

# Linux Foundation CKA Web-Based Practice Exam - Reliable Online Self-Assessment Test

A screenshot of a web terminal window titled "Readme" and "Web Terminal" with "THE LINUX FOUNDATION" logo in the top right. The terminal displays the following output:

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: kuoc1
  name: kuoc1
spec:
  containers:
  - image: nginx
    name: nginx
  - image: consul
    name: consul
```

BONUS!!! Download part of VCE4Dumps CKA dumps for free: [https://drive.google.com/open?id=15G0DGIUyD\\_b6Gaj\\_i6iK7dxJemndzv](https://drive.google.com/open?id=15G0DGIUyD_b6Gaj_i6iK7dxJemndzv)

Completing the preparation for the Certified Kubernetes Administrator (CKA) Program Exam exam on time is the most important aspect. The other thing is to prepare for the Certified Kubernetes Administrator (CKA) Program Exam exam by evaluating your preparation using authentic exam questions. VCE4Dumps provides the most authentic Certified Kubernetes Administrator (CKA) Program Exam (CKA) Exam Questions compiled according to the rules or patterns supplied by Certified Kubernetes Administrator (CKA) Program Exam (CKA) professionals. We provide you with everything you need to pass the CKA exam, which verifies you as a Linux Foundation certified specialist in the domain of Linux Foundation Data Modeling.

The CKA certification exam is a challenging test that requires candidates to demonstrate their mastery of Kubernetes. CKA exam covers a wide range of topics, including installation, configuration, and management of Kubernetes clusters. It also tests a candidate's ability to work with the Kubernetes API, troubleshoot common issues, and perform advanced tasks such as scaling and rolling updates.

The CKA program is becoming increasingly popular due to the growing demand for Kubernetes administrators in the industry. Organizations are looking for individuals who can manage and administer Kubernetes clusters effectively, ensuring that applications are deployed and maintained efficiently. The CKA program provides individuals with the necessary skills and knowledge to meet these demands.

The CKA Certification Exam is a hands-on test that requires candidates to perform Kubernetes administration tasks using a command line interface. CKA exam evaluates the candidate's ability to deploy and manage Kubernetes clusters, configure networking, and implement security policies. It also tests their knowledge of Kubernetes architecture, troubleshooting, and best practices. CKA exam is challenging and requires extensive preparation and practice to pass. However, passing the CKA certification exam can significantly enhance the career prospects of Kubernetes professionals, as it is a widely recognized credential in the industry.

>> [Pass4sure CKA Study Materials](#) <<

## Pass4sure CKA Study Materials & Leading Provider in Qualification Exams & CKA Accurate Answers

On the basis of the current social background and development prospect, the CKA certifications have gradually become accepted prerequisites to stand out the most in the workplace. But it is not easy for every one to achieve their CKA certification since the CKA Exam is quite difficult and takes time to prepare for it. Our CKA exam materials are pleased to serve you as such an exam tool to win the exam at your first attempt. If you don't believe it, just come and try!

## Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q74-Q79):

### NEW QUESTION # 74

Create a busybox pod and add "sleep 3600" command

#### Answer:

Explanation:

See the solution below.

Explanation

```
kubectl run busybox --image=busybox --restart=Never -- /bin/sh -c
"sleep 3600"
```

### NEW QUESTION # 75

Create a pod with init container which create a file "test.txt" in "workdir" directory. Main container should check a file "test.txt" exists and execute sleep 9999 if the file exists.

- A. // create an initial yaml file with this

```
kubectl run init-cont-pod --image=alpine --restart=Never --dry-run -o
yaml > init-cont-pod.yaml
// edit the yml as below and create it
vim init-cont-pod.yaml
apiVersion: v1
kind: Pod
metadata:
name: init-cont-pod
labels:
app: myapp
spec:
volumes:
- name: test-volume
emptyDir: {}
containers:
- name: main-container
image: busybox:1.28
command: ['sh', '-c', 'if [ -f/workdir/test.txt ]; then sleep
9999; fi']
volumeMounts:
- name: test-volume
mountPath: /workdir
initContainers:
- name: init-myservice
image: busybox:1.28
command: ['sh', '-c', 'mkdir /workdir; echo >
/workdir/test.txt']
volumeMounts:
- name: test-volume
mountPath: /workdir
// Create the pod
kubectl apply -f init-cont-pod.yaml
kubectl get pods
// Check Events by doing
kubectl describe po init-cont-pod
Init Containers:
init-myservice:
Container ID:
docker://ebdbf5fad1c95111d9b0e0e2e743c2e347c81b8d4eb5abcccdfe1dd74524
0d4f
Image: busybox:1.28
Image ID: dockerpullable://busybox@sha256:141c253bc4c3fd0a201d32dc1f493bc3fff003b6df
416dea4f41046e0f37d47
```

```
Port: <none>
Host Port: <none>
Command:
sh
-c
mkdir /workdir; echo > /workdir/test.txt
State: Terminated Reason: Completed
```

- B. // create an initial yaml file with this

