Linux Foundation CKA Exam | CKA Labs - Help you Prepare CKA: Certified Kubernetes Administrator (CKA) Program Exam Exam Easily

BONUS!!! Download part of PremiumVCEDump CKA dumps for free: https://drive.google.com/open?id=1IYWK_p1vcv52hiuKodnYbpVBk4nFRFR5

The clients at home and abroad strive to buy our CKA study materials because they think our products are the best study materials which are designed for preparing the test Linux Foundation certification. They trust our CKA study materials deeply not only because the high quality and passing rate of our CKA study materials but also because our considerate service system. They treat our CKA Study Materials as the magic weapon to get the Linux Foundation certificate and the meritorious statesman to increase their wages and be promoted. You may be not quite familiar with our CKA study materials and we provide the detailed explanation of our CKA study materials as follow for you have an understanding before you decide to buy.

The CKA certification exam aims to test the candidate's practical skills in deploying and managing Kubernetes clusters. CKA exam consists of a performance-based test, where the candidate must complete a set of tasks in a live Kubernetes cluster within a specified time. CKA exam covers various topics such as Kubernetes architecture, installation and configuration, networking, security, scheduling, storage, and troubleshooting. Passing the CKA Certification Exam demonstrates the candidate's ability to manage Kubernetes clusters efficiently and effectively.

>> CKA Labs <<

CKA Free Practice Exams | CKA Reliable Test Dumps

People who want to pass CKA exam also need to have a good command of the newest information about the coming CKA exam. However, it is not easy for a lot of people to learn more about the information about the study materials. Luckily, the CKA preparation materials from our company will help all people to have a good command of the newest information. Because our company have employed a lot of experts and professors to renew and update the CKA test training guide for all customer in order to provide all customers with the newest information.

Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q79-Q84):

NEW QUESTION # 79
You are deploying an application on Kubernetes. You need to ensure that a minimum of three pods are always running for this application. How can you achieve this? Describe how to configure the deployment with a replica count and a liveness probe to monitor the health of the pods.

Answer:

Explanation:

See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a Deployment with a Replica Count:
- Create a YAML file named 'deployment.yamr with the following content:

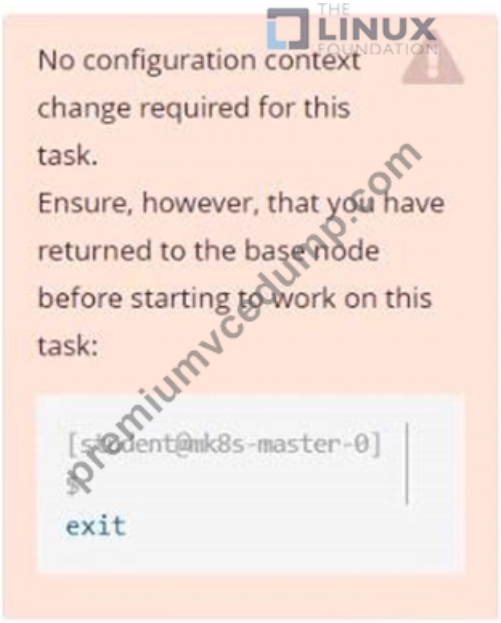
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
      - name: myapp
        image: myapp:latest
        # ... other container configurations ...
```


- Apply the YAML file using 'kubectl apply -f deployment.yaml&. 2. Configure a Liveness Probe: - Update the 'deployment.yaml' file to include a liveness probe. For example, you could use a HTTP probe:

```
# ... other deployment configurations ...
      containers:
      - name: myapp
        image: myapp:latest
        # ... other container configurations ...
        livenessProbe:
          httpGet:
            path: /health
            port: 8080
          initialDelaySeconds: 15
          periodSeconds: 20
          failureThreshold: 3
```

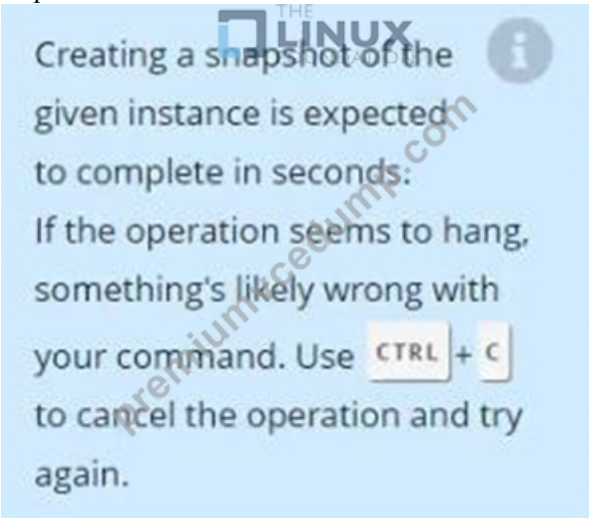

- Apply the updated YAML file using 'kubectl apply -f deployment.yaml'. 3. Verify the Deployment: - Check the status of the deployment using get deployments myapp-deploymen. - Ensure that three pods are running and that the liveness probe is monitoring their health. You can use 'kubectl describe pod myapp-deployment-XXXX' (where XXXX is the pod name) to see the details of the pod and the liveness probe status.

NEW QUESTION # 80
Score: 7%



Task

First, create a snapshot of the existing etcd instance running at https://127.0.0.1:2379, saving the snapshot to /srv/data/etcd-snapshot.db.



Next, restore an existing, previous snapshot located at /var/lib/backup/etcd-snapshot-previo us.db

Answer:

Explanation:
Solution:
#backup
ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt --key=/opt/KUIN000601/etcd-client.key snapshot save /etc/data/etcd-snapshot.db
#restore
ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt --key=/opt/KUIN000601/etcd-client.key snapshot restore /var/lib/backup/etcd-snapshot-previoys.db

NEW QUESTION # 81
You need to set up a load balancer for your Nginx service with the following requirements:
- Session affinity: Preserve client sessions across multiple pods, even if the pod is restarted or rescheduled.
- Health checks: Regularly check the health of Nginx pods and automatically remove unhealthy pods from the load balancer pool.
- Custom header: Add a custom header with the name "X-App-Version" and value "vl .0" to all requests to your Nginx service.
How would you configure your Kubernetes resources to meet these requirements?

Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Define the Service:
- Create a Service of type "LoadBalancer" for your Nginx service.
- Include the sessionAffinity' field with a value of 'ClientIP' to enable client IP-based session affinity.
- Example:

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

2. Configure the Deployment: - In your Nginx Deployment, define a liveness probe and readiness probe to check the health of your Nginx containers. - Example:

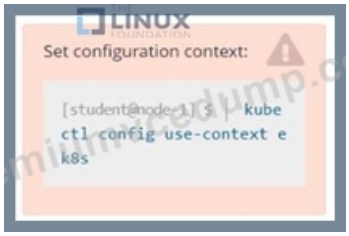
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        livenessProbe:
          tcpSocket:
            port: 80
          initialDelaySeconds: 15
          periodSeconds: 20
          failureThreshold: 3
        readinessProbe:
          tcpSocket:
            port: 80
          initialDelaySeconds: 5
          periodSeconds: 10
          failureThreshold: 2
```

3. Implement the Custom Header: - Configure an Ingress resource with the nginx.ingress.kubernetes.io/add-request-headeo annotation. - Example:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  annotations:
    nginx.ingress.kubernetes.io/add-request-header: "X-App-Version: v1.0"
spec:
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: nginx-service
            port:
              number: 80
```

4. Apply the Configurations: - Apply the updated Service, Deployment, and Ingress resources using 'kubectl apply -f service.yaml -f deployment.yaml -f ingress.yamr. 5. Verify the Load Balancer: - Access the Nginx service using the external IP address provided by the LoadBalancer. - Verify session affinity by making multiple requests and observing that they are consistently routed to the same pod. - Check the "X-App-Version" header in the responses to confirm that it is set to "vl .0".

NEW QUESTION # 82
Score: 4%



Task
Set the node named ek8s-node-1 as unavailable and reschedule all the pods running on it.

Answer:

Explanation:
See the solution below.
Explanation
SOLUTION:
[student@node-1] > ssh ek8s
kubectl cordon ek8s-node-1
kubectl drain ek8s-node-1 --delete-local-data --ignore-daemonsets --force

NEW QUESTION # 83
Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadWriteMany. The type of volume is hostPath and its location is /srv/app-data.

Answer:

Explanation:
Persistent Volume
A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a cluster-level resource like nodes, which don't belong to any namespace. It is provisioned by the administrator and has a particular file size. This way, a developer deploying their app on Kubernetes need not know the underlying infrastructure. When the developer needs a certain amount of persistent storage for their application, the system administrator configures the cluster so that they consume the PersistentVolume provisioned

in an easy way.
Creating Persistent Volume
kind: PersistentVolume
apiVersion: v1
metadata:
name:app-data
spec:
capacity: # defines the capacity of PV we are creating
storage: 2Gi #the amount of storage we are tying to claim
accessModes: # defines the rights of the volume we are creating
- ReadWriteMany
hostPath:
path: "/srv/app-data" # path to which we are creating the volume
Challenge
* Create a Persistent Volume named app-data, with access mode ReadWriteMany, storage classname shared, 2Gi of storage capacity and the host path /srv/app-data.

2. Save the file and create the persistent volume.
Image for post

3. View the persistent volume.

* Our persistent volume status is available meaning it is available and it has not been mounted yet. This status will change when we mount the persistentVolume to a persistentVolumeClaim.
PersistentVolumeClaim
In a real ecosystem, a system admin will create the PersistentVolume then a developer will create a PersistentVolumeClaim which will be referenced in a pod. A PersistentVolumeClaim is created by specifying the minimum size and the access mode they require from the persistentVolume.
Challenge
* Create a Persistent Volume Claim that requests the Persistent Volume we had created above. The claim should request 2Gi. Ensure that the Persistent Volume Claim has the same storageClassName as the persistentVolume you had previously created.
kind: PersistentVolume
apiVersion: v1
metadata:
name:app-data
spec:
accessModes:
- ReadWriteMany

resources:
requests:
storage: 2Gi
storageClassName: shared
2. Save and create the pvc
njerry191@cloudshell:~ (extreme-clone-2654111)$ kubect1 create -f app-data.yaml persistentvolumeclaim/app-data created
3. View the pvc
Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pvc
NAME    STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS
pv      Bound     pv        512m        RWX             shared
```

4. Let's see what has changed in the pv we had initially created.
Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
NAME    CAPACITY    ACCESS MODES    RECLAIM POLICY    STATUS    CLAIM         STORAGECLASS    REASON    AGE
pv      512m        RWX             Retain            Bound     default/pv    shared          16m
```

Our status has now changed from available to bound.
5. Create a new pod named myapp with image nginx that will be used to Mount the Persistent Volume Claim with the path /var/app/config.
Mounting a Claim
apiVersion: v1
kind: Pod
metadata:
creationTimestamp: null
name: app-data
spec:
volumes:
- name:congigpvc
persistenVolumeClaim:
claimName: app-data
containers:
- image: nginx
name: app
volumeMounts:
- mountPath: "/srv/app-data "
name: configpvc

NEW QUESTION # 84
.....

□ □CKA Exam Simulations

- CKA Test Online □ Valid CKA Test Camp □ Composite Test CKA Price □ Search on □ www.pdfvce.com □ for { CKA } to obtain exam materials for free download □CKA Reliable Mock Test
- Quiz The Best Linux Foundation - CKA - Certified Kubernetes Administrator (CKA) Program Exam Labs □ Search for { CKA } and download exam materials for free through □ www.prepawayete.com □ □Reliable CKA Braindumps Free
- Latest CKA VCE Torrent - CKA Pass4sure PDF - CKA Latest VCE □ Easily obtain free download of▶ CKA ◀ by searching on 【 www.pdfvce.com 】 ↪CKA Exam Reference
- Latest CKA Test Testking ➡□ Download CKA Fee □ CKA Preparation □ Search for 「 CKA 」 and download exam materials for free through▶ www.torrentvce.com ◀ □CKA Preparation
- 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, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, 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, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.yiwang.shop, 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, 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, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

P.S. Free 2025 Linux Foundation CKA dumps are available on Google Drive shared by PremiumVCEDump: https://drive.google.com/open?id=1IYWK_p1vcv52hiuKodnYbpVBk4nFRFR5