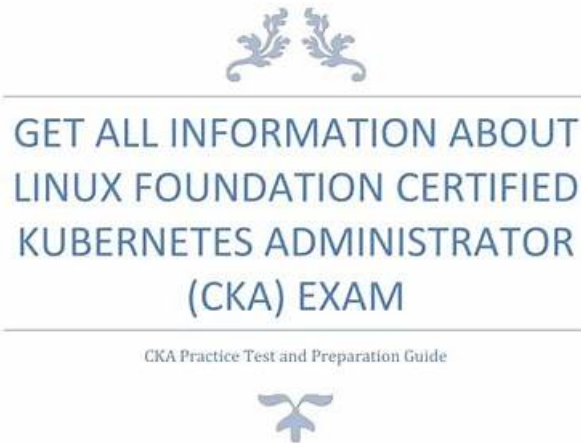


Reliable Linux Foundation CKA Exam Topics, New CKA Test Guide



GET COMPLETE DETAIL ON CKA EXAM GUIDE TO CRACK KUBERNETES V1.26. YOU CAN COLLECT ALL INFORMATION ON CKA TUTORIAL, PRACTICE TEST, BOOKS, STUDY MATERIAL, EXAM QUESTIONS, AND SYLLABUS. FIRM YOUR KNOWLEDGE ON KUBERNETES V1.26 AND GET READY TO CRACK CKA CERTIFICATION. EXPLORE ALL INFORMATION ON CKA EXAM WITH NUMBER OF QUESTIONS, PASSING PERCENTAGE AND TIME DURATION TO COMPLETE TEST.

P.S. Free & New CKA dumps are available on Google Drive shared by LatestCram: https://drive.google.com/open?id=1mE2Kf4g37ICI3-PT0x_9W2eJP6LRGIo8

Now, our CKA exam questions have gained wide popularity among candidates. Almost all customers are willing to introduce our CKA practice quiz to their classmates and friends. And sometimes, they may buy our exam products together. After they have tried our study materials, most of them have successfully passed the CKA Exam and made a lot of money. There are a lot of the feedbacks that they have left on our website to praise the good quality of our exam materials.

The CKA exam is a hands-on, performance-based exam that requires candidates to demonstrate their knowledge of Kubernetes administration through the use of practical tasks and challenges. CKA exam is designed to test an individual's ability to configure and manage Kubernetes clusters, troubleshoot common issues, and deploy applications using Kubernetes. CKA Exam is administered online and can be taken from anywhere in the world.

>> Reliable Linux Foundation CKA Exam Topics <<

Latest Updated Reliable CKA Exam Topics - Linux Foundation New Certified Kubernetes Administrator (CKA) Program Exam Test Guide

Likewise, Web-Based Linux Foundation CKA exam questions are supported by all the major browsers like Chrome, Opera, Safari, Firefox, and IE. In the same way, the Web-based Certified Kubernetes Administrator (CKA) Program Exam pdf exam requires no special plugin. Lastly, the web-based Certified Kubernetes Administrator (CKA) Program Exam (CKA) practice exam is customizable and requires an active Internet connection.

The Linux Foundation CKA exam consists of a set of performance-based tasks that are designed to test the candidate's ability to work with Kubernetes in a real-world environment. CKA exam is conducted in a supervised, online environment, and the candidate has access to a Kubernetes cluster during the exam. CKA Exam is designed to test a candidate's skills in various areas, including cluster architecture, deployment, security, troubleshooting, and automation.

Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q44-Q49):

NEW QUESTION # 44

You have a Deployment named 'wordpress-deployment' running a WordPress application with two pods. The WordPress application needs to access a shared secret key stored in a Secret named 'wordpress-secret'. How do you ensure the secret key is securely mounted and available to both pods?

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Create the Secret:

- Create a Secret named 'wordpress-secret' containing the shared secret key. Use 'kubectl create secret generic' with the '-from-literal' flag to create a generic Secret:

```
kubectl create secret generic wordpress-secret --from-literal=wordpress-secret="your_secret_key"
```

