



**Free Questions for CKA by go4braindumps**

**Shared by Valenzuela on 12-12-2023**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

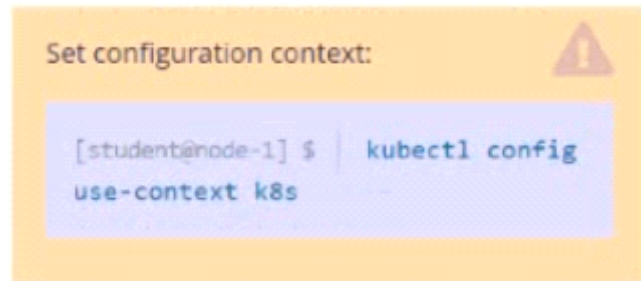
# Question 1


---

Question Type: MultipleChoice

---

Task Weight: 4%



```
Set configuration context:   
[student@node-1] $ kubectl config  
use-context k8s
```

Task

Scale the deployment webserver to 3 pods.

Options:

---

A- Explanation:

Solution:

```
student@node-1:~$ kubectl scale deployment/webserver --replicas=3  
deployment.apps/webserver scaled  
student@node-1:~$ kubectl scale deploy webserver --replicas=3
```

**Answer:**

---

A


## Question 2

---

**Question Type:** MultipleChoice

---

Score:7%

```
Set configuration context:   
[student@node-1] $ | kube  
ctl config use-context o  
k8s
```

Task

Create a new PersistentVolumeClaim

\* Name: pv-volume

\* Class: csi-hostpath-sc

\* Capacity: 10Mi

Create a new Pod which mounts the PersistentVolumeClaim as a volume:

\* Name: web-server

\* Image: nginx

\* Mount path: /usr/share/nginx/html

Configure the new Pod to have ReadWriteOnce access on the volume.

Finally, using kubectl edit or kubectl patch expand the PersistentVolumeClaim to a capacity of 70Mi and record that change.

## Options:

---

**A-** Explanation:

Solution:

```
vi pvc.yaml
```

```
storageclass pvc
```

```
apiVersion: v1
```

```
kind: PersistentVolumeClaim
```

```
metadata:
```

```
name: pv-volume
```

```
spec:
```

```
accessModes:
```

```
- ReadWriteOnce
```

```
volumeMode: Filesystem
resources:
requests:
storage: 10Mi
storageClassName: csi-hostpath-sc
# vi pod-pvc.yaml
apiVersion: v1
kind: Pod
metadata:
name: web-server
spec:
containers:
- name: web-server
image: nginx
volumeMounts:
- mountPath: '/usr/share/nginx/html'
name: my-volume
volumes:
- name: my-volume
persistentVolumeClaim:
claimName: pv-volume
# craete
kubectl create -f pod-pvc.yaml
#edit
kubectl edit pvc pv-volume --record
```

**Answer:**

---

A


## Question 3

---

**Question Type:** MultipleChoice

---

Score: 4%

```
Set configuration context:   
[student@node-1] $ | kube  
ctl config use-context h  
k8s
```

Task

Create a persistent volume with name app-data , of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-data .

## Options:

---

**A-** Explanation:

Solution:

```
#vi pv.yaml
```

```
apiVersion: v1
```

```
kind: PersistentVolume
```

```
metadata:
```

```
name: app-config
```

```
spec:
```

```
capacity:
```

```
storage: 1Gi
```

```
accessModes:
```

```
- ReadOnlyMany
```

```
hostPath:
```

```
path: /srv/app-config
```

```
#
```

```
kubectl create -f pv.yaml
```

## Answer:

---

A


## Question 4

---

## Question Type: MultipleChoice

---

Score: 4%

```
Set configuration context:   
[student@node-1] $ | kube  
ctl config use-context k  
8s
```

Task

Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified): nginx + redis + memcached .

### Options:

---

**A-** Explanation:

Solution:

```
kubectl run kucc8 --image=nginx --dry-run -o yaml > kucc8.yaml
```

```
# vi kucc8.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```



```
metadata:  
creationTimestamp: null  
name: kucc8  
spec:  
containers:  
- image: nginx  
name: nginx  
- image: redis  
name: redis  
- image: memcached  
name: memcached  
- image: consul  
name: consul  
#  
kubectl create -f kucc8.yaml  
#12.07
```

**Answer:**

---

A


## Question 5

---

**Question Type:** MultipleChoice

---

Score: 4%

```
Set configuration context:   
[student@node-1] $ | kube  
ctl config use-context k  
8s
```

Task

Check to see how many nodes are ready (not including nodes tainted NoSchedule ) and write the number to /opt/KUSC00402/kusc00402.txt.

## Options:

---

**A-** Explanation:

Solution:

