

Reliable CKAD Exam Sims, CKAD Latest Exam Questions

11/14/2019

Practice Enough With These 150 Questions for the CKAD Exam

Practice Enough With These 150 Questions for the CKAD Exam

Exercises get you ready for the Certified Kubernetes Application Developer exam



Bhargav Bachina [Follow](#)

Nov 11 · 21 min read ★



Photo by Tim Foster on Unsplash

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. The CNCF/Linux Foundation offers this performance-based exam which targets the developer aspect of kubernetes skills such as

<https://medium.com/@b-tutorials-and-thoughts/practice-enough-with-these-questions-for-the-ckad-exam-29421228552>

1/40

2026 Latest PassLeader CKAD PDF Dumps and CKAD Exam Engine Free Share: https://drive.google.com/open?id=1JJFTV_HUgIVcv2A83BP3pXoPjlg_RU5k

Different from other similar education platforms, the CKAD study materials will allocate materials for multi-plate distribution, rather than random accumulation without classification. How users improve their learning efficiency is greatly influenced by the scientific and rational design and layout of the learning platform. The CKAD study materials are absorbed in the advantages of the traditional learning platform and realize their shortcomings, so as to develop the CKAD Study Materials more suitable for users of various cultural levels. If just only one or two plates, the user will inevitably be tired in the process of learning on the memory and visual fatigue, and the CKAD study materials provided many study parts of the plates is good enough to arouse the enthusiasm of the user, allow the user to keep attention of highly concentrated.

Our PassLeader's CKAD test training materials can test your knowledge, when you prepare for CKAD test; and can also evaluate your performance at the appointed time. Our CKAD exam training materials is the result of PassLeader's experienced IT experts with constant exploration, practice and research for many years. Its authority is undeniable. If you have any concerns, you can first try CKAD PDF VCE free demo and answers, and then make a decision whether to choose our CKAD dumps or not.

>> **Reliable CKAD Exam Sims** <<

100% Pass 2026 Linux Foundation CKAD Fantastic Reliable Exam Sims

As we all know, it is difficult to prepare the CKAD exam by ourselves. Excellent guidance is indispensable. If you urgently need

help, come to buy our study materials. Our company has been regarded as the most excellent online retailers of the CKAD exam question. So our assistance is the most professional and superior. You can totally rely on our study materials to pass the exam. All the key and difficult points of the CKAD exam have been summarized by our experts. They have rearranged all contents, which is convenient for your practice. Perhaps you cannot grasp all crucial parts of the CKAD Study Tool by yourself. You also can refer to other candidates' review guidance, which might give you some help. Then we can offer you a variety of learning styles. Our printable CKAD real exam dumps, online engine and windows software are popular among candidates. So you will never feel bored when studying on our CKAD study tool.

Linux Foundation Certified Kubernetes Application Developer Exam Sample Questions (Q95-Q100):

NEW QUESTION # 95

You are building a new web application that utilizes a microservice architecture- One of the microservices, 'recommendation-service', is responsible for providing personalized product recommendations to users.

This service uses a machine learning model for generating recommendations based on user purchase history and browsing behavior.

The model is trained offline and its weights are stored in a 'model-store' service.

Design a multi-container Pod for the 'recommendation-service' that incorporates the following considerations:

- The Pod should include a primary container for the 'recommendation-service' application.
- The Pod should include a secondary container that runs the 'model-store' service to provide access to the trained model weights.
- Both containers should share a common volume to ensure that the model weights are available to the 'recommendation-service' container-
- The recommendation-service' should be able to access the model weights from the 'model-store' container without relying on a network call to another service-
- The recommendation-service' container should be configured to periodically update the model weights from the 'model-store' container when a new version of the model is available.

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Create the Deployment YAML:

- Define a Deployment with the name 'recommendation-service'
- Set the replicas to for redundancy and scalability.
- Specify the labels `Sapp: recommendation-service'` for selecting the Pods in the Deployment.
- Create a 'template' section to define the Pod specification

