

Ace Your Linux Foundation CKAD Exam With Web-based Practice Tests



```
student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS    RESTARTS   AGE
cache     1/1     Running   0           9s
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME      TYPE      DATA   AGE
default-token-4kvr5   kubernetes.io/service-account-token   3       2d11h
some-secret           Opaque                                 1       3s
student@node-1:~$ kubectl run nginx-secret --image=nginx --dry-run=client -o yaml > nginx_secret.yaml
student@node-1:~$ vim nginx_secret.yaml
student@node-1:~$ kubectl create -f nginx_secret.yaml
pod/nginx-secret created
student@node-1:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
liveness-http  1/1     Running   0           6h38m
nginx-101     1/1     Running   0           6h39m
nginx-secret  0/1     ContainerCreating   0           4s
poller       1/1     Running   0           6h39m
student@node-1:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
liveness-http  1/1     Running   0           6h38m
nginx-101     1/1     Running   0           6h39m
nginx-secret  1/1     Running   0           8s
poller       1/1     Running   0           6h39m
student@node-1:~$
```

What's more, part of that Itbraindumps CKAD dumps now are free: https://drive.google.com/open?id=1VXoX3GsC0WXIfYWA1_ub5PE-vrSIVc5W

Get the test CKAD certification is not achieved overnight, we need to invest a lot of time and energy to review, and the review process is less a week or two, more than a month or two, or even half a year, so CKAD exam questions are one of the biggest advantage is that it is the most effective tools for saving time for users. Users do not need to spend too much time on CKAD questions torrent, only need to use their time pieces for efficient learning, the cost is about 20 to 30 hours, users can easily master the test key and difficulties of questions and answers of CKAD Prep Guide, and in such a short time acquisition of accurate examination skills, better answer out of step, so as to realize high pass the qualification test, has obtained the corresponding qualification certificate.

Are you considering the questions that how you can pass the CKAD exam and get a certificate? The best answer is to download and learn our CKAD quiz torrent. Our CKAD exam questions will help you get what you want in a short time. You just need little time to download and install it after you purchase our CKAD training prep, then you just need spend about 20~30 hours to learn it. We are glad that you are going to spare your precious time to have a look to our CKAD exam guide.

>> CKAD Exam Pass Guide <<

CKAD test-preparation routine proven to help you pass the exams

The second format is a web-based practice exam which offers a flexible and accessible option for students trying to assess and improve their preparation for the Linux Foundation Certification Exams. The CKAD web-based practice test can be accessed online through browsers like Firefox, Microsoft Edge, Google Chrome, and Safari. Customers need a stable internet connection in order to access web-based formats easily without facing issues.

Linux Foundation CKAD certification exam is an online, proctored exam that can be taken from anywhere in the world. CKAD exam consists of a set of performance-based tasks that are designed to test the candidate's ability to use Kubernetes to deploy, manage and troubleshoot containerized applications. CKAD exam is timed and candidates have 2 hours to complete the exam. To pass the exam, candidates must score 66% or higher.

Linux Foundation CKAD Exam is an excellent opportunity for developers to demonstrate their Kubernetes skills and gain recognition for their expertise. Whether you are a seasoned Kubernetes professional or just starting out, this certification can help you take your career to the next level and open up new opportunities in the rapidly evolving world of cloud-native development.

Linux Foundation Certified Kubernetes Application Developer Exam Sample Questions (Q84-Q89):

NEW QUESTION # 84

You are running a web application on Kubernetes, and you need to schedule a daily backup of your application's data. The backup

process involves running a script that archives the database and uploads it to a remote storage service. Explain how you would implement this using a CronJob in Kubernetes. Provide a sample YAML configuration for the Cronjob.

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Create a Container Image:

- Create a Dockerfile that packages the backup script and any necessary dependencies into a container image-
- Build the image and push it to a container registry (e.g., Docker Hub).

dockerfile

FROM ubuntu:latest

Install necessary packages

RUN apt-get update && apt-get install -y zip

Copy the backup script to the container

COPY backup.sh 'backup.sh'

Set entrypoint to the backup script

ENTRYPOINT ["/backup.sh"]

2. Create the CronJob YAML:

- Define the 'cronJob' resource in a YAML file. Specify the schedule, the container image, and the command to run.

