

## CKA<sup>Q&As</sup>

Certified Kubernetes Administrator (CKA) Program

### Pass Linux Foundation CKA Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.leads4pass.com/cka.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



## QUESTION 1

**SIMULATION** Create a deployment as follows: Name: nginx-random Exposed via a service nginx-random Ensure that the service and pod are accessible via their respective DNS records The container(s) within any pod(s) running as a part of this deployment should use the nginx Image Next, use the utility nslookup to look up the DNS records of the service and pod and write the output to /opt/KUNW00601/service.dns and /opt/ KUNW00601/pod.dns respectively.

Correct Answer: Check the answer in explanation.

Solution:

```

Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~#
root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yaml

```

```

Readme Web Terminal THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
  name: busybox1
  labels:
    name: busybox
spec:
  containers:
  - image: busybox:1.28
    command:
    - sleep
    - "3600"
    name: busybox

```

```

Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yaml
root@node-1:~# k create -f dns.yaml
pod/busybox1 created
root@node-1:~# k get po -o wide | grep nginx-random
nginx-random-6d5766bbdc-ptzv2 1/1 Running 0 103s 10.244.2.16 k8s-node-1 <none> <none>
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name: nginx-random
Address 1: 10.111.37.132 nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random > /opt/KUNW00601/service.dns
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name: 10-244-2-16.default.pod
Address 1: 10.244.2.16 10-244-2-16.nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod > /opt/KUNW00601/pod.dns

```

## QUESTION 2

Create a busybox pod and add "sleep 3600" command

Correct Answer: Check the answer in explanation.

Solution

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

---

## QUESTION 3

Get list of all the pods showing name and namespace with a jsonpath expression.

Correct Answer: Check the answer in explanation.

```
kubectl get pods -o=jsonpath="{.items[*][\metadata.name\ , \metadata.namespace\]}"
```

---

## QUESTION 4

List all the pods showing name and namespace with a json path expression

Correct Answer: Check the answer in explanation.

Solution

```
kubectl get pods -o=jsonpath="{.items[*][\metadata.name\ , \metadata.namespace\]}"
```

---

## QUESTION 5

### SIMULATION

Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadWriteMany. The type of volume is hostPath and its location is /srv/app-data.

Correct Answer: Check the answer in explanation.

Persistent Volume A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a cluster-level resource like nodes, which don't belong to any namespace. It is provisioned by the administrator and has a particular file size. This way, a developer deploying their app on Kubernetes need not know the underlying infrastructure. When the developer needs a certain amount of persistent storage for their application, the system administrator configures the cluster so that they consume the PersistentVolume provisioned in an easy way. Creating Persistent Volume kind: PersistentVolume apiVersion: v1 metadata: name:app-data spec: capacity: # defines the capacity of PV we are creating storage: 2Gi #the amount of storage we are trying to claim accessModes: # defines the rights of the volume we are creating

-ReadWriteMany

hostPath:



Save and create the pvc njerry191@cloudshell:~ (extreme-clone-2654111)\$ kubectl create -f app-data.yaml  
persistentvolumeclaim/app-data created

3.

View the pvc

4.

Let's see what has changed in the pv we had initially created.

```
njerry191@cloudshell:~ (extreme-clone-2654111)$ kubectl get pvc
NAME:      STATUS   VOLUME:  CAPACITY   ACCESS MODES   STORAGECLASS
pv         Bound    pv       512m      RWX            shared
```

```
njerry191@cloudshell:~ (extreme-clone-2654111)$ kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM          STORAGECLASS   REASON   AGE
pv        512m      RWX            Retain           Bound   default/pv    shared        16m
```

Our status has now changed from available to bound.

5. Create a new pod named myapp with image nginx that will be used to Mount the Persistent Volume Claim with the path /var/app/config. Mounting a Claim apiVersion: v1 kind: Pod metadata: creationTimestamp: null name: app-data spec: volumes:

-name: configpvc persistentVolumeClaim: claimName: app-data containers:

-image: nginx name: app volumeMounts:

-mountPath: "/srv/app-data " name: configpvc

[CKA PDF Dumps](#)

[CKA VCE Dumps](#)

[CKA Exam Questions](#)