2. Deploy the Resources: - Apply the Deployment using 'kubectl apply -f deployment-yaml' 3. Verify the Deployment: - Check the status of the Deployment using 'kubectl get deployments recommendation-service' and ensure that three Pods are running. 4. Configure the 'recommendation-service' - Modify the 'recommendation-service' application to load the model weights from the specified path `C:\mode\latest-model\weightS`. - Implement a mechanism within the 'recommendation-service' to periodically check for updated model weights in the shared volume. 5. Configure the 'model-store' service: - Ensure that the model-store service is properly configured to store and retrieve the model weights. - Implement a mechanism in the 'model-store' service to notify the 'recommendation-service' when a new model version is available. This notification can be achieved using a shared volume or a separate messaging system. 6. Test the Application: - Send requests to the 'recommendation-service' to generate recommendations. - Monitor the 'model-store' service and the shared volume to verify that the model weights are being updated correctly and the recommendation-service' is using the latest model version. Important Considerations: - Ensure that the 'recommendation-service' application is properly configured to access and load the model weights from the shared volume. - Implement a robust model management strategy, including versioning and rollback mechanisms, to ensure that the recommendation-service always uses the appropriate model. - Consider using a dedicated model store service that provides a dedicated API for retrieving and updating model weights. This can simplify the communication between the 'recommendation-service' and the model store. - Monitor the performance and resource usage of both services to ensure optimal performance.,

NEW QUESTION # 96

Refer to Exhibit.

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:

- * The pod must run in the web namespace.

The namespace has already been created

- * The name of the pod should be cache

- * Use the Ifccncf/redis image with the 3.2 tag

- * Expose port 6379

Answer:

Explanation:

Solution:

□

NEW QUESTION # 97

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] \$ ssh ckad00021

Task

Create a Cronjob named grep that executes a Pod running the following single container:

name: busybox

image: busybox:stable

command: ["grep", "-i", "nameserv", "/etc/resolv.conf"]

Configure the CronJob to:

- * execute Once every 30 minutes

- * keep 96 completed Job

- * keep 192 failed Job

- * never restart podsterminate pods after 8 seconds

Manually create and execute once job

named grep-test from the grep Cronjob

Answer:

Explanation:

See the Explanation below for complete solution.

Explanation:

ssh ckad00021

Below is the clean, CKAD-friendly way (YAML + apply + verify + manual job).

1) Create the CronJob grep

Create a file (anywhere, e.g. in your home):

```
cat <<'EOF' > grep-cronjob.yaml
```

```
apiVersion: batch/v1
```

```
kind: CronJob
```

```
metadata:
```

```
name: grep
```

```
spec:
```

```
schedule: "*/30 * * * *"
```

```
successfulJobsHistoryLimit: 96
```

```
failedJobsHistoryLimit: 192
```

```
jobTemplate:
```

```
spec:
```

```
activeDeadlineSeconds: 8
```

```
template:
```

```
spec:
```

```
restartPolicy: Never
```

```
containers:
```

```
- name: busybox
```

```
image: busybox:stable
```

```
command: ["grep", "-i", "nameserver", "/etc/resolv.conf"]
```

```
EOF
```

Apply it:

```
kubectl apply -f grep-cronjob.yaml
```

Verify:

```
kubectl get cronjob grep
```

```
kubectl describe cronjob grep
```

Confirm the key fields quickly:

```
kubectl get cronjob grep -o jsonpath='{.spec.schedule} {"\n"} {.spec.successfulJobsHistoryLimit} {"\n"} {.spec.failedJobsHistoryLimit} {"\n"}'
```

```
kubectl get cronjob grep -o jsonpath='{.spec.jobTemplate.spec.activeDeadlineSeconds} {"\n"} {.spec.jobTemplate.spec.template.spec.restartPolicy} {"\n"}'
```

2) Manually create and execute the one-off Job grep-test from the CronJob Create the Job from the CronJob:

```
kubectl create job --from=cronjob/grep grep-test
```

Watch it:

```
kubectl get jobs grep-test
```

```
kubectl get pods -l job-name=grep-test
```

Get logs (most important proof):

```
POD=$(kubectl get pods -l job-name=grep-test -o jsonpath='{.items[0].metadata.name}') kubectl logs "$POD" You should see one or more nameserver ... lines from /etc/resolv.conf.
```

NEW QUESTION # 98

Context

Task:

1- Update the Horizontal scaling configuration of the Deployment web1 in the ckad00015 namespace setting maxSurge to 2 and maxUnavailable to 59

2- Update the web1 Deployment to use version tag 1.13.7 for the Ifconf/nginx container image.

3- Perform a rollback of the web1 Deployment to its previous version

Answer:

Explanation:

Solution:

☐

NEW QUESTION # 99

Exhibit:

Task

You are required to create a pod that requests a certain amount of CPU and memory, so it gets scheduled to a node that has those resources available.

* Create a pod named nginx-resources in the pod-resources namespace that requests a minimum of 200m CPU and 1Gi memory for its container

* The pod should use the nginx image

* The pod-resources namespace has already been created

- **A. Solution:**

☐

☐

☐

☐

☐

- **B. Solution:**

☐

☐

☐

☐

Answer: A