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: daily-backup
spec:
  schedule: "0 0 * * *" # Run at midnight every day
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: backup-container
              image: your-registry.com/your-username/backup-image:latest
              command: ["/backup.sh"]
          backoffLimit: 4
          parallelism: 1
```

3. Apply the CronJob: - Apply the YAML file to your Kubernetes cluster using 'kubectl apply -f cronjob.yaml' 4. Verification: - Use 'kubectl get cronjobs' to check if the CronJob was successfully created- - Check the Kubernetes logs for the CronJob to ensure it's running as expected.

NEW QUESTION # 85

You are tasked with deploying a stateful application, a distributed database, that requires persistent storage and consistent ordering of pods. The application's pods need to communicate With each other using a specific port (5432). How would you configure a StatefulSet to achieve this?

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Create the StatefulSet YAML:

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: my-database
spec:
  serviceName: "my-database"
  replicas: 3
  selector:
    matchLabels:
      app: my-database
  template:
    metadata:
      labels:
        app: my-database
    spec:
      containers:
        - name: database
          image: postgres:14.2
          ports:
            - containerPort: 5432
          volumeMounts:
            - name: data
              mountPath: /var/lib/postgresql/data
      volumes:
        - name: data
          persistentVolumeClaim:
            claimName: my-database-pvc

```

2. Create a PersistentVolumeClaim (PVC):

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-database-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

```

3. Apply the StatefulSet and PVC: `bash kubectl apply -f statefulset.yaml kubectl apply -f pvc.yaml` 4. Check the StatefulSet and Pods: `bash kubectl get statefulsets my-database kubectl get pods -l app=my-database`

StatefulSet This defines the desired state for the database pods, ensuring their order and persistent storage.

- **serviceName**: This field defines the service name used to access the database instances.
- **replicas**: Defines the desired number of database instances (3 in this example).
- **selector**: Matches pods with the "app: my-database" label.
- **template**: Defines the pod template to use for each instance.
- **containers**: Contains the database container definition.
- **ports**: Exposes the database's internal port (5432) to the outside world.
- **volumeMounts**: Mounts the persistent volume claim to the container's storage directory.
- **volumes**: Defines the volume to use, in this case, a persistent volume claim.
- **persistentVolumeClaim**: Links the StatefulSet to the PVC.

PVC (my-database-pvc): Requests a persistent volume of 1 Gi for each database pod. This ensures data persistence between restarts.

- **accessModes**: `ReadWriteOnce`: Allows only one pod to access the volume at a time.
- **resources-requests-storage**: Specifies the storage request for each PVC. This setup ensures that each database pod:
- Has a unique name based on its ordinal position within the StatefulSet
- Has persistent storage using the PVC.
- Can communicate with other pods through the defined service.
- Maintains consistent ordering, essential for distributed database functionality

NEW QUESTION # 86

You have a multi-container Pod that runs a web server (Nginx) and a database (MySQL) container. The database container requires data to be initialized before the web server container can start. How would you configure the Pod to ensure the database container is initialized before the web server container starts?

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Use initContainers:

- Define one or more 'initContainers' within the Pod's 'spec.template.spec' section.
- The 'initContainers' will run before any other container in the Pod.
- In this case, you would create an 'initcontainer' for the MySQL database.

2. Configure the initContainer:

- The 'initcontainer' should have the following attributes:
- Name: A unique name for the container.
- Image: The Docker image containing the necessary tools to initialize the database.
- Command: The command to execute for database initialization.
- LivenessProbe: Optional, but recommended to check if the database initialization process is successful.

3. Sequence the containers:

- Ensure the 'initContainers' are listed before the main containers in the Pod's 'spec-template-spec-containers' section.

4. Example YAML:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: nginx
      image: nginx:latest
      ports:
        - containerPort: 80
    - name: mysql
      image: mysql:latest
      ports:
        - containerPort: 3306
  initContainers:
    - name: mysql-init
      image: mysql:latest
      command: ["sh", "-c", "mysql -u root -e 'CREATE DATABASE mydatabase;']"]
      livenessProbe:
        tcpSocket:
          port: 3306
        initialDelaySeconds: 10
        periodSeconds: 5
```

- The 'mysql-init' 'initcontainer' will run before the 'nginx' and 'mysql' containers-
- The 'command' in the 'initContainer' Will create a database named within the MySQL container.
- The livenessprobe' will ensure that the database is reachable on port 3306 after the initialization process completes. Note: This solution assumes that the 'mysql' image already includes the necessary database initialization tools. You might need to use a custom image with these tools if the default image doesn't provide them,

NEW QUESTION # 87

Context



Task

Create a new deployment for running nginx with the following parameters;

- * Run the deployment in the kdpd00201 namespace. The namespace has already been created
- * Name the deployment frontend and configure with 4 replicas
- * Configure the pod with a container image of lfcncf/nginx:1.13.7
- * Set an environment variable of NGINX__PORT=8080 and also expose that port for the container above

