



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

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.passapply.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 pod is running on the cluster but it is not responding.

#### Task

The desired behavior is to have Kubernetes restart the pod when an endpoint returns an HTTP 500 on the /healthz endpoint. The service, probe-pod, should never send traffic to the pod while it is failing. Please complete the following:

1.

The application has an endpoint, /started, that will indicate if it can accept traffic by returning an HTTP 200. If the endpoint returns an HTTP 500, the application has not yet finished initialization.

2.

The application has another endpoint /healthz that will indicate if the application is still working as expected by returning an HTTP 200. If the endpoint returns an HTTP 500 the application is no longer responsive.

3.

Configure the probe-pod pod provided to use these endpoints

4.

The probes should use port 8080

A. Please check explanations

B. Place Holder

Correct Answer: A

apiVersion: v1



kind: Pod

metadata:

labels:

test: liveness

name: liveness-exec

spec:

containers:

-name: liveness

image: k8s.gcr.io/busybox

args:

-/bin/sh

- -c

-touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600 livenessProbe:

exec:

command:

-cat

-/tmp/healthy

initialDelaySeconds: 5

periodSeconds: 5

In the configuration file, you can see that the Pod has a single Container. The periodSeconds field specifies that the kubelet should perform a liveness probe every 5 seconds. The initialDelaySeconds field tells the kubelet that it should wait 5

seconds before performing the first probe. To perform a probe, the kubelet executes the command `cat /tmp/healthy` in the target container. If the command succeeds, it returns 0, and the kubelet considers the container to be alive and healthy.

If the command returns a non-zero value, the kubelet kills the container and restarts it.

When the container starts, it executes this command:

```
/bin/sh -c "touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600" For the first 30 seconds of the container's life, there is a /tmp/healthy file. So during the first 30 seconds, the command cat /tmp/healthy returns a success code. After 30
```

seconds, `cat /tmp/healthy` returns a failure code.

Create the Pod:



kubectl apply -f <https://k8s.io/examples/pods/probe/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 "k8s.gcr.io/busybox"

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "k8s.gcr.io/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 "k8s.gcr.io/busybox"

36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "k8s.gcr.io/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 1m

```

## QUESTION 2

CORRECT TEXT



### Context

You have been tasked with scaling an existing deployment for availability, and creating a service to expose the deployment within your infrastructure.

### Task

Start with the deployment named kdsn00101-deployment which has already been deployed to the namespace kdsn00101. Edit it to:

1.  
Add the func=webFrontEnd key/value label to the pod template metadata to identify the pod for the service definition

2.

Have 4 replicas

Next, create and deploy in namespace kdsn00101 a service that accomplishes the following:

1.

Exposes the service on TCP port 8080

2.

is mapped to the pods defined by the specification of kdsn00101-deployment

3.

Is of type NodePort

4.

Has a name of cherry

A. Please check explanations

B. Place Holder



Correct Answer: A

```
student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
```

Readme Web Terminal THE LINUX FOUNDATION

```
ⓘ Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2020-10-09T08:50:39Z"
  generation: 1
  labels:
    app: nginx
  name: kdsn00101-deployment
  namespace: kdsn00101
  resourceVersion: "4786"
  selfLink: /apis/apps/v1/namespaces/kdsn00101/deployments/kdsn00101-deployment
  uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
"/tmp/kubectl-edit-d4y5r.yaml" 70L, 1957C 1,1 Top
```



```
Readme Web Terminal THE LINUX FOUNDATION

uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 4
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
        func: webFrontEnd
    spec:
      containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
        - containerPort: 80
```

```
student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
deployment.apps/kdsn00101-deployment edited
student@node-1:~$ kubectl get deployment kdsn00101-deployment -n kdsn00101
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
kdsn00101-deployment 4/4     4            4           7h17m
student@node-1:~$ kubectl expose deployment kdsn00101-deployment -n kdsn00101 --type NodePort --
port 8080 --name cherry
service/cherry exposed
```

### QUESTION 3

CORRECT TEXT

You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

Task:

The pod for the Deployment named nosql in the crayfish namespace fails to start because its container runs out of



resources.

Update the nosql Deployment so that the Pod:

```
➤ The nosql Deployment's manifest file can be found at  
~/chief-cardinal/nosql.yaml.
```

A. Please check explanations

B. Place Holder

Correct Answer: A





```
candidate@node-1:~$ kubectl config use-context k8s  
Switched to context "k8s".  
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
```

