



CKAD^{Q&As}

Certified Kubernetes Application Developer (CKAD) Program

Pass Linux Foundation CKAD Exam with 100% Guarantee

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

<https://www.pass4itsure.com/ckad.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****CORRECT TEXT****Context**

A project that you are working on has a requirement for persistent data to be available.

Task

To facilitate this, perform the following tasks:

1.

Create a file on node sk8s-node-0 at /opt/KDSP00101/data/index.html with the content Acct=Finance

2.

Create a PersistentVolume named task-pv-volume using hostPath and allocate 1Gi to it, specifying that the volume is at /opt/KDSP00101/data on the cluster's node.

The configuration should specify the access mode of ReadWriteOnce. It should define the StorageClass name exam for the PersistentVolume, which will be used to bind PersistentVolumeClaim requests to this PersistentVolume.

1.

Create a PersistentVolumeClaim named task-pv-claim that requests a volume of at least 100Mi and specifies an access mode of ReadWriteOnce



2.

Create a pod that uses the PersistentVolumeClaim as a volume with a label app: my-storage-app mounting the resulting volume to a mountPath /usr/share/nginx/html inside the pod



You can access `sk8s-node-0` by 
issuing the following
command:

```
[student@node-1] $ | ssh sk8  
s-node-0
```

Ensure that you return to the 
base node (with hostname
`node-1`) once you have completed
your work on `sk8s-node-0`  copy

A. Please check explanations

B. Place Holder

Correct Answer: A

[Readme](#) [Web Terminal](#)THE **LINUX** FOUNDATION

```
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$
```

[Readme](#) [Web Terminal](#)THE **LINUX** FOUNDATION

```
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Fri Oct 9 08:52:09 UTC 2020

System load: 2.02          Users logged in: 0
Usage of /: 10.3% of 242.29GB IP address for eth0: 10.250.3.115
Memory usage: 2%          IP address for docker0: 172.17.0.1
Swap usage: 0%            IP address for cni0: 10.244.1.1
Processes: 38

* Kubernetes 1.19 is out! Get it in one command with:

    sudo snap install microk8s --channel=1.19 --classic

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

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

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@sk8s-node-0:~$
```

[Readme](#) [Web Terminal](#)THE **LINUX** FOUNDATION

```
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml
```



```
THE LINUX FOUNDATION

apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: storage
  hostPath:
    path: /opt/KDSP00101/data
    type: Directory
```

```
THE LINUX FOUNDATION

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: storage
```

```
THE LINUX FOUNDATION

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: storage
```

```
student@sk8s-node-01:~$ kubectl create -f pv.yml
persistentvolume/task-pv-volume created
student@sk8s-node-01:~$ kubectl create -f pvc.yml
persistentvolumeclaim/task-pv-claim created
student@sk8s-node-01:~$ kubectl get pv
NAME                CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM                STORAGECLASS  AGE
task-pv-volume      1Gi       RWO           Retain          Bound   default/task-pv-claim  storage       11s
student@sk8s-node-01:~$ kubectl get pvc
NAME                STATUS  VOLUME              CAPACITY  ACCESS MODES  STORAGECLASS  AGE
task-pv-claim       Bound   task-pv-volume      1Gi       RWO           storage       9s
student@sk8s-node-01:~$ vim pod.yml
```

```
THE LINUX FOUNDATION

apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: my-storage-app
spec:
  containers:
    - name: myfrontend
      image: nginx
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: mypod
      volumes:
        - name: mypod
          persistentVolumeClaim:
            claimName: task-pv-claim
```

```
student@sk8s-node-01:~$ kubectl create -f pod.yml
pod/mypod created
student@sk8s-node-01:~$ kubectl get
```

```
THE LINUX FOUNDATION

student@sk8s-node-01:~$ kubectl get pods
NAME    READY  STATUS             RESTARTS  AGE
mypod   0/1    ContainerCreating   0          4s
student@sk8s-node-01:~$ kubectl get pods
NAME    READY  STATUS             RESTARTS  AGE
mypod   0/1    ContainerCreating   0          8s
student@sk8s-node-01:~$ kubectl get pods
NAME    READY  STATUS             RESTARTS  AGE
mypod   1/1    Running            0         10s
student@sk8s-node-01:~$ logout
Connection to 10.250.3.115 closed.
student@node-1:~$
```



QUESTION 2

CORRECT TEXT



Context

It is always useful to look at the resources your applications are consuming in a cluster.

Task

From the pods running in namespace cpu-stress, write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030/pod.txt, which has already been created.