Answer:

Explanation:

Solution:

```
student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml
student@node-1:~$ vim nginx_deployment.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: api
    name: api
    namespace: kdpd00201
spec:
  replicas: 4
  selector:
    matchLabels:
      app: api
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: api
    spec:
      containers:
      - image: lfcncf/nginx:1.13.7-alpine
        name: nginx
        resources: {}
status: {}
```

"nginx_deployment.yml" 25L, 421C

4,1

All

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: api
    name: api
    namespace: kdpd00201
spec:
  replicas: 4
  selector:
    matchLabels:
      app: api
  template:
    metadata:
      labels:
        app: api
    spec:
      containers:
      - image: lfcncf/nginx:1.13.7-alpine
        name: nginx
        ports:
        - containerPort: 8080
        env:
        - name: NGINX_PORT
          value: "8080"
```

23,8

All

```
Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml
student@node-1:~$ vim nginx_deployment.yml
student@node-1:~$ kubectl create nginx_deployment.yml
Error: must specify one of -f and -k

error: unknown command "nginx_deployment.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_deployment.yml
error: error validating "nginx_deployment.yml": error validating data: ValidationError(Deployment.spec.template.spec): unknown field "env" in io.k8s.api.core.v1.PodSpec; if you choose to ignore these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_deployment.yml
student@node-1:~$ kubectl create -f nginx_deployment.yml
deployment.apps/api created
student@node-1:~$ kubectl get pods -n kdpd00201
NAME                                READY   STATUS    RESTARTS   AGE
api-745677f7dc-7hnmv               1/1     Running   0           13s
api-745677f7dc-9q5vp               1/1     Running   0           13s
api-745677f7dc-fd4gk               1/1     Running   0           13s
api-745677f7dc-mbnpc               1/1     Running   0           13s
student@node-1:~$
```

NEW QUESTION # 88



Context

A pod is running on the cluster but it is not responding.

Task

The desired behavior is to have Kubernetes restart the pod when an endpoint returns an HTTP 500 on the /healthz endpoint. The service, probe-pod, should never send traffic to the pod while it is failing. Please complete the following:

- * The application has an endpoint, /started, that will indicate if it can accept traffic by returning an HTTP 200.

If the endpoint returns an HTTP 500, the application has not yet finished initialization.

- * The application has another endpoint /healthz that will indicate if the application is still working as expected by returning an HTTP 200. If the endpoint returns an HTTP 500 the application is no longer responsive.

- * Configure the probe-pod pod provided to use these endpoints

- * The probes should use port 8080

Answer:

Explanation:

See the solution below.

Explanation

Solution:

apiVersion: v1

kind: Pod

metadata:

labels:

test: liveness

name: liveness-exec

spec:

```

containers:
- name: liveness
  image: k8s.gcr.io/busybox
  args:
  - /bin/sh
  - -c
  - touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600
livenessProbe:
  exec:
  command:
  - cat
  - /tmp/healthy
  initialDelaySeconds: 5
  periodSeconds: 5

```

In the configuration file, you can see that the Pod has a single Container. The periodSeconds field specifies that the kubelet should perform a liveness probe every 5 seconds. The initialDelaySeconds field tells the kubelet that it should wait 5 seconds before performing the first probe. To perform a probe, the kubelet executes the command `cat /tmp/healthy` in the target container. If the command succeeds, it returns 0, and the kubelet considers the container to be alive and healthy. If the command returns a non-zero value, the kubelet kills the container and restarts it.

When the container starts, it executes this command:

```
/bin/sh -c "touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600"
```

For the first 30 seconds of the container's life, there is a `/tmp/healthy` file. So during the first 30 seconds, the command `cat /tmp/healthy` returns a success code. After 30 seconds, `cat /tmp/healthy` returns a failure code.

Create the Pod:

```
kubectl apply -f https://k8s.io/examples/pods/probe/exec-liveness.yaml
```

Within 30 seconds, view the Pod events:

```
kubectl describe pod liveness-exec
```

The output indicates that no liveness probes have failed yet:

```
FirstSeen LastSeen Count From SubobjectPath Type Reason Message
```

```

-----
24s 24s 1 {default-scheduler} Normal Scheduled Successfully assigned liveness-exec to worker0
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "k8s.gcr.io/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image
"k8s.gcr.io/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id
86849c15382e; Security:[seccomp=unconfined]
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id
86849c15382e

```

After 35 seconds, view the Pod events again:

```
kubectl describe pod liveness-exec
```

At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated.