```
kubectl describe nodes | grep ready|wc -l
```

```
kubectl describe nodes | grep -i taint | grep -i noschedule |wc -l
```

```
echo 3 > /opt/KUSC00402/kusc00402.txt
```

```
#
```

```
kubectl get node | grep -i ready |wc -l
```

```
# taintsnoSchedule
kubectl describe nodes | grep -i taints | grep -i noschedule | wc -l
#
echo 2 > /opt/KUSC00402/kusc00402.txt
```

**Answer:**

---

A


## Question 6

---

**Question Type:** MultipleChoice

---

Score: 4%

```
Set configuration context: 
[student@node-1] $ | kube
ctl config use-context k
8s
```

## Task

Schedule a pod as follows:

\* Name: nginx-kusc00401

\* Image: nginx

\* Node selector: disk=ssd

## Options:

---

**A-** Explanation:

Solution:

#yaml

apiVersion: v1

kind: Pod

metadata:

name: nginx-kusc00401

spec:

containers:

- name: nginx

image: nginx

imagePullPolicy: IfNotPresent

nodeSelector:

disk: spinning

```
#  
kubectl create -f node-select.yaml
```

**Answer:**

---

A


## Question 7

---

**Question Type:** MultipleChoice

---

Score: 4%

```
Set configuration context:   
[student@node-1] $ | kubectl config use-context k8s
```

Task

Scale the deployment presentation to 6 pods.

### Options:

---

A- Explanation:

Solution:

```
kubectl get deployment
```

```
kubectl scale deployment.apps/presentation --replicas=6
```

### Answer:

---

A

## Question 8

---

**Question Type:** MultipleChoice

---

Score: 7%

Set configuration context:




```
[student@node-1] $ | kube  
ctl config use-context k  
8s
```

## Task

Create a new nginx Ingress resource as follows:

- \* Name: ping
- \* Namespace: ing-internal
- \* Exposing service hi on path /hi using service port 5678

The availability of service hi   
can be checked using the  
following command, which  
should return hi :

```
[student@node-1] $ | curl  
-kL <INTERNAL_IP>/hi
```

## Options:

---

**A-** Explanation:

Solution:

```
vi ingress.yaml
```

```
#
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
name: ping
```

```
namespace: ing-internal
```

```
spec:
```

```
rules:
```

```
- http:
```



```
paths:  
- path: /hi  
pathType: Prefix  
backend:  
service:  
name: hi  
port:  
number: 5678  
#  
kubectl create -f ingress.yaml
```

**Answer:**

---

A

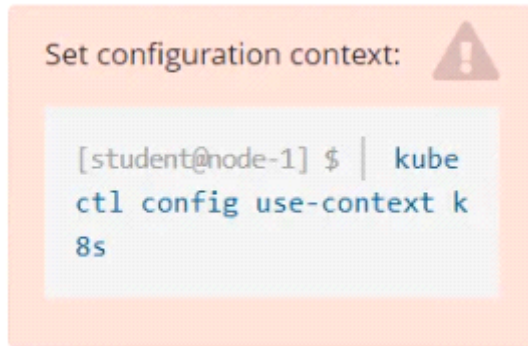
## Question 9

---

**Question Type:** MultipleChoice

---

Score: 7%



## Task

Reconfigure the existing deployment front-end and add a port specification named http exposing port 80/tcp of the existing container nginx.

Create a new service named front-end-svc exposing the container port http.

Configure the new service to also expose the individual Pods via a NodePort on the nodes on which they are scheduled.

## Options:

---

**A-** Explanation:

Solution:

```
kubectl get deploy front-end
```

```
kubectl edit deploy front-end -o yaml
```

```
#port specification named http
```

```
#service.yaml
```

```
apiVersion: v1
```

```
kind: Service
metadata:
name: front-end-svc
labels:
app: nginx
spec:
ports:
- port: 80
protocol: tcp
name: http
selector:
app: nginx
type: NodePort
# kubectl create -f service.yaml
# kubectl get svc
# port specification named http
kubectl expose deployment front-end --name=front-end-svc --port=80 --target-port=80 --type=NodePort
```

## Answer:

---

A


## Question 10

---

**Question Type: MultipleChoice**

---

Score: 7%

```
Set configuration context:   
[student@node-1] $ | kubeCopy  
ctl config use-context h  
k8s
```

Task

Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace echo. Ensure that the new NetworkPolicy allows Pods in namespace my-app to connect to port 9000 of Pods in namespace echo.

Further ensure that the new NetworkPolicy:

- \* does not allow access to Pods, which don't listen on port 9000
- \* does not allow access from Pods, which are not in namespace my-app

**Options:**

---

**A-** Explanation:

Solution:

```
#network.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-port-from-namespace
  namespace: internal
spec:
  podSelector:
    matchLabels: {
    }
  policyTypes:
  - Ingress
  ingress:
  - from:
  - podSelector: {
  }
  ports:
  - protocol: TCP
    port: 8080
#spec.podSelector namespace pod
kubectl create -f network.yaml
```

**Answer:**

---

A

**To Get Premium Files for CKA Visit**

<https://www.p2pexams.com/products/cka>

**For More Free Questions Visit**

<https://www.p2pexams.com/linux-foundation/pdf/cka>