```
File Edit View Terminal Tabs Help  
---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: nosql  
  namespace: crayfish  
  labels:  
    app.kubernetes.io/name: nosql  
    app.kubernetes.io/component: backend  
spec:  
  selector:  
    matchLabels:  
      app.kubernetes.io/name: nosql  
      app.kubernetes.io/component: backend  
  replicas: 1  
  template:  
    metadata:  
      labels:  
        app.kubernetes.io/name: nosql  
        app.kubernetes.io/component: backend  
    spec:  
      containers:  
        - name: mongo  
          image: mongo:4.2  
          args:  
            - --bind_ip  
            - 0.0.0.0  
          ports:  
            - containerPort: 27017  
-- INSERT --
```

```
File Edit View Terminal Tabs Help  
- name: mongo  
  image: mongo:4.2  
  args:  
    - --bind_ip  
    - 0.0.0.0  
  ports:  
    - containerPort: 27017  
  resources:  
    requests:  
      memory: "168Mi"  
    limits:  
      memory: "328Mi"  
:  
:wq
```



```
File Edit View Terminal Tabs Help
To: <any> (traffic not restricted by destination)
Policy Types: Ingress, Egress

Name: default-deny
Namespace: ckad00018
Created on: 2022-09-24 04:27:37 +0000 UTC
Labels: <none>
Annotations: <none>
Spec:
  PodSelector: <none> (Allowing the specific traffic to all pods in this namespace)
  Allowing ingress traffic:
    <none> (Selected pods are isolated for ingress connectivity)
  Not affecting egress traffic
  Policy Types: Ingress
candidate@node-1:~$ kubectl label pod ckad00018-newpod -n ckad00018 web-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl label pod ckad00018-newpod -n ckad00018 db-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ kubectl apply -f ~/chief-cardinal/nosql.yaml
deployment.apps/nosql configured
candidate@node-1:~$ kubectl get pods -n crayfish
NAME                                READY   STATUS    RESTARTS   AGE
nosql-74cccf7d64-lkqlg             1/1     Running   0           3m2s
candidate@node-1:~$ kubectl get deploy -n crayfish
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nosql   1/1     1             1           7h16m
candidate@node-1:~$
```

#### QUESTION 4

CORRECT TEXT



Context

You sometimes need to observe a pod's logs, and write those logs to a file for further analysis.

Task

Please complete the following;

1.



Deploy the counter pod to the cluster using the provided YAMLSpec file at /opt/KDOB00201/counter.yaml

2.

Retrieve all currently available application logs from the running pod and store them in the file /opt/KDOB00201/log\_Output.txt, which has already been created

A. Please check explanations

B. Place Holder

Correct Answer: A

```
student@node-1:~$ kubectl create -f /opt/KDOB00201/counter.yaml
pod/counter created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
counter       1/1     Running   0           10s
liveness-http 1/1     Running   0           6h45m
nginx-101     1/1     Running   0           6h46m
nginx-configmap 1/1     Running   0           107s
nginx-secret  1/1     Running   0           7m21s
poller        1/1     Running   0           6h46m
student@node-1:~$ kubectl logs counter
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828fbe5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$
```

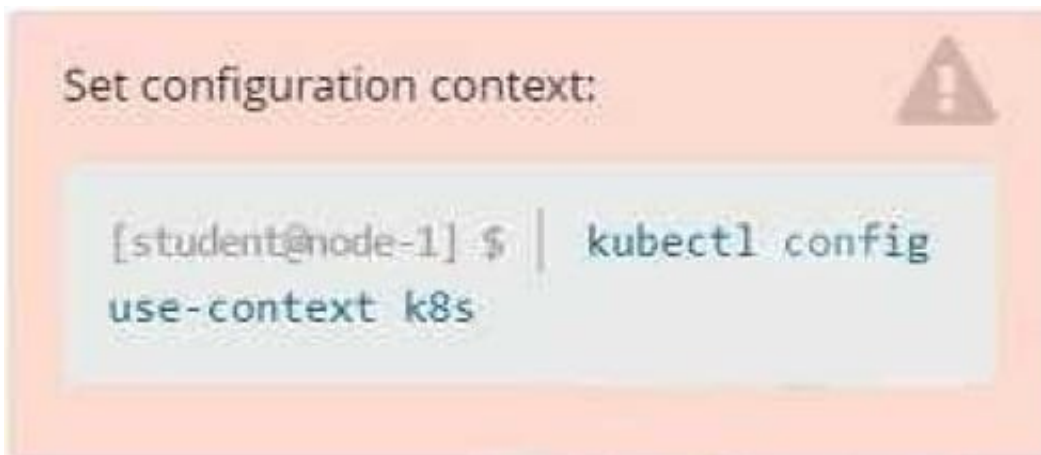
```
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$ cd /opt/KDOB00201/log_output.txt
```