```
kubectl run init-cont-pod --image=alpine --restart=Never --dry-run -o
yaml > init-cont-pod.yaml
// edit the yml as below and create it
vim init-cont-pod.yaml
apiVersion: v1
kind: Pod
metadata:
name: init-cont-pod
labels:
app: myapp
spec:
volumes:
- name: test-volume
emptyDir: {}
containers:
- name: main-container
image: busybox:1.28
command: ['sh', '-c', 'if [ -f /workdir/test.txt ]; then sleep
9999; fi']
volumeMounts:
image: busybox:1.28
command: ['sh', '-c', "mkdir /workdir; echo >
/workdir/test.txt"]
volumeMounts:
- name: test-volume
mountPath: /workdir
// Create the pod
kubectl apply -f init-cont-pod.yaml
kubectl get pods
// Check Events by doing
kubectl describe po init-cont-pod
Init Containers:
init-myservice:
Container ID:
docker://ebdbf5fad1c95111d9b0e0e2e743c2e347c81b8d4eb5abcccdfe1dd74524
0d4f
Image: busybox:1.28
Image ID: dockerpullable://busybox@sha256:141c253bc4c3fd0a201d32dc1f493bc3fff003b6df
416dea4f41046e0f37d47
Port: <none>
Host Port: <none>
Command:
sh
-c
mkdir /workdir; echo > /workdir/test.txt
State: Terminated Reason: Completed
```

**Answer: A**

## NEW QUESTION # 76

Create a Job with an image node which prints node version and

verifies there is a pod created for this job

- A. `kubectrl create job nodeversion --image=node -- node -v`  
`kubectrl get job -w`  
`kubectrl get pod`  
YAML File:  
apiVersion: batch/v1  
kind: Job  
metadata:  
labels:  
job-name: nodeversion  
name: nodeversion  
spec:  
completions: 1  
parallelism: 1  
selector:  
matchLabels:  
job-name: nodeversion  
template:  
metadata:  
labels:  
job-name: nodeversion  
spec:  
containers:  
- command:  
- node  
- -v  
image: node  
imagePullPolicy: Always  
name: nodeversion  
restartPolicy: Never
- B. `kubectrl create job nodeversion --image=node -- node -v`  
`kubectrl get job -w`  
`kubectrl get pod`  
YAML File:  
apiVersion: batch/v1  
kind: Job  
metadata:  
labels:  
job-name: nodeversion  
name: nodeversion  
spec:  
completions: 1  
parallelism: 1  
labels:  
job-name: nodeversion  
spec:  
containers:  
- command:  
- node  
- -v  
image: node  
imagePullPolicy: Always  
name: nodeversion  
restartPolicy: Never

**Answer: A**

#### NEW QUESTION # 77

Create 2 nginx image pods in which one of them is labelled with `env=prod` and another one labelled with `env=dev` and verify the

same.

**Answer:**

Explanation:

```
kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml > nginx-prodpod.yaml Now, edit nginx-prod-pod.yaml file and remove entries like "creationTimestamp: null" "dnsPolicy: ClusterFirst" vim nginx-prod-pod.yaml
```

```
apiVersion: v1 kind: Pod metadata:
```

```
labels:  
env: prod  
name: nginx-prod
```

```
spec:  
containers:  
- image: nginx  
name: nginx-prod  
restartPolicy: Always
```

```
# kubectl create -f nginx-prod-pod.yaml
```

```
kubectl run --generator=run-pod/v1 --image=nginx --  
labels=env=dev nginx-dev --dry-run -o yaml > nginx-dev-pod.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
labels:  
env: dev  
name: nginx-dev
```

```
spec:  
containers:  
- image: nginx  
name: nginx-dev  
restartPolicy: Always
```

```
# kubectl create -f nginx-prod-dev.yaml
```

Verify :

