



CKA^{Q&As}

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.passapply.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

Create a nginx pod with label env=test in engineering namespace .

Correct Answer: Check the answer in explanation.

```
kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml > nginx-pod.yaml  
kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml | kubectl create -n engineering -f YAML File:
```

```
apiVersion: v1 kind: Pod metadata: name: nginx namespace: engineering labels: env: test spec: containers:
```

```
-name: nginx image: nginx imagePullPolicy: IfNotPresent restartPolicy: Never
```

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

QUESTION 2

Create a namespace called 'development' and a pod with image nginx called nginx on this namespace.

Correct Answer: Check the answer in explanation.

```
kubectl create namespace development kubectl run nginx --image=nginx --restart=Never -n development
```

QUESTION 3

Get list of all pods in all namespaces and write it to file "/opt/pods-list.yaml"

Correct Answer: Check the answer in explanation.

```
kubectl get po -all-namespaces > /opt/pods-list.yaml
```

QUESTION 4

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:

path: "/srv/app-data" # path to which we are creating the volume Challenge

Create a Persistent Volume named app-data, with access mode ReadWriteMany, storage classname shared, 2Gi of storage capacity and the host path /srv/app-data.



```
Readme Web Terminal THE LINUX FOUNDATION

authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
syncFrequency: 0s
volumeStatsAggPeriod: 0s
:wg
```

```
Readme Web Terminal THE LINUX FOUNDATION

https://microk8s.io/ has docs and details.

4 packages can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@bk8s-master-0:~$ sudo -i
root@bk8s-master-0:~# vim /var/lib/kubelet/config.yaml
root@bk8s-master-0:~# systemctl restart kubelet
root@bk8s-master-0:~# systemctl enable kubelet
root@bk8s-master-0:~# kubectl get nodes

NAME           STATUS    ROLES    AGE   VERSION
bk8s-master-0  Ready    master   77d   v1.18.2
bk8s-node-0    Ready    <none>   77d   v1.18.2
root@bk8s-master-0:~#
root@bk8s-master-0:~# exit
logout
student@bk8s-master-0:~$ exit
logout
Connection to 10.250.4.77 closed.
root@node-1:~#
```




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

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM  STORAGECLASS  REASON  AGE
app-data  2Gi      RWX           Retain          Available  shared  shared        31s
```

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pvc
NAME      STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS
pv        Bound   pv      512m     RWX           shared
```

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

QUESTION 5

CORRECT TEXT



```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
big-corp-app                        1/1     Running   0           6h13m
foo                                  1/1     Running   0           6h13m
front-end-6bc87b9748-n7v8h         1/1     Running   0           18m
front-end-6bc87b9748-zmb8g         1/1     Running   0           18m
kucc8                                2/2     Running   0           10m
nginx-kusc00401                     1/1     Running   0           13m
overloaded-cpu-98b9se               1/1     Running   0           6h12m
overloaded-cpu-ab2d3s               1/1     Running   0           6h12m
overloaded-cpu-kipb9a               1/1     Running   0           6h12m
presentation-684cd7ccb6-4gf56      1/1     Running   0           6h16m
presentation-684cd7ccb6-6zjls      1/1     Running   0           15m
presentation-684cd7ccb6-vshxj      1/1     Running   0           6h16m
student@node-1:~$ kubectl logs f
Error from server (NotFound): pods "f" not found
student@node-1:~$ kubectl logs foo
Mon Apr 25 09:28:09 UTC 2022 - INFO - application started
Mon Apr 25 09:28:09 UTC 2022 - WARN - new version available
Mon Apr 25 09:28:09 UTC 2022 - INFO - Listening on port 9432
Mon Apr 25 09:28:09 UTC 2022 - DEBUG - configuration loaded
Mon Apr 25 09:28:09 UTC 2022 - WARN - primary DNS server is down
Mon Apr 25 09:28:09 UTC 2022 - ERROR - file-not-found
Mon Apr 25 09:28:09 UTC 2022 - INFO - exiting application
student@node-1:~$ kubectl logs foo | grep "file-not-found"
Mon Apr 25 09:28:09 UTC 2022 - ERROR - file-not-found
student@node-1:~$
```

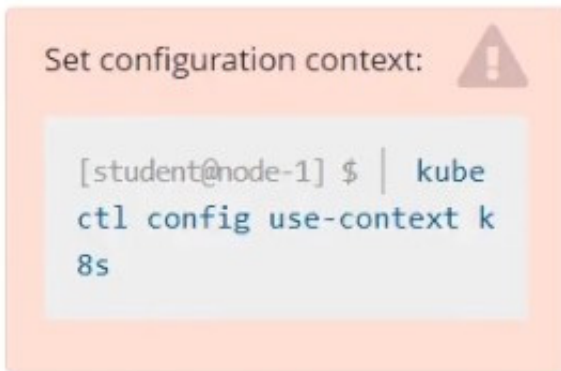
```
student@node-1:~$ kubectl logs foo | grep "file-not-found"
Mon Apr 25 09:28:09 UTC 2022 - ERROR - file-not-found
student@node-1:~$
student@node-1:~$
student@node-1:~$ kubectl logs foo | grep "file-not-found" > /opt/KUTR00101/foo
student@node-1:~$ cat /opt/KUTR00101/foo
Mon Apr 25 09:28:09 UTC 2022 - ERROR - file-not-found
student@node-1:~$
```

Context An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e.g. `kubectl logs`). Adding a streaming sidecar container is a good and common way to accomplish this requirement. Task

Add a sidecar container named `sidecar`, using the `busybox` Image, to the existing Pod `big-corp-app`. The new sidecar container has to run the following command:

```
/bin/sh -c tail -n+1 -f /var/log/big-corp-app.log
```

Use a Volume, mounted at `/var/log`, to make the log file `big-corp-app.log` available to the sidecar container.



Correct Answer: Check the answer in explanation.

```
# kubectl get pod big-corp-app -o yaml # apiVersion: v1 kind: Pod metadata: name: big-corp-app spec: containers:
```

```
-name: big-corp-app
```

```
image: busybox
```

```
args:
```

```
-/bin/sh
```

```
- -c
```

```
->
```

```
i=0;
```

```
while true;
```

```
do
```

```
echo "$(date) INFO $i" >> /var/log/big-corp-app.log; i=$((i+1));
```

```
sleep 1;
```

```
done
```

```
volumeMounts:
```

```
-
```

```
name: logs
```

```
mountPath: /var/log
```

```
-
```

```
name: count-log-1
```

```
image: busybox
```

```
args: [/bin/sh, -c, \\'tail -n+1 -f /var/log/big-corp-app.log\\'] volumeMounts:
```




```
-  
name: logs  
mountPath: /var/log  
volumes:  
-  
name: logs  
emptyDir: {  
}  
# kubectl logs big-corp-app -c count-log-1
```

[Latest CKA Dumps](#)

[CKA Practice Test](#)

[CKA Study Guide](#)