



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

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



[Readme](#) [Web Terminal](#)

THE **LINUX** FOUNDATION

```
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 2

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:

```
<namespace>/<pod>
```

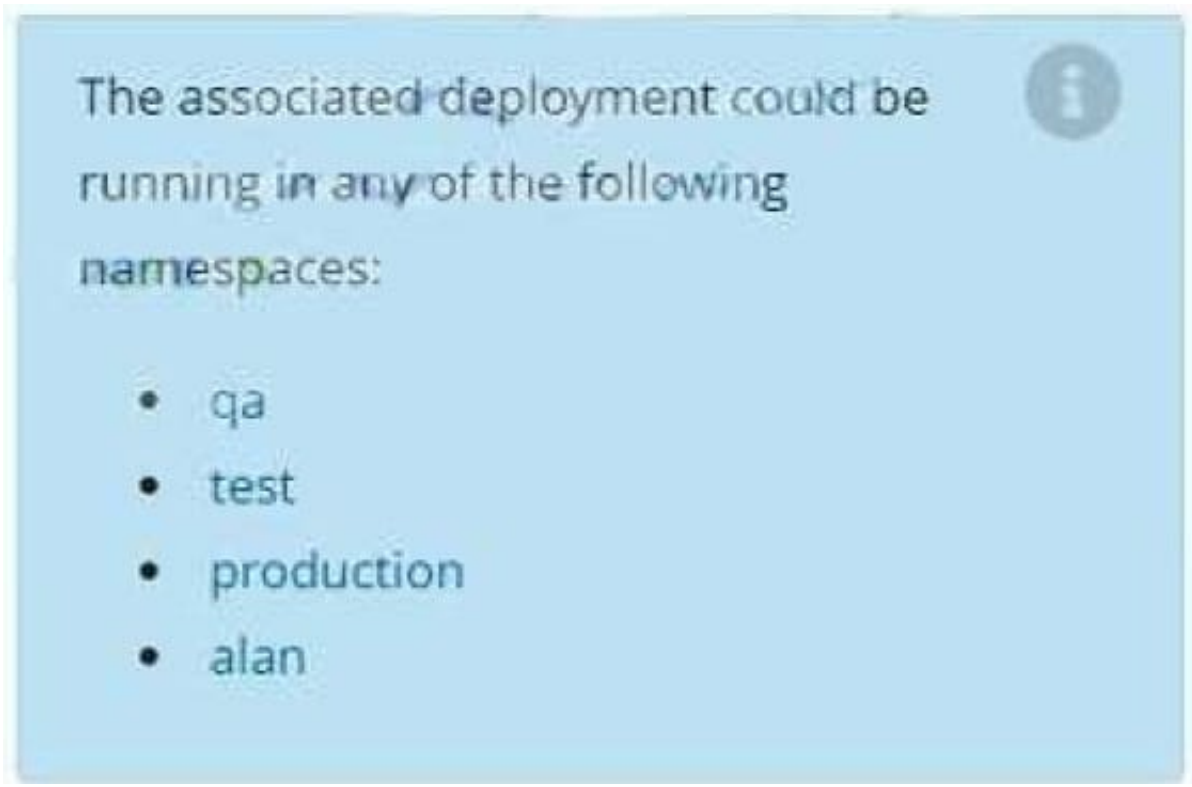
The output file has already been created

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 -----



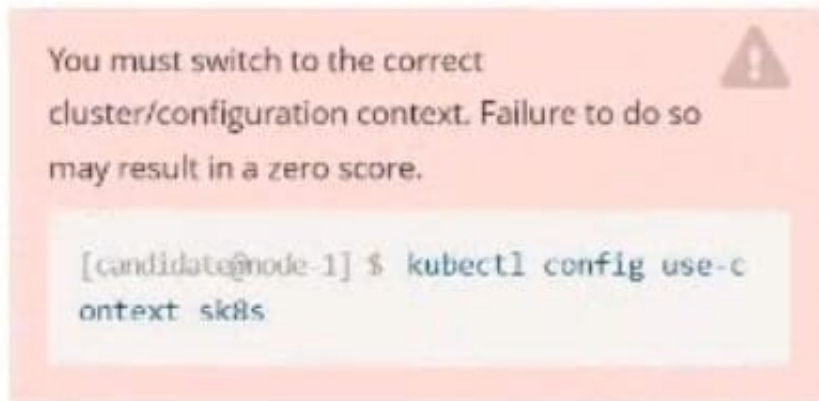
```

