



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.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 Anytime a team needs to run a container on Kubernetes they will need to define a pod within which to run the container. Task Please complete the following:

1.

Create a YAML formatted pod manifest

2.

/opt/KDPD00101/pod1.yml to create a pod named app1 that runs a container named app1cont using image lfcncf/arg-output with these command line arguments: -lines 56 -F

3.


Create the pod with the kubectl command using the YAML file created in the previous step

4.

When the pod is running display summary data about the pod in JSON format using the kubectl command and redirect the output to a file named /opt/KDPD00101/out1.json

5.

All of the files you need to work with have been created, empty, for your convenience



When creating your pod, you do not need to specify a container command, only args.

A. Please check explanations

B. Place Holder

Correct Answer: A

```
student@node-1:~$ kubectl run appl --image=lfcncf/arg-output --dry-run=client -o yaml > /opt/KDPD00101/pod1.yml
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```



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

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

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: app1
  name: app1
spec:
  containers:
  - image: lfccncf/arg-output
    name: app1
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

```
"/opt/KDPD00101/pod1.yml" 15L, 242C
```

3,1

All



```
Readme Web Terminal THE LINUX FOUNDATION

apiVersion: v1
kind: Pod
metadata:
  labels:
    run: appl1
  name: appl1
spec:
  containers:
  - image: lfccncf/arg-output
    name: appl1
    args: ["--lines", "50", "--s
```

```
pod/appl1 created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS             RESTARTS   AGE
appl1         0/1     ContainerCreating   0           5s
counter       1/1     Running             0           4m44s
liveness-http 1/1     Running             0           6h50m
nginx-101     1/1     Running             0           6h51m
nginx-configmap 1/1     Running             0           6m21s
nginx-secret  1/1     Running             0           11m
poller        1/1     Running             0           6h51m
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
appl1         1/1     Running   0           26s
counter       1/1     Running   0           5m5s
liveness-http 1/1     Running   0           6h50m
nginx-101     1/1     Running   0           6h51m
nginx-configmap 1/1     Running   0           6m42s
nginx-secret  1/1     Running   0           12m
poller        1/1     Running   0           6h51m
student@node-1:~$ kubectl delete pod appl1
pod "appl1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```

```
Readme Web Terminal

nginx-configmap 1/1     Running   0           6m2
nginx-secret    1/1     Running   0           11m
poller          1/1     Running   0           6h5
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
appl1         1/1     Running   0           26s
counter       1/1     Running   0           5m5s
liveness-http 1/1     Running   0           6h50m
nginx-101     1/1     Running   0           6h51m
nginx-configmap 1/1     Running   0           6m42s
nginx-secret  1/1     Running   0           12m
poller        1/1     Running   0           6h51m
student@node-1:~$ kubectl delete pod appl1
pod "appl1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/appl1 created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
appl1         1/1     Running   0           20s
counter       1/1     Running   0           6m57s
liveness-http 1/1     Running   0           6h52m
nginx-101     1/1     Running   0           6h53m
nginx-configmap 1/1     Running   0           8m34s
nginx-secret  1/1     Running   0           14m
poller        1/1     Running   0           6h53m
student@node-1:~$ kubectl get pod appl1 -o json >
```