```
kubectl get po --show-labels
```

```
kubectl get po -l env=prod
```

```
kubectl get po -l env=dev
```

**NEW QUESTION # 78**

You have a two-tier application with a frontend service 'frontend-svc' exposing a Node.js application running in pods labeled 'app: frontend', and a backend service 'backend-svc' exposing a Python application running in pods labeled 'app: backend'. The frontend pods need to communicate with the backend pods through a service. Design a Kubernetes network setup that allows the frontend pods to access the backend service, ensuring that the backend service is reachable only by the frontend service and not directly from outside the cluster.

**Answer:**

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Create a Service for the Backend:

- Define a Service for the backend pods:

□ - This Service creates a ClusterIP service, accessible only within the cluster. The 'targetPort' specifies the port exposed by the backend pods. 2. Create a Service for the Frontend: - Define a Service for the frontend pods:

□ - This Service creates a LoadBalancer service, accessible from outside the cluster. 3. Configure NetworkPolicy for the Frontend Service: - Define a NetworkPolicy that allows traffic from the 'frontend-svc' to the 'backend-svc':

□ - This NetworkPolicy allows ingress traffic from the 'frontend-svc' and egress traffic to the 'backend-svc'. 4. Apply the

Configurations: - Apply the YAML files using 'kubectl apply -f backend-svc.yaml' , 'kubectl apply -f frontend-svc.yaml' , and

kubectl apply -f frontend-to-backend.yaml. 5. Verification: - Check the status of the services: 'kubectl get services' - Check the

network policy status: 'kubectl get networkpolicies' Now, the frontend pods can communicate with the backend service through the

'backend-svc' service. External clients can access the frontend application through the 'frontend-svc' service. The backend service is not accessible directly from outside the cluster due to the NetworkPolicy restricting traffic from external sources. ]

## NEW QUESTION # 79

.....

Linux Foundation is one of the most powerful and rapidly growing fields nowadays. Everyone is trying to get the Linux Foundation CKA certification to improve their futures with it. Success in the test plays an important role in the up gradation of your CV and getting a good job or working online to achieve your dreams. The students are making up their minds for the Linux Foundation CKA test but they are mostly confused about where to prepare for it successfully on the first try.

**CKA Accurate Answers:** <https://www.vce4dumps.com/CKA-valid-torrent.html>

- Linux Foundation CKA Exam Questions: Attain Your Professional Career Goals [2026]  Immediately open  [www.vceengine.com](http://www.vceengine.com)   and search for  CKA  to obtain a free download  New CKA Test Pass4sure
- Preparation CKA Store  CKA Testdump  CKA Detailed Answers  Search for 「 CKA 」 and easily obtain a free download on ( [www.pdfvce.com](http://www.pdfvce.com) )  Test CKA Assessment
- Pass4sure CKA Study Materials Exam Latest Release | Updated Linux Foundation CKA: Certified Kubernetes Administrator (CKA) Program Exam  Simply search for  CKA  for free download on  [www.exam4labs.com](http://www.exam4labs.com)    CKA Valid Exam Forum
- New CKA Test Pass4sure  Test CKA Assessment  Pass4sure CKA Exam Prep  Open website  [www.pdfvce.com](http://www.pdfvce.com)  and search for  CKA   for free download  Best CKA Vce
- Pass4sure CKA Study Materials Exam Latest Release | Updated Linux Foundation CKA: Certified Kubernetes Administrator (CKA) Program Exam  Open website  [www.torrentvce.com](http://www.torrentvce.com)  and search for  CKA  for free download  Preparation CKA Store
- Free PDF 2026 Linux Foundation CKA: Trustable Pass4sure Certified Kubernetes Administrator (CKA) Program Exam Study Materials  Open  [www.pdfvce.com](http://www.pdfvce.com)  enter 「 CKA 」 and obtain a free download  Reliable CKA Exam Online
- Test CKA Assessment  CKA Free Test Questions  CKA Reliable Dumps  Download  CKA  for free by simply entering 「 [www.troytecdumps.com](http://www.troytecdumps.com) 」 website  New CKA Test Labs
- Get Free 365 Days Update on Linux Foundation CKA Dumps  Search for  CKA  and obtain a free download on { [www.pdfvce.com](http://www.pdfvce.com) }  New CKA Test Pass4sure
- CKA New APP Simulations  Dumps CKA Discount  Best CKA Vce  “ [www.dumpsquestion.com](http://www.dumpsquestion.com) ” is best website to obtain  CKA  for free download  New CKA Test Pass4sure
- Free PDF 2026 Linux Foundation CKA: Trustable Pass4sure Certified Kubernetes Administrator (CKA) Program Exam Study Materials  Search for  CKA  and download exam materials for free through  [www.pdfvce.com](http://www.pdfvce.com)    CKA Latest Exam Online
- Best CKA Vce  Reliable CKA Exam Voucher  New CKA Test Labs  Search for  CKA   and easily obtain a free download on  [www.testkingpass.com](http://www.testkingpass.com)   CKA Exam Reviews
- [skillkaro.com](http://skillkaro.com), [www.stes.tyc.edu.tw](http://www.stes.tyc.edu.tw), [ncon.edu.sa](http://ncon.edu.sa), [www.jkkdh.com](http://www.jkkdh.com), [1.gdsfjd.net](http://1.gdsfjd.net), [forum.phuongname.edu.vn](http://forum.phuongname.edu.vn), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [www.stes.tyc.edu.tw](http://www.stes.tyc.edu.tw), [www.stes.tyc.edu.tw](http://www.stes.tyc.edu.tw), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), [myportal.utt.edu.tt](http://myportal.utt.edu.tt), Disposable vapes

P.S. Free 2026 Linux Foundation CKA dumps are available on Google Drive shared by VCE4Dumps:  
[https://drive.google.com/open?id=15G0DGiUIyD\\_b6GaAj\\_i6iK7dxlemdzv](https://drive.google.com/open?id=15G0DGiUIyD_b6GaAj_i6iK7dxlemdzv)