High Certified Kubernetes Administrator (CKA) Program Exam passing score, CKA exam review

P.S. Free 2025 Linux Foundation CKA dumps are available on Google Drive shared by TestPDF: https://drive.google.com/open?id=17Fx7eI78EUwX-CfACElLZrSKvFTULyy2

High salary is everyone's dream. You salary is always based on your career competitive. In IT filed qualification is important. Our CKA questions and answers will help you hold opportunities and face difficulties bravely, then make a great achievement. Passing tests and get a certification is certainly a valid method that proves your competitions. CKA Questions and answers is surely helpful study guide for candidates all over the world.

The CKA program is designed for IT professionals who are interested in Kubernetes administration, including system administrators, DevOps engineers, and cloud administrators. Certified Kubernetes Administrator (CKA) Program Exam certification program covers a wide range of topics, including Kubernetes architecture and components, deploying applications on Kubernetes, configuring Kubernetes networking and security, and troubleshooting Kubernetes clusters.

The CKA certification exam consists of a performance-based test that requires candidates to demonstrate their ability to perform Kubernetes administrative tasks in a real-world environment. CKA Exam is conducted online and features a set of challenging scenarios that simulate common Kubernetes administration challenges. Candidates have two hours to complete the exam and must score at least 74% to pass. CKA exam is designed to test the candidate's knowledge of Kubernetes concepts and best practices, as well as their ability to troubleshoot issues and perform complex tasks.

>> CKA Free Download Pdf <<

Valid CKA Test Cost | CKA Test King

Our world is in the state of constant change and evolving. If you want to keep pace of the time and continually transform and challenge yourself you must attend one kind of CKA certificate test to improve your practical ability and increase the quantity of your knowledge. Buying our CKA Study Materials can help you pass the test smoothly. Our CKA study materials have gone through strict analysis and verification by senior experts and are ready to supplement new resources at any time.

The CKA certification is recognized globally and is highly valued by organizations that use Kubernetes in their production environments. Certified Kubernetes Administrator (CKA) Program Exam certification provides IT professionals with a competitive edge in the job market and opens up new career opportunities. The Linux Foundation offers comprehensive training and preparation materials to help candidates prepare for the CKA Certification Exam. The training includes hands-on labs, online courses, and study materials that cover all the topics and skills required for the exam.

Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q20-Q25):

NEW QUESTION # 20
List all the pods sorted by name

Answer:

Explanation:
kubect1 get pods --sort-by=.metadata.name

NEW QUESTION # 21
Score: 7%

Task
Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace echo. Ensure that the new NetworkPolicy allows Pods in namespace my-app to connect to port 9000 of Pods in namespace echo.
Further ensure that the new NetworkPolicy:
* does not allow access to Pods, which don't listen on port 9000
* does not allow access from Pods, which are not in namespace my-app

Answer:

Explanation:
See the solution below.
Explanation
Solution:
#network.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: internal
spec:
podSelector:
matchLabels: {
}
policyTypes:
- Ingress
ingress:
- from:
- podSelector: {
}
ports:
- protocol: TCP
port: 8080
#spec.podSelector namespace pod
kubectl create -f network.yaml

NEW QUESTION # 22
You have a Kubernetes cluster with two worker nodes and a single Nginx service deployed. You want to expose this service externally using a LoadBalancer service type but only want traffic to be directed to pods on a specific worker node. How would you achieve this?

Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a Node Selector:
- Create a Node Selector label on the worker node where you want to host the Nginx pods.
- Example:

```
apiVersion: v1
kind: Node
metadata:
  name: worker-node-1
spec:
  labels:
    worker-type: nginx
```

- Apply this configuration using 'kubectl apply -f node-config.yaml'. 2. Configure the Deployment: - Update the Nginx deployment to include the Node Selector label in its pod template. - Example:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
      nodeSelector:
        worker-type: nginx
```

- Apply the updated deployment configuration using 'kubectl apply -f nginx-deployment.yamr. 3. Create a LoadBalancer Service: - Create a LoadBalancer type service that selects the Nginx pods with the 'app=nginx' label. - Example:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: LoadBalancer
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
```

- Apply the service configuration using 'kubectl apply -f nginx-service.yamP. 4. Verify the Deployment: - Confirm the deployment of

the Nginx pods on the specified worker node using 'kubectl get pods -l app=nginx -o wide'. - Check the LoadBalancer service's external IP address using 'kubectl get services nginx-service'. - Access the Nginx service using the external IP address. All traffic should be routed to the pods on the worker node with the 'worker-type: nginx' label. ---

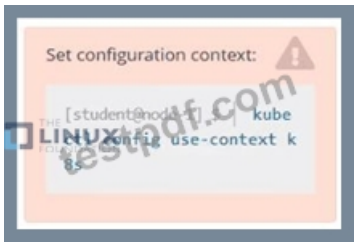
NEW QUESTION # 23
You must connect to the correct host.
Failure to do so may result in a zero score.
[candidate@base] $ ssh Cka000047
Task
A MariaDB Deployment in the mariadb namespace has been deleted by mistake. Your task is to restore the Deployment ensuring data persistence. Follow these steps:
Create a PersistentVolumeClaim (PVC) named mariadb in the mariadb namespace with the following specifications:
Access mode ReadWriteOnce
Storage 250Mi
You must use the existing retained PersistentVolume (PV).
Failure to do so will result in a reduced score.
There is only one existing PersistentVolume .
Edit the MariaDB Deployment file located at ~/mariadb-deployment.yaml to use PVC you created in the previous step.
Apply the updated Deployment file to the cluster.
Ensure the MariaDB Deployment is running and stable.