```
Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$ cat /opt/KDOB00201/log_output.txt
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828f8e5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
11: 5493cd16a1790a5fb9512b0c9d4c5dd1
12: 03f169e93e6143438e6dfe4ecb3cc9ed
13: 764b37fe611373c42d0b47154041f6eb
14: 1a56fbe1896b0ee6394136166281839e
15: ecc492eb17715de090c47345a98d98d3
16: 7974a6bec0fb44b6b8bbfc71aa3fbe74
17: 9ae01bef01748b12cc9f97a5f9f72cd6
18: 23fb22ee34d4272e4c9e005f1774515f
19: ec7e1a5d314da9a0ad45d53be5a7acae
20: 0bccdd8ee02cd42029e8162cd1c1197c
21: d6851ea43546216b95bcb81ced997102
22: 7ed9a38ea8bf0d86206569481442af44
23: 29b8416ddc63dbfcb987ab3c8198e9fe
24: 1f2062001df51a108ab25010f506716f
student@node-1:~$
```

### QUESTION 5

CORRECT TEXT



Task

Create a new deployment for running nginx with the following parameters:

1.

Run the deployment in the kdpd00201 namespace. The namespace has already been created

2.



Name the deployment frontend and configure with 4 replicas

3.

Configure the pod with a container image of lfcncf/nginx:1.13.7

4.

Set an environment variable of NGINX\_\_PORT=8080 and also expose that port for the container above

A. Please check explanations

B. Place Holder

Correct Answer: A



```
student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4  
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml  
student@node-1:~$ vim nginx_deployment.yml
```

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  creationTimestamp: null  
  labels:  
    app: api  
  name: api  
  namespace: kdpd00201  
spec:  
  replicas: 4  
  selector:  
    matchLabels:  
      app: api  
  strategy: {}  
  template:  
    metadata:  
      creationTimestamp: null  
      labels:  
        app: api  
    spec:  
      containers:  
      - image: lfcncf/nginx:1.13.7-alpine  
        name: nginx  
        resources: {}  
status: {}  
~  
"nginx_deployment.yml" 25L, 421C 4,1 All
```

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  labels:  
    app: api  
  name: api  
  namespace: kdpd00201  
spec:  
  replicas: 4  
  selector:  
    matchLabels:  
      app: api  
  template:  
    metadata:  
      labels:  
        app: api  
    spec:  
      containers:  
      - image: lfcncf/nginx:1.13.7-alpine  
        name: nginx  
        ports:  
        - containerPort: 8080  
      env:  
      - name: NGINX_PORT  
        value: "8080"  
~  
23,8 All
```

```
student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4  
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml  
student@node-1:~$ vim nginx_deployment.yml  
student@node-1:~$ kubectl create nginx_deployment.yml  
Error: must specify one of -f and -k  
  
error: unknown command "nginx_deployment.yml"  
See 'kubectl create -h' for help and examples  
student@node-1:~$ kubectl create -f nginx_deployment.yml  
error: error validating "nginx_deployment.yml": error validating data: ValidationError(Deployment.spec.template.spec): unknown field "env" in io.k8s.api.core.v1.PodSpec; if you choose to ignore these errors, turn validation off with --validate=false  
student@node-1:~$ vim nginx_deployment.yml  
student@node-1:~$ kubectl create -f nginx_deployment.yml  
deployment.apps/api created  
student@node-1:~$ kubectl get pods -n kdpd00201  
NAME READY STATUS RESTARTS AGE  
api-745677f7dc-7hnmv 1/1 Running 0 13s  
api-745677f7dc-9q5vp 1/1 Running 0 13s  
api-745677f7dc-fd4gk 1/1 Running 0 13s  
api-745677f7dc-mbnpk 1/1 Running 0 13s  
student@node-1:~$
```



VCE & PDF

PassApply.com

<https://www.passapply.com/ckad.html>

2024 Latest passapply CKAD PDF and VCE dumps Download

---

[CKAD VCE Dumps](#)

[CKAD Study Guide](#)

[CKAD Braindumps](#)