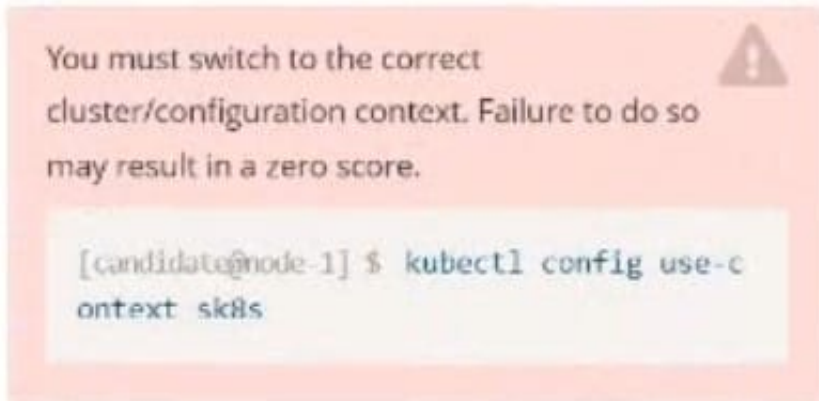
A. Please check explanations

B. Place Holder

Correct Answer: A

QUESTION 3

CORRECT TEXT



Context

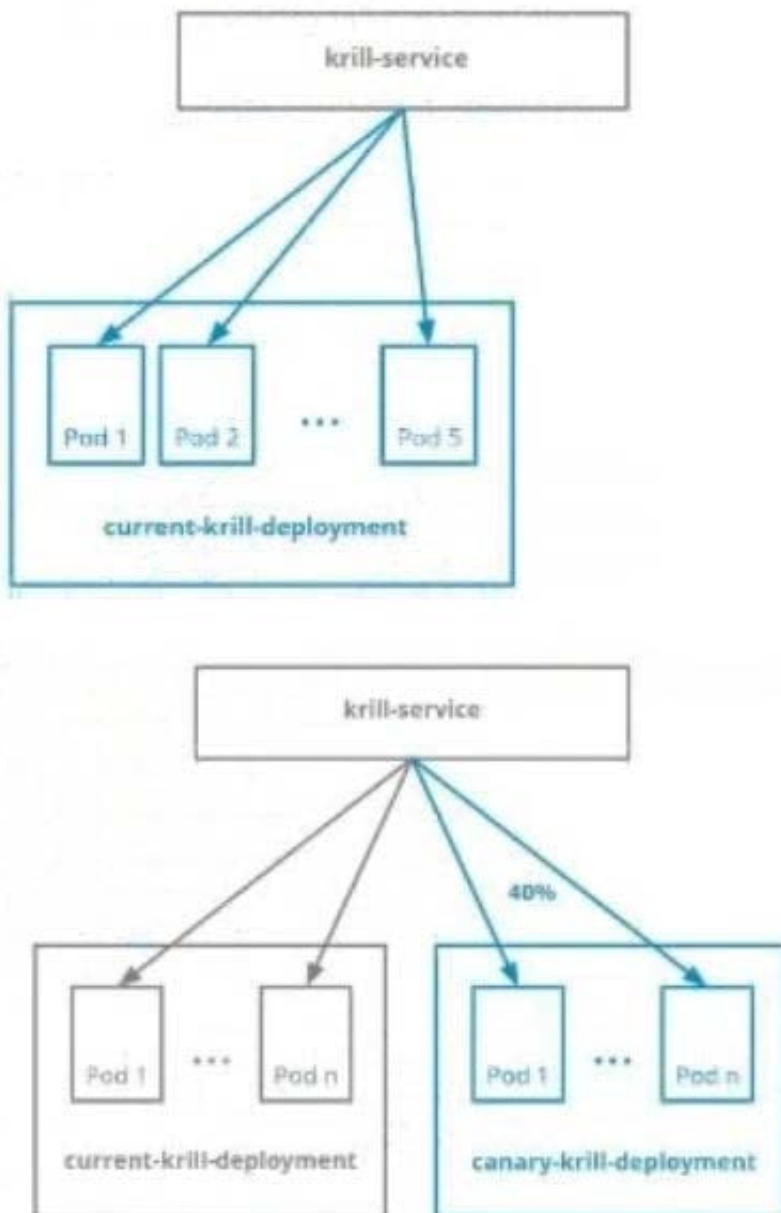
You are asked to prepare a Canary deployment for testing a new application release.

Task:

A Service named krill-Service in the goshark namespace points to 5 pod created by the Deployment named current-krill-deployment



Figure 10-10



The Service is exposed on NodePort 30000. To test its load-balancing, run:

```
[candidate@node-1] $ curl http://k8s-master-0:30000/
```