```
FirstSeen LastSeen Count From SubobjectPath Type Reason Message
```

```

-----
37s 37s 1 {default-scheduler} Normal Scheduled Successfully assigned liveness-exec to worker0
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "k8s.gcr.io/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image
"k8s.gcr.io/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id
86849c15382e; Security:[seccomp=unconfined]
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id
86849c15382e

```

```

2s 2s 1 {kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can't open
'/tmp/healthy': No such file or directory

```

Wait another 30 seconds, and verify that the container has been restarted:

```
kubectl get pod liveness-exec
```

The output shows that RESTARTS has been incremented:

```
NAME READY STATUS RESTARTS AGE
```

```
liveness-exec 1/1 Running 1 1m
```

NEW QUESTION # 89

.....

There is a ton of Linux Foundation Certified Kubernetes Application Developer Exam (CKAD) prep material available on the internet. But the main thing to notice is their validity and reliability. Many applicants remain unsuccessful in locating the right Linux Foundation Certified Kubernetes Application Developer Exam (CKAD) practice test and lose their time and money.

CKAD Examcollection Questions Answers: https://www.itbraindumps.com/CKAD_exam.html

- High Quality CKAD Test Materials - Linux Foundation Certified Kubernetes Application Developer Exam Qualification Dump ☐ Search for ➤ CKAD ☐ and download exam materials for free through (www.dumpsquestion.com) ☐ ☐ New CKAD Test Labs
- CKAD Reliable Exam Question ☐ CKAD Exam Forum ☐ Latest CKAD Exam Pass4sure ☐ Open 「 www.pdfvce.com 」 enter ➡ CKAD ☐ and obtain a free download ☐ CKAD Learning Engine
- 2025 CKAD – 100% Free Exam Pass Guide | Reliable CKAD Examcollection Questions Answers ☐ Go to website ☐ www.lead1pass.com ☐ open and search for ➤ CKAD ◁ to download for free ☐ CKAD Exam Forum
- 100% Pass 2025 Linux Foundation High-quality CKAD: Linux Foundation Certified Kubernetes Application Developer Exam Exam Pass Guide ☐ Open website ☐ www.pdfvce.com ☐ and search for “CKAD” for free download ☐ CKAD Reliable Exam Question
- Free PDF Quiz 2025 Linux Foundation CKAD: Linux Foundation Certified Kubernetes Application Developer Exam Useful Exam Pass Guide ☐ Enter 《 www.examcollectionpass.com 》 and search for ➡ CKAD ☐ to download for free ☐ ☐ CKAD Reliable Exam Question
- Free PDF Quiz 2025 Linux Foundation CKAD: Linux Foundation Certified Kubernetes Application Developer Exam Useful Exam Pass Guide ☐ Go to website [www.pdfvce.com] open and search for ➡ CKAD ☐ ☐ ☐ to download for free ☐ ☐ Exam CKAD Materials
- New CKAD Practice Questions ☐ CKAD Learning Engine ☐ CKAD Latest Real Exam ☐ The page for free download of 「 CKAD 」 on ► www.getvalidtest.com ◀ will open immediately ☐ CKAD Test Collection
- Reliable CKAD Test Question ☐ Valid Test CKAD Tutorial ☐ Exam CKAD Materials ☐ Go to website ✨ www.pdfvce.com ☐ ✨ ☐ open and search for ✓ CKAD ☐ ✓ ☐ to download for free ☐ CKAD Learning Engine
- CKAD Training Materials - CKAD Study Materials - CKAD Exam Torrent ☐ Go to website ⇒ www.real4dumps.com ⇐ open and search for 「 CKAD 」 to download for free ☐ CKAD Learning Engine
- CKAD Exam Exam Pass Guide - Trustable CKAD Examcollection Questions Answers Pass Success ☐ Easily obtain free download of ☐ CKAD ☐ by searching on [www.pdfvce.com] ☐ CKAD Reliable Test Test
- CKAD Latest Real Exam ☐ CKAD Reliable Test Test ☐ CKAD Updated Demo ☐ Go to website ☐ www.prep4sures.top ☐ open and search for ➤ CKAD ☐ to download for free ✨ CKAD Reliable Exam Question
- www.stes.tyc.edu.tw, ncon.edu.sa, developer.codesys.cn, study.stes.edu.np, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, motionentrance.edu.np, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes

BTW, DOWNLOAD part of Itbraindumps CKAD dumps from Cloud Storage: https://drive.google.com/open?id=1VXoX3GsC0WXIfYWA1_ub5PE-vrSIVc5W