



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.geekcert.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

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
```



```
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 2

CORRECT TEXT

```
Set configuration context: 

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

Context

A web application requires a specific version of redis to be used as a cache.

Task



Create a pod with the following characteristics, and leave it running when complete:

1.

The pod must run in the web namespace.

2.

The namespace has already been created

3.

The name of the pod should be cache

4.

Use the lfccncf/redis image with the 3.2 tag

5.

Expose port 6379

A. Please check explanations

B. Place Holder

Correct Answer: A

```
student@node-1:~$ kubectl run cache --image=lfccncf/redis:3.2 --port=6379 -n web
pod/cache created
student@node-1:~$ kubectl get pods -n web
NAME     READY   STATUS             RESTARTS   AGE
cache    0/1     ContainerCreating   0           6s
student@node-1:~$ kubectl get pods -n web
NAME     READY   STATUS    RESTARTS   AGE
cache    1/1     Running   0           9s
student@node-1:~$
```

QUESTION 3

CORRECT TEXT



Given a container that writes a log file in format A and a container that converts log files from format A to format B, create a deployment that runs both containers such that the log files from the first container are converted by the second container, emitting logs in format

Task:

1.

Create a deployment named deployment-xyz in the default namespace, that:

2.

Includes a primary

3.

Ifccncf/busybox:1 container, named logger-dev

4.

Includes a sidecar Ifccncf/fluentd:v0.12 container, named adapter-zen Mounts a shared volume /tmp/log on both containers, which does not persist when the pod is deleted

5.

Instructs the logger-dev

container to run the command

```
while true; do
  echo "i luv cncf" >> /
  tmp/log/input.log;
  sleep 10;
done
```

which should output logs to /tmp/log/input.log in plain text format, with example values:



```
i luv cncf  
i luv cncf  
i luv cncf
```

The adapter-zen sidecar container should read /tmp/log/input.log and output the data to /tmp/log/output.* in Fluentd JSON format. Note that no knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at /opt/KDMC00102/fluentd-configmap.yaml , and mount that ConfigMap to /fluentd/etc in the adapter-zen sidecar container

A. Please check explanations

B. Place Holder

Correct Answer: A

```
student@node-1:~$ kubectl create deployment deployment-xyz --image=lfccncf/busybox:1 --dry-run=c  
lient -o yaml > deployment_xyz.yml  
student@node-1:~$ vim deployment_xyz.yml
```

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  creationTimestamp: null  
  labels:  
    app: deployment-xyz  
  name: deployment-xyz  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: deployment-xyz  
  strategy: {}  
  template:  
    metadata:  
      creationTimestamp: null  
      labels:  
        app: deployment-xyz  
    spec:  
      containers:  
      - image: lfccncf/busybox:1  
        name: busybox  
        resources: {}  
status: {}  
~  
~  
"deployment_xyz.yml" 24L, 434C 3,1 All
```




```
Readme Web Terminal THE LINUX FOUNDATION
Kind: Deployment
metadata:
  labels:
    app: deployment-xyz
  name: deployment-xyz
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deployment-xyz
  template:
    metadata:
      labels:
        app: deployment-xyz
    spec:
      volumes:
      - name: myvoll
        emptyDir: {}
      containers:
      - image: lfccncf/busybox:1
        name: logger-dev
        volumeMounts:
        - name: myvoll
          mountPath: /tmp/log
      - image: lfccncf/Fluentd:v0.12
        name: adapter-zen
3 lines yanked 27,22 Bot
```

```
Readme Web Terminal THE LINUX FOUNDATION
replicas: 1
selector:
  matchLabels:
    app: deployment-xyz
template:
  metadata:
    labels:
      app: deployment-xyz
  spec:
    volumes:
    - name: myvoll
      emptyDir: {}
    containers:
    - image: lfccncf/busybox:1
      name: logger-dev
      command: ["/bin/sh", "-c", "while [ true ]; do echo 'i lov cncf' >> /tmp/log/input.log; sl
esp 10; done"]
      volumeMounts:
      - name: myvoll
        mountPath: /tmp/log
    - image: lfccncf/Fluentd:v0.12
      name: adapter-zen
      command: ["/bin/sh", "-c", "tail -f /tmp/log/input.log >> /tmp/log/output.log"]
      volumeMounts:
      - name: myvoll
        mountPath: /tmp/log
29,83 Bot
```

```
Readme Web Terminal THE LINUX FOUNDATION
metadata:
  labels:
    app: deployment-xyz
spec:
  volumes:
  - name: myvoll
    emptyDir: {}
  - name: myvoll2
    configMap:
      name: logconf
  containers:
  - image: lfccncf/busybox:1
    name: logger-dev
    command: ["/bin/sh", "-c", "while [ true ]; do echo 'i lov cncf' >> /tmp/log/input.log; sl
esp 10; done"]
    volumeMounts:
    - name: myvoll
      mountPath: /tmp/log
  - image: lfccncf/Fluentd:v0.12
    name: adapter-zen
    command: ["/bin/sh", "-c", "tail -f /tmp/log/input.log >> /tmp/log/output.log"]
    volumeMounts:
    - name: myvoll
      mountPath: /tmp/log
    - name: myvoll2
      mountPath: /fluentd/et
37,33 Bot
```

```
student@node-1:~$ kubectl create -f deployment_xyz.yml
deployment.apps/deployment-xyz created
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 0/1     1             0           5s
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 0/1     1             0           9s
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 1/1     1             1          12s
student@node-1:~$
```




QUESTION 4

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

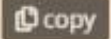
2.

