

Linux Foundation CKA Latest Study Plan, Online CKA Version

Questions & Answers PDF

P-1

Linux Foundation CKA Exam Certified Kubernetes Administrator (CKA) Program



www.certkillers.net

DOWNLOAD the newest Exam4Labs CKA PDF dumps from Cloud Storage for free: <https://drive.google.com/open?id=1BkJ8pAk8B2oHR47zweEUkmHCmS5tXpkU>

The Linux Foundation CKA practice exam will be a great help because you are left with little time to prepare for the Linux Foundation CKA certification exam which you cannot waste to make time for the Linux Foundation CKA Exam Questions. Get the Linux Foundation CKA certification by preparing through Linux Foundation CKA exam questions that will help you pass the Linux Foundation CKA exam.

Linux Foundation Certified Kubernetes Administrator (CKA) Program is an industry-leading certification designed for IT professionals who want to prove their expertise in Kubernetes administration. Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. The CKA program is designed to assess an individual's skills in Kubernetes administration and to provide a certification that is recognized by the industry.

>> Linux Foundation CKA Latest Study Plan <<

Online CKA Version & CKA New Soft Simulations

Just imagine that if you get the CKA certification, then getting high salary and promotion will completely have no problem. At the same time, you will have more income to lead a better life and develop your life quality. Who will refuse such a wonderful dream? So you must struggle for a better future. Life is a long journey. It is never too late to learn new things. Our CKA Study Materials will never disappoint you. And you will get all you desire with our CKA exam questions.

The CKA Program Certification Exam is a valuable certification for professionals who want to work with Kubernetes. It is recognized by companies around the world as a standard for Kubernetes expertise. Certified Kubernetes Administrator (CKA) Program Exam certification can help professionals advance their careers and increase their earning potential. It is also a great way to demonstrate a commitment to professional development and open-source technologies.

Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q69-Q74):

NEW QUESTION # 69

List all the pods that are serviced by the service "webservice" and copy the output in /opt/\$USER/webservice.targets Note: You need to list the endpoints

Answer:

Explanation:

```
kubectl describe svc webservice | grep -i "Endpoints"> /opt/$USER/webservice.targets kubectl get endpoints webservice > /opt/$USER/webservice.targets
```

NEW QUESTION # 70

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

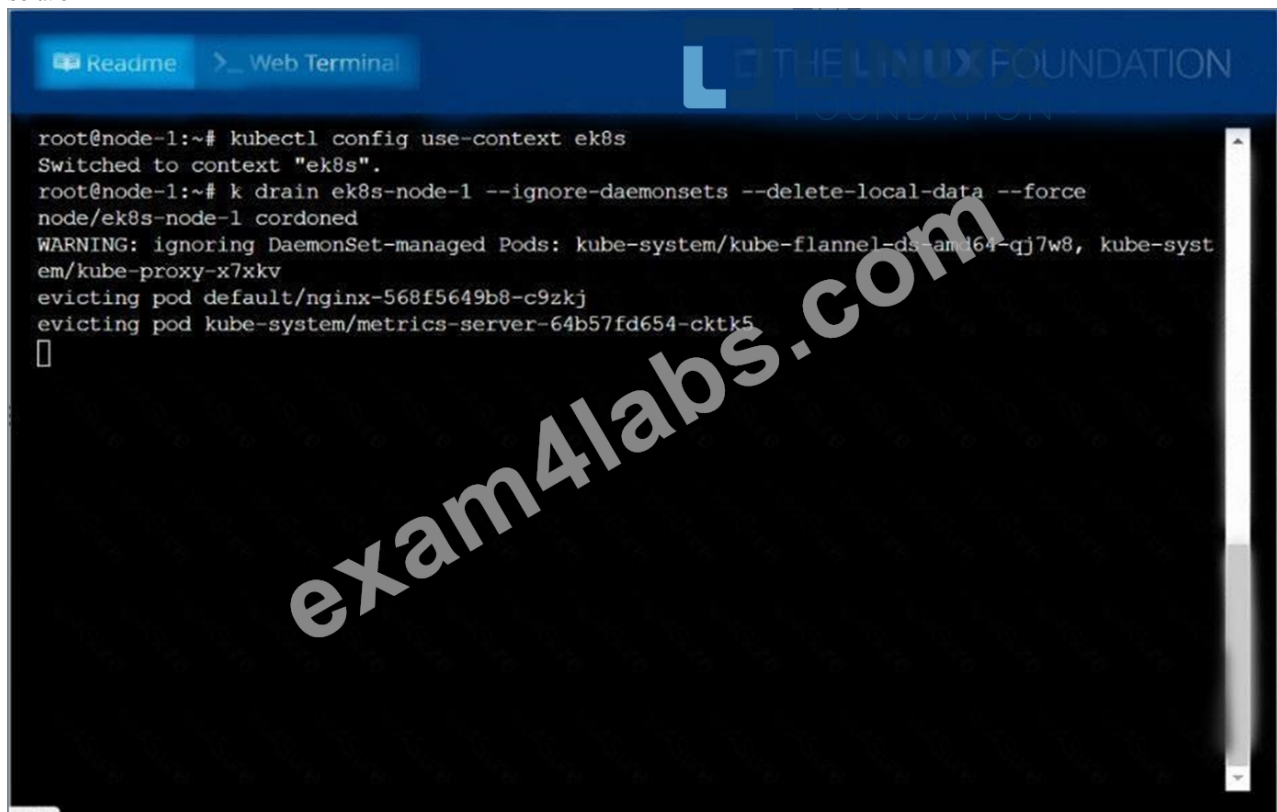
Answer:

