Explanation:**

```
kubectl run nginx-dev1 --image=nginx --restart=Never -- labels=env=dev
kubectl run nginx-dev2 --image=nginx --restart=Never -- labels=env=dev
kubectl run nginx-dev3 --image=nginx --restart=Never --
labels=env=dev
kubectl run nginx-prod1 --image=nginx --restart=Never -- labels=env=prod
kubectl run nginx-prod2 --image=nginx --restart=Never -- labels=env=prod
```

QUESTION NO: 2 - (SIMULATION)

Get list of persistent volumes and persistent volume claim in the cluster

ANSWER: See Explanation Below For Answer**Explanation:**

```
kubectl get pv
kubectl get pvc
```

QUESTION NO: 3 - (SIMULATION)

Create a busybox pod and add "sleep 3600" command

ANSWER: See Explanation Below For Answer**Explanation:**

```
kubectl run busybox --image=busybox --restart=Never -- /bin/sh -c "sleep 3600"
```

QUESTION NO: 4 - (SIMULATION)

Scale down the deployment to 1 replica

ANSWER: See Explanation Below For Answer

Explanation:

```
kubectl scale deployment webapp --replicas=1//Verify kubectl get deploy
```

```
kubectl get po,rs
```

QUESTION NO: 5 - (SIMULATION)

Create a pod that having 3 containers in it? (Multi-Container)

ANSWER: See Explanation Below For Answer

Explanation:

image=nginx, image=redis, image=consul Name nginx container as "nginx-container" Name redis container as "redis-container" Name consul container as "consul-container"

Create a pod manifest file for a container and append container section for rest of the images

```
kubectl run multi-container --generator=run-pod/v1 --image=nginx -- dry-run -o yaml > multi-container.yaml
```

then

```
vim multi-container.yaml apiVersion: v1
```

```
kind: Pod metadata:
```

```
labels:
```

```
run: multi-container name: multi-container spec:
```

```
containers:
```

```
- image: nginx
```

```
name: nginx-container
```

```
- image: redis
```

```
name: redis-container
```

```
- image: consul
```

```
name: consul-container restartPolicy: Always
```

QUESTION NO: 6 - (SIMULATION)

Check the image version in pod without the describe command

ANSWER: See Explanation Below For Answer

Explanation:

```
kubectl get po nginx -o
```

```
jsonpath='{.spec.containers[].image}{"\n"}'
```

QUESTION NO: 7 - (SIMULATION)

Create an nginx pod with container Port 80 and it should only receive traffic only it checks the endpoint / on port 80 and verify and delete the pod.

ANSWER: See Explanation Below For Answer

Explanation:

```
kubectl run nginx --image=nginx --restart=Never --port=80 -- dry-run -o yaml > nginx-pod.yaml
```

```
// add the readinessProbe section and create vim nginx-pod.yaml
```

```
apiVersion: v1 kind: Pod metadata: labels:
```

```
run: nginx name: nginx spec: containers:
```

```
- image: nginx name: nginx ports:
```

```
- containerPort: 80 readinessProbe: httpGet:
```

```
path: / port: 80
```

```
restartPolicy: Never
```

```
kubectl apply -f nginx-pod.yaml
```

```
// verify
```

```
kubectl describe pod nginx | grep -i readiness kubectl delete po nginx
```

QUESTION NO: 8 - (SIMULATION)

Make the node schedulable by uncordon the node

ANSWER: See Explanation Below For Answer

Explanation:

```
kubectl uncordon node-1
```

```
//verify kubectl get no
```

QUESTION NO: 9 - (SIMULATION)

Create a configmap called cfgvolume with values var1=val1,

var2=val2 and create an nginx pod with volume nginx-volume which reads data from this configmap cfgvolume and put it on the path

/etc/cfg

ANSWER: See Explanation Below For Answer**Explanation:**

```
// first create a configmap cfgvolume
```

```
kubectl create cm cfgvolume --from-literal=var1=val1 --fromliteral=var2=val2
```

```
// verify the configmap
```

```
kubectl describe cm cfgvolume
```

```
// create the config map
```

```
kubectl create -f nginx-volume.yml vim nginx-configmap-pod.yaml apiVersion: v1
```

```
kind: Pod metadata: labels: run: nginx
```

```
name: nginx spec:
```

```
volumes:
```

```
- name: nginx-volume configMap:
```

```
name: cfgvolume containers:
```

```
- image: nginx name: nginx volumeMounts:
```

```
- name: nginx-volume mountPath: /etc/cfg restartPolicy: Always
```

```
k kubectl apply -f nginx-configmap-pod.yaml
```

```
/ // Verify
```

```
// exec into the pod
```

```
kubectl exec -it nginx -- /bin/sh
```

```
// check the path cd /etc/cfg
```