Answer:

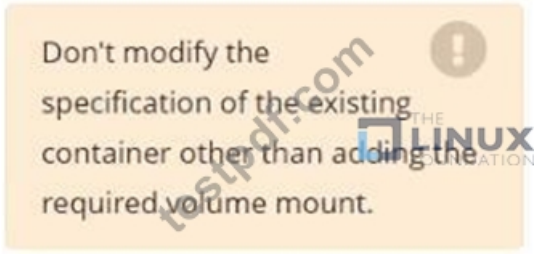
Explanation:
Task Overview
You're restoring a MariaDB deployment in the mariadb namespace with persistent data.
Tasks:
* SSH into cka000047
* Create a PVC named mariadb:
* Namespace: mariadb
* Access mode: ReadWriteOnce
* Storage: 250Mi
* Use the existing retained PV (there's only one)
* Edit ~/mariadb-deployment.yaml to use the PVC
* Apply the deployment
* Verify MariaDB is running and stable
Step-by-Step Solution
1## SSH into the correct host
ssh cka000047
Required - skipping = zero score
2## Inspect the existing PersistentVolume
kubectl get pv
Identify the only existing PV, e.g.:
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS
mariadb-pv 250Mi RWO Retain Available <none> manual
Ensure the status is Available, and it is not already bound to a claim.
3## Create the PVC to bind the retained PV
Create a file mariadb-pvc.yaml:
cat <<EOF > mariadb-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: mariadb
namespace: mariadb
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 250Mi

volumeName: mariadb-pv # Match the PV name exactly
EOF
Apply the PVC:
kubectl apply -f mariadb-pvc.yaml
This binds the PVC to the retained PV.
4## Edit the MariaDB Deployment YAML
Open the file:
nano ~/mariadb-deployment.yaml
Look under the spec.template.spec.containers.volumeMounts and spec.template.spec.volumes sections and update them like so:
Add this under the container:
yaml
CopyEdit
volumeMounts:
- name: mariadb-storage
mountPath: /var/lib/mysql
And under the pod spec:
volumes:
- name: mariadb-storage
persistentVolumeClaim:
claimName: mariadb
These lines mount the PVC at the MariaDB data directory.
5## Apply the updated Deployment
kubectl apply -f ~/mariadb-deployment.yaml
6## Verify the Deployment is running and stable
kubectl get pods -n mariadb
kubectl describe pod -n mariadb <mariadb-pod-name>
Ensure the pod is in Running state and volume is mounted.
Final Command Summary
ssh cka000047
kubectl get pv # Find the retained PV
Create PVC
cat <<EOF > mariadb-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: mariadb
namespace: mariadb
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 250Mi
volumeName: mariadb-pv
EOF
kubectl apply -f mariadb-pvc.yaml
Edit Deployment
nano ~/mariadb-deployment.yaml
Add volumeMount and volume to use the PVC as described
kubectl apply -f ~/mariadb-deployment.yaml
kubectl get pods -n mariadb

NEW QUESTION # 24
Score:7%



Context
An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e. g. kubectl logs).
Adding a streaming sidecar container is a good and common way to accomplish this requirement.
Task
Add a sidecar container named sidecar, using the busybox Image, to the existing Pod big-corp-app. The new sidecar container has to run the following command:
/bin/sh -c tail -n+1 -f /va r/log/big-corp-app.log
Use a Volume, mounted at /var/log, to make the log file big-corp-app.log available to the sidecar container.



Answer:

Explanation:
Solution:
#
kubectl get pod big-corp-app -o yaml
#
apiVersion: v1
kind: Pod
metadata:
name: big-corp-app
spec:
containers:
- name: big-corp-app
image: busybox
args:
- /bin/sh
- -c
- >
i=0;
while true;
do
echo "$(date) INFO $i" >> /var/log/big-corp-app.log;
i=$((i+1));
sleep 1;
done
volumeMounts:
- name: logs
mountPath: /var/log
- name: count-log-1
image: busybox
args: [/bin/sh, -c, 'tail -n+1 -f /var/log/big-corp-app.log']
volumeMounts:
- name: logs
mountPath: /var/log
volumes:

- name: logs
emptyDir: {
}
#
kubectl logs big-corp-app -c count-log-1

NEW QUESTION # 25
.....