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 sk8s-node-0
```

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

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
Web Terminal
-- INSERT --
0,1 All

```

```

THE LINUX FOUNDATION
Web Terminal
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
Web Terminal
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      9s
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
Web Terminal
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  restartPolicy: Never
  containers:
    - name: myfrontend
      image: nginx
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: mypod
    - name: mypod
      image: mypod
      volumeMounts:
        - 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

```

```

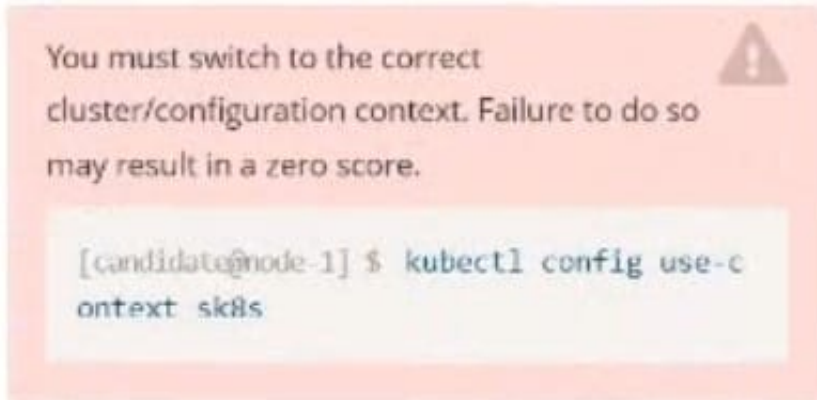
THE LINUX FOUNDATION
Web Terminal
student@sk8s-node-01:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
mypod   0/1     ContainerCreating   0          4s
student@sk8s-node-01:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
mypod   0/1     ContainerCreating   0          8s
student@sk8s-node-01:~$ 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 5

CORRECT TEXT



Task:

To run 2 replicas of the pod

Add the following label on the pod:

Role userUI

A. Please check explanations

B. Place Holder

Correct Answer: A



```
File Edit View Terminal Tabs Help
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2022-09-24T04:27:03Z"
  generation: 1
  labels:
    app: nginx
    name: ckad00017-deployment
    namespace: ckad00017
    resourceVersion: "3349"
    uid: lcd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
    labels:
-- INSERT --
```

33, 14

5%

```
File Edit View Terminal Tabs Help
name: ckad00017-deployment
namespace: ckad00017
resourceVersion: "3349"
uid: lcd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: nginx
      role: userUI
  spec:
    containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
          - containerPort: 80
            protocol: TCP
        resources: {}
-- INSERT --
```

35, 21

33%



```
File Edit View Terminal Tabs Help
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 6h38m
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:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$
```

```
File Edit View Terminal Tabs Help
candidate@node-1:~$ kubectl get pods -n gorilla
NAME READY STATUS RESTARTS AGE
buffalo-deployment-776844df7f-r5fsb 1/1 Running 0 6h38m
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:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose service ckad00017-deployment -n ckad0001
--name=cherry --port=8888 --type=NodePort
service/cherry exposed
candidate@node-1:~$
```

```
candidate@node-1:~$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 77d
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
cherry NodePort 10.100.100.176 <none> 8888:30683/TCP 24s
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017
--name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
cherry NodePort 10.100.100.176 <none> 8888:30683/TCP 45s
candidate@node-1:~$
```



```
File Edit View Terminal Tabs Help
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
cherry    NodePort  10.100.100.176  <none>           8888:30683/TCP  46s
candidate@node-1:~$ history
1  vi ~/spicy-pikachu/backend-deployment.yaml
2  kubectl config use-context sk8s
3  vim .vimrc
4  vim ~/spicy-pikachu/backend-deployment.yaml
5  kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
6  kubectl get pods -n staging
7  kubectl get deploy -n staging
8  vim ~/spicy-pikachu/backend-deployment.yaml
9  kubectl config use-context k8s
10 kubectl set serviceaccount deploy app-1 app -n frontend
11 kubectl config use-context k8s
12 vim ~/prompt-escargot/buffalo-deployment.yaml
13 kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
14 kubectl get pods -n gorilla
15 kubectl get deploy -n gorilla
16 kubectl config use-context k8s
17 kubectl edit deploy ckad00017-deployment -n ckad00017
18 kubectl expose deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
19 kubectl get svc
20 kubectl get svc -n ckad00017
21 kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
22 kubectl get svc -n ckad00017
23 history
candidate@node-1:~$
```

[CKAD VCE Dumps](#)

[CKAD Practice Test](#)

[CKAD Exam Questions](#)