A. Please check explanations

B. Place Holder

Correct Answer: A



```
candidate@node-1:~/humane-storks$ kubectl scale deploy canary-krill-deployment --replicas 4 -n goshawk
deployment.apps/canary-krill-deployment scaled
candidate@node-1:~/humane-storks$ kubectl get deploy -n goshawk
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
canary-krill-deployment 4/4      4             4           46s
current-krill-deployment 5/5      5             5           7h22m
candidate@node-1:~/humane-storks$ wget https://k8s.io/examples/
```

```
File Edit View Terminal Tabs Help
candidate@node-1:~/humane-storks$ wget https://k8s.io/examples/admin/resource/quota-pod.yaml
--2022-09-24 11:43:51-- https://k8s.io/examples/admin/resource/quota-pod.yaml
Resolving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::
Connecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://kubernetes.io/examples/admin/resource/quota-pod.yaml [following]
--2022-09-24 11:43:52-- https://kubernetes.io/examples/admin/resource/quota-pod.yaml
Resolving kubernetes.io (kubernetes.io)... 147.75.40.148
Connecting to kubernetes.io (kubernetes.io)|147.75.40.148|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90 [application/x-yaml]
Saving to: 'quota-pod.yaml'

quota-pod.yaml      100%[=====>]          90 --.-KB/s   in 0s

2022-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

candidate@node-1:~/humane-storks$ vim quota-pod.yaml
```

```
File Edit View Terminal Tabs Help
2022-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

candidate@node-1:~/humane-storks$ vim quota-pod.yaml
candidate@node-1:~/humane-storks$ kubectl create -f quota-pod.yaml
resourcequota/pod-demo created
candidate@node-1:~/humane-storks$ kubectl get quota -n go
No resources found in go namespace.
candidate@node-1:~/humane-storks$ kubectl get quota -n goshawk
NAME      AGE  REQUEST  LIMIT
pod-demo  19s  pods: 9/10
candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c7995c-kvtjr
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c7995c-4whfm
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-dfk7l
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-z5zrt
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-2774b
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$
```

QUESTION 4

CORRECT TEXT



Set configuration context:



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

Context

Developers occasionally need to submit pods that run periodically.

Task

Follow the steps below to create a pod that will start at a predetermined time and]which runs to completion only once each time it is started:

Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml that runs the following shell command: date in a single busybox container.

The command should run every minute and must complete within 22 seconds or be terminated by Kubernetes. The Cronjob name and container name should both be hello

Create the resource in the above manifest and verify that the job executes successfully at least once

A. Please check explanations

B. Place Holder

Correct Answer: A



```
Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yaml", allowed formats are: go-t
emplate, go-template-file, json, jsonpath, jsonpath-as-json, jsonpath-file, name, template, templatefile
, yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
```

```
Readme Web Terminal THE LINUX FOUNDATION

apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    metadata:
      name: hello
    spec:
      template:
        spec:
          containers:
            - image: busybox
              name: hello
              args: ["/bin/sh", "-c", "date"]
          restartPolicy: Never
  schedule: '*/* * * * *'
  startingDeadlineSeconds: 22
  concurrencyPolicy: Allow
```

```
Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yaml", allowed formats are: go-t
emplate, go-template-file, json, jsonpath, jsonpath-as-json, jsonpath-file, name, template, templatefile
, yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
student@node-1:~$ kubectl create -f /opt/KDPD00301/periodic.yaml
cronjob.batch/hello created
student@node-1:~$ kubectl get cronjob
NAME      SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
hello     */1 * * * *   False    0        <none>           6s
student@node-1:~$
```

**QUESTION 5****CORRECT TEXT****Context**

A user has reported an application is unreachable due to a failing livenessProbe .

Task

Perform the following tasks:

Find the broken pod and store its name and namespace to /opt/KDOB00401/broken.txt in the format:



The output file has already been created

1.

Store the associated error events to a file /opt/KDOB00401/error.txt, The output file has already been created. You will need to use the -o wide output specifier with your command

2.

Fix the issue.



The associated deployment could be running in any of the following namespaces:

- qa
- test
- production
- alan

A. Please check explanations

B. Place Holder

Correct Answer: A

Create the Pod: `kubectl create -f http://k8s.io/docs/tasks/configure-pod-container/exec-liveness.yaml` Within 30 seconds, view the Pod events: `kubectl describe pod liveness-exec` The output indicates that no liveness probes have failed yet:

```
FirstSeen LastSeen Count From SubobjectPath Type Reason Message
-----
----- 24s 24s 1 {default-scheduler} Normal Scheduled Successfully assigned liveness-exec to worker0
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/google_containers/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google_containers/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e
After 35 seconds, view the Pod events again: kubectl describe pod liveness-exec At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated.
FirstSeen LastSeen Count From SubobjectPath Type Reason Message
-----
----- 37s 37s 1 {default-scheduler} Normal Scheduled Successfully assigned liveness-exec to worker0
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/google_containers/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google_containers/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e
2s 2s 1 {kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can't open \'/tmp/healthy\': No such file or directory
Wait another 30 seconds, and verify that the Container has been restarted: kubectl get pod liveness-exec The output shows that RESTARTS has been incremented: NAME READY STATUS RESTARTS AGE liveness-exec 1/1 Running 1 m
```

[Latest CKAD Dumps](#)

[CKAD Practice Test](#)

[CKAD Exam Questions](#)