2. Configure the Deployment:

- Modify the 'wordpress-deployment' Deployment to mount the 'wordpress-secret' Secret as a volume in the Pod template. Use 'volumeMounts' to specify the mount path and 'volumes' to define the Secret as a volume source:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: wordpress
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
        - name: wordpress
          image: wordpress:latest
          volumeMounts:
            - name: wordpress-secret-volume
              mountPath: /var/run/secrets/wordpress
      volumes:
        - name: wordpress-secret-volume
          secret:
            secretName: wordpress-secret
```

3. Apply the Changes: - Apply the modified Deployment YAML using 'kubectl apply -f wordpress-deployment.yaml'. 4. Verify the Secret Mount: - Use 'kubectl exec' to access a pod in the 'wordpress-deployment' and verify that the Secret is mounted at the

specified path (e.g., 'Ivar/run/secrets/wordpress'). 5. Access the Secret Key: - Within the WordPress application's code, access the secret key from the mounted path.

NEW QUESTION # 45

Label a node as app=test and verify

Answer:

Explanation:

kubectl label node node-name app=test // Verify kubectl get no -show-labels kubectl get no -l app=test

NEW QUESTION # 46

Create a ETCD backup of kubernetes cluster

Note : You don't need to memorize command, refer -

<https://kubernetes.io/docs/tasks/administer-cluster/configureupgrade-etcd/> during exam

- A. `ETCDCTL_API=3 etcdctl --endpoints=[ENDPOINT] --cacert=[CA CERT] --cert=[ETCD SERVER CERT] --key=[ETCD SERVER KEY] snapshot save [BACKUP FILE NAME]`

In exam, cluster setup is done with kubeadm, this means ETCD used by the kubernetes cluster is coming from static pod.

`kubectl get pod -n kube-system`

`kubectl describe pod etcd-master -n kube-system`

You can locate the information on

endpoint: - `advertise-client-urls=https://172.16.0.18:2379`

ca certificate: - `trusted-cafile=/etc/kubernetes/pki/etcd/ca.crt`

server certificate : - `certfile=/etc/kubernetes/pki/etcd/server.crt`

key: - `key-file=/etc/kubernetes/pki/etcd/server.key`

To Create backup

`export ETCDCTL_API=3`

(or)

`ETCDCTL_API=3 etcdctl ETCDCTL_API=3 etcdctl --`

`endpoints=https://172.17.0.15:2379 --`

`key=/etc/kubernetes/pki/etcd/server.key snapshot save etcdsnapshot.db`

//Verify

`ETCDCTL_API=3 etcdctl --write-out=table snapshot status`

`snapshot.db`

- B. `ETCDCTL_API=3 etcdctl --endpoints=[ENDPOINT] --cacert=[CA CERT] --cert=[ETCD SERVER CERT] --key=[ETCD SERVER KEY] snapshot save [BACKUP FILE NAME]`

In exam, cluster setup is done with kubeadm, this means ETCD used by the kubernetes cluster is coming from static pod.

`kubectl get pod -n kube-system`

`kubectl describe pod etcd-master -n kube-system`

You can locate the information on

endpoint: - `advertise-client-urls=https://172.17.0.15:2379`

ca certificate: - `trusted-cafile=/etc/kubernetes/pki/etcd/ca.crt`

server certificate : - `certfile=/etc/kubernetes/pki/etcd/server.crt`

key: - `key-file=/etc/kubernetes/pki/etcd/server.key`

To Create backup

`export ETCDCTL_API=3`

(or)

`ETCDCTL_API=3 etcdctl ETCDCTL_API=3 etcdctl --`

`endpoints=https://172.17.0.15:2379 --`

`cacert=/etc/kubernetes/pki/etcd/ca.crt --`

`cert=/etc/kubernetes/pki/etcd/server.crt --`

`key=/etc/kubernetes/pki/etcd/server.key snapshot save etcdsnapshot.db`

//Verify

ETCDCTL_API=3 etcdctl --write-out=table snapshot status
snapshot.db

Answer: B

NEW QUESTION # 47

You have a Deployment named 'my-app-deployment' running a complex microservice application with several pods. You suspect that some pods are experiencing high CPU utilization, causing performance issues. How would you monitor and troubleshoot the CPU usage of individual pods within this deployment?

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Identify the Deployment Pods:

- Use 'kubectl get pods -l app=my-app' to list the pods associated with the 'my-app-deployment' deployment.

Note the pod names.

2. Check CPU Utilization:

- Use 'kubectl top pods' to check the real-time CPU usage of the pods you identified in step 1 .

- Alternatively, use 'kubectl describe pod ' to see the average CPU usage over time.

3. Analyze CPU Metrics:

- Look for pods with consistently high CPU usage (e.g., 80% or more).

- Identify the specific containers within the pod consuming the most CPU resources. You can use 'kubectl top pod -c' for this.

4. Investigate the Cause:

- Review the pod logs: Use 'kubectl logs' to examine the logs of the affected pods. Look for error messages, unusual patterns, or indications of resource-intensive operations.

- Examine the application code: If possible, access the source code of the application running within the affected pods. Analyze the code for potential CPU-intensive operations or bottlenecks.

- Check resource limits: Ensure that the resource requests and limits defined for the containers in the pod are appropriate. If the limits are too low, the containers might be throttled, leading to high CPU usage. Use 'kubectl describe pod' to check resource limits.

5. Possible Solutions:

- Increase resource limits: If the containers are being throttled, increase their CPU limits in the Deployment YAML.

- Optimize application code: If the application code is inefficient, you might need to optimize it to reduce CPU consumption.

- Scale down pods: If the high CPU usage is due to a temporary spike in demand, consider scaling down the number of pods in the deployment.

- Use a horizontal pod autoscaler (HPA): An HPA can automatically adjust the number of pods based on CPU utilization metrics.

6. Monitor and Iterate:

- After implementing any changes, monitor the CPU usage of the pods to ensure that the issue is resolved.

- Repeat this process of monitoring, troubleshooting, and optimization until the CPU usage is within acceptable levels.

NEW QUESTION # 48

Check the image version in pod without the describe command

Answer:

Explanation:

See the solution below.

Explanation

kubectl get po nginx -o

jsonpath='{.spec.containers[].image}' {"n"}'

NEW QUESTION # 49

.....

New CKA Test Guide: <https://www.latestcram.com/CKA-exam-cram-questions.html>

- 2025 Latest LatestCram CKA PDF Dumps and CKA Exam Engine Free Share: https://drive.google.com/open?id=1mE2Kf4g37ICI3-PT0x_9W2eJP6LRGIo8

2025 Latest LatestCram CKA PDF Dumps and CKA Exam Engine Free Share: https://drive.google.com/open?id=1mE2Kf4g37ICI3-PT0x_9W2eJP6LRGIo8