Readme

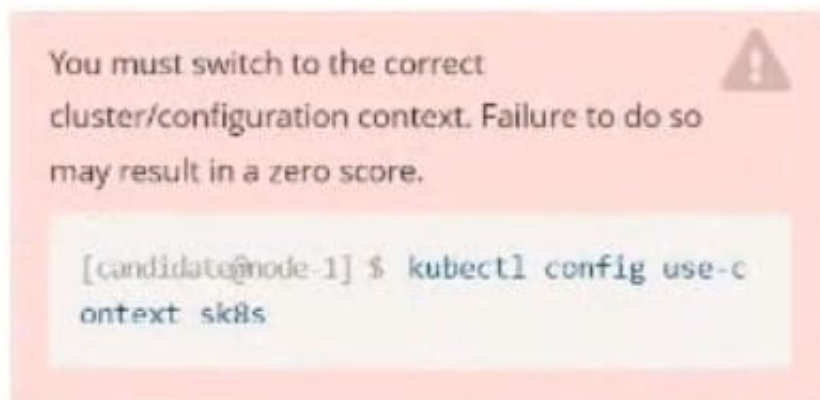
Web Terminal

THE LINUX FOUNDATION

```
poller          1/1      Running          0          6h51m
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
app1          1/1     Running    0          26s
counter       1/1     Running    0          5m5s
liveness-http 1/1     Running    0          6h50m
nginx-101     1/1     Running    0          6h51m
nginx-configmap 1/1     Running    0          6m42s
nginx-secret   1/1     Running    0          12m
poller        1/1     Running    0          6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
app1          1/1     Running    0          20s
counter       1/1     Running    0          6m57s
liveness-http 1/1     Running    0          6h52m
nginx-101     1/1     Running    0          6h53m
nginx-configmap 1/1     Running    0          8m34s
nginx-secret   1/1     Running    0          14m
poller        1/1     Running    0          6h53m
student@node-1:~$ kubectl get pod app1 -o json > /opt/KDPD00101/out1.json
student@node-1:~$
student@node-1:~$
```

QUESTION 2

CORRECT TEXT



Task:

A pod within the Deployment named buffalo-deployment and in namespace gorilla is logging errors.

Look at the logs identify errors messages.

Find errors, including User "system:serviceaccount:gorilla:default" cannot list resource "deployment" [...] in the namespace "gorilla"

The buffalo-deployment `S manifest can be found at -/prompt/escargot/buffalo- deployment.yaml



A. Please check explanations

B. Place Holder

Correct Answer: A

```
File Edit View Terminal Tabs Help
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME                                READY   STATUS    RESTARTS   AGE
backend-deployment-59d449b99d-cxct6 1/1     Running   0           20s
backend-deployment-59d449b99d-h2zjq 0/1     Running   0           9s
backend-deployment-78976f74f5-b8c85 1/1     Running   0           6h40m
backend-deployment-78976f74f5-flfsj 1/1     Running   0           6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3     3             3           6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3     3             3           6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME                                READY   STATUS              RESTARTS   AGE
buffalo-deployment-776844df7f-r5fsb 1/1     Running             0           5h38m
buffalo-deployment-859898c6f5-zx5gj 0/1     ContainerCreating   0           8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
buffalo-deployment  1/1     1             1           6h38m
candidate@node-1:~$
```

QUESTION 3


CORRECT TEXT






Task

You have rolled out a new pod to your infrastructure and now you need to allow it to communicate with the web and storage pods but nothing else. Given the running pod `kdsn00201` -newpod edit it to use a network policy that will allow it to send and receive traffic only to and from the web and storage pods.



All work on this item should be conducted in the `kdsn00201` namespace.



All required NetworkPolicy resources are already created and ready for use as appropriate. You should not create, modify or delete any network policies whilst completing this item.

A. Please check explanations

B. Place Holder

Correct Answer: A

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: internal-policy

namespace: default

spec:

podSelector:

matchLabels:



name: internal

policyTypes:

-Egress

-Ingress ingress:

-{} egress:

-to:

-podSelector: matchLabels:

name: mysql ports:

-protocol: TCP port: 3306

-to:

-podSelector: matchLabels: name: payroll ports:

-protocol: TCP port: 8080

-ports:

-

port: 53 protocol: UDP

-

port: 53 protocol: TCP

QUESTION 4

CORRECT TEXT

Set configuration context:



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




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/kubectrl-edit-d4y5r.yaml" 70L, 1957C 1,1 Top
```



```
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 5

CORRECT TEXT





Context

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

Task

The desired behavior is to have Kubemetes 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

[CKAD VCE Dumps](#)

[CKAD Practice Test](#)

[CKAD Braindumps](#)