----- 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

```

QUESTION 3

CORRECT TEXT



Task:

1.

Update the Propertunel scaling configuration of the Deployment web1 in the ckad00015 namespace setting maxSurge to 2 and maxUnavailable to 59

2.

Update the web1 Deployment to use version tag 1.13.7 for the lflconf/nginx container image.

3.

Perform a rollback of the web1 Deployment to its previous version

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 edit deploy web1 -n ckad00015
```

```
File Edit View Terminal Tabs Help
  app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 2%
      maxUnavailable: 5%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: lfccncf/nginx:1.13.7
        imagePullPolicy: IfNotPresent
        name: nginx
        ports:
        - containerPort: 80
          protocol: TCP
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
  status:
    availableReplicas: 2
    conditions:
      - lastTransitionTime: "2022-09-24T04:26:41Z"
```

```
File Edit View Terminal Tabs Help
Switched to context "k8s".
candidate@node-1:~$ kubectl create secret generic app-secret -n default --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME          TYPE          DATA   AGE
app-secret    Opaque        1       4s
candidate@node-1:~$ kubectl run nginx-secret -n default --image=nginx:stable --dry-run=client -o yaml > sec.yaml
candidate@node-1:~$ vim sec.yaml
candidate@node-1:~$ kubectl create -f sec.yaml
pod/nginx-secret created
candidate@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-secret  1/1     Running   0           7s
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy web1 -n ckad00015
deployment.apps/web1 edited
candidate@node-1:~$ kubectl rollout status deploy web1 -n ckad00015
deployment "web1" successfully rolled out
candidate@node-1:~$ kubectl rollout undo deploy web1 -n ckad00015
deployment.apps/web1 rolled back
candidate@node-1:~$ kubectl rollout history deploy web1 -n ckad00015
deployment.apps/web1
REVISION   CHANGE-CAUSE
2          <none>
3          <none>
candidate@node-1:~$ kubectl get rs -n ckad00015
NAME          DESIRED   CURRENT   READY   AGE
web1-56f98bcb79  0         0         0       63s
web1-85775b6b79  2         2         2       6h53m
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: 40d3a9c8e46f5533bb4828fbc5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
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:~$ cat /opt/KDOB00201/log_output.txt
```

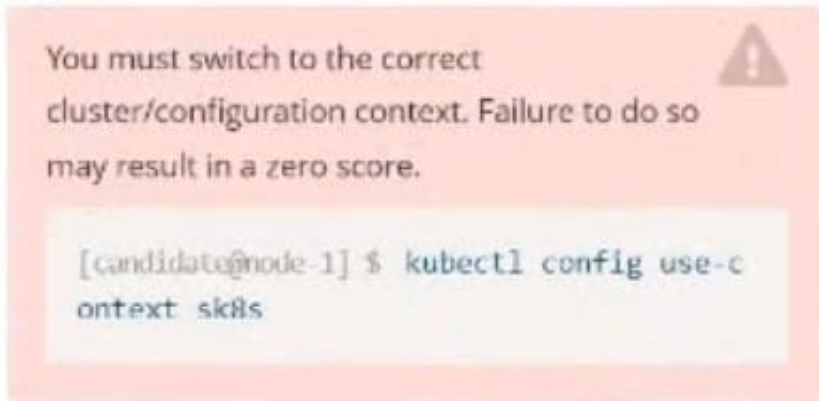
Readme > Web Terminal

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: 40d3a9c8e46f5533bb4828fbc5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
11: 5493cd16a1790a5fb9512b0c9d4c5dd1
12: 03f169e93e6143438e6dfe4ecb3cc9ed
13: 764b37fe611373c42d0b47154041f6eb
14: 1a56fbc1896b0ee6394136166281839e
15: ecc492eb17715de090c47345a98d98d3
16: 7974a6bec0fb44b6b8bbfc71aa3fbc74
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 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"
```

:wq



```
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 lfccnrf/nginx:1.13.7 --dry-run=client -o yaml>
```

[CKAD VCE Dumps](#)[CKAD Study Guide](#)[CKAD Exam Questions](#)