Explanation:

See the solution below.

Explanation

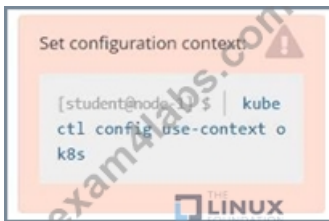
solution



```
root@node-1:~# kubectl config use-context ek8s
Switched to context "ek8s".
root@node-1:~# k drain ek8s-node-1 --ignore-daemonsets --delete-local-data --force
node/ek8s-node-1 cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-amd64-qj7w8, kube-system/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zkj
evicting pod kube-system/metrics-server-64b57fd654-cktk5
[]
```

NEW QUESTION # 71

Score:7%



Task

Create a new PersistentVolumeClaim

- * Name: pv-volume
- * Class: csi-hostpath-sc
- * Capacity: 10Mi

Create a new Pod which mounts the PersistentVolumeClaim as a volume:

- * Name: web-server
- * Image: nginx
- * Mount path: /usr/share/nginx/html

Configure the new Pod to have ReadWriteOnce access on the volume.

Finally, using kubectl edit or kubectl patch expand the PersistentVolumeClaim to a capacity of 70Mi and record that change.

Answer:

Explanation:

Solution:

vi pvc.yaml

storageclass pvc

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pv-volume

spec:

accessModes:

- ReadWriteOnce

volumeMode: Filesystem

resources:

requests:

storage: 10Mi

storageClassName: csi-hostpath-sc

vi pod-pvc.yaml

apiVersion: v1

kind: Pod

metadata:

name: web-server

spec:

containers:

- name: web-server

image: nginx

volumeMounts:

- mountPath: "/usr/share/nginx/html"

name: my-volume

volumes:

- name: my-volume

persistentVolumeClaim:

claimName: pv-volume

craete

kubectl create -f pod-pvc.yaml

#edit

kubectl edit pvc pv-volume --record

NEW QUESTION # 72

Print pod name and start time to "/opt/pod-status" file

Answer:

Explanation:

See the solution below.

Explanation

```
kubectl get pods -o=jsonpath='{range items[*]} {metadata.name} {"\t"} {status.podIP} {"\n"} {end}'
```

NEW QUESTION # 73

You have a Deployment named 'database-deployment' with 2 replicas of a database container. You want to implement a rolling update strategy that allows for a maximum of one pod to be unavailable at any given time. However, you also want to ensure that the database container is restarted automatically when it encounters a failure or crashes. This restart policy should be applied only to the database containers within the deployment and not affect any other containers in the same pod.

Answer:

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1. Update the Deployment YAML:

- Update the 'replicas' to 2.
- Define 'maxUnavailable: 1' and 'maxSurge: 0' in the 'strategy.rollingUpdate' section to control the rolling update process.
- Configