



# 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



Task

A deployment is falling on the cluster due to an incorrect image being specified. Locate the deployment, and fix the problem.

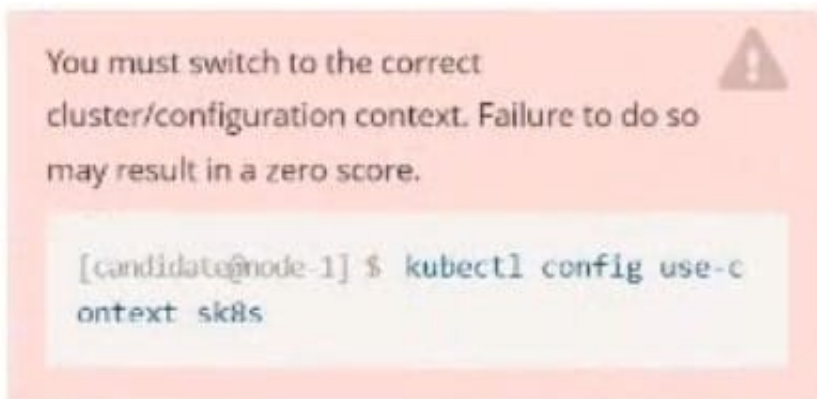
- A. Please check explanations
- B. Place Holder

Correct Answer: A

```
create deploy hello-deploy --image=nginx --dry-run=client -o yaml > hello-deploy.yaml Update deployment image to  
nginx:1.17.4: kubectl set image deploy/hello-deploy nginx=nginx:1.17.4
```

## QUESTION 2

CORRECT TEXT





Task:

Create a Pod named nginx resources in the existing pod resources namespace.

Specify a single container using nginx:stable image.

Specify a resource request of 300m cpus and 1Gi of memory for the Pod's container.

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:~$ kubectl run nginx-resources -n pod-resources --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
```

```
File Edit View Terminal Tabs Help
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx:stable
    name: nginx-resources
    resources:
      requests:
        cpu: 300m
        memory: "1Gi"
```



```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl run nginx-resources -n pod-resources --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
candidate@node-1:~$ kubectl create -f hw.yaml
pod/nginx-resources created
candidate@node-1:~$ kubectl get pods -n pod-resources
NAME          READY   STATUS    RESTARTS   AGE
nginx-resources 1/1     Running   0           13s
candidate@node-1:~$ kubectl describe pods -n pod-resources
```

```
File Edit View Terminal Tabs Help
memory: 1Gi
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dmx9j (ro)
Conditions:
  Type           Status
  Initialized     True
  Ready          True
  ContainersReady True
  PodScheduled   True
Volumes:
  kube-api-access-dmx9j:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
QoS Class:           Burstable
Node-Selectors:      <none>
Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                     node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason      Age   From          Message
  ----    -
  Normal  Scheduled  20s   default-scheduler  Successfully assigned pod-resources/nginx-resources to k8s-node-0
  Normal  Pulling    19s   kubelet        Pulling image "nginx:stable"
  Normal  Pulled     13s   kubelet        Successfully pulled image "nginx:stable" in 6.55664052s
  Normal  Created    13s   kubelet        Created container nginx-resources
  Normal  Started    12s   kubelet        Started container nginx-resources
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl create deploy expose -n ckad00014 --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml >
```

**QUESTION 3**

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 4

CORRECT TEXT



Context

Your application's namespace requires a specific service account to be used.

Task

Update the app-a deployment in the production namespace to run as the restrictedservice service account.

The service account has already been created.





A. Please check explanations

B. Place Holder

Correct Answer: A

```
student@node-1:~$ kubectl get serviceaccount -n production
NAME          SECRETS  AGE
default       1        6h46m
restrictedservice 1        6h46m
student@node-1:~$ kubectl get deployment -n production
NAME    READY  UP-TO-DATE  AVAILABLE  AGE
app-a   3/3    3           3          6h46m
student@node-1:~$ kubectl set serviceaccount deployment app-a restrictedservice -n production
deployment.apps/app-a serviceaccount updated
student@node-1:~$
```

### QUESTION 5

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: "166Mi"  
  limits:  
    memory: "320Mi"  
:  
: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:~$
```

[CKAD PDF Dumps](#)

[CKAD Practice Test](#)

[CKAD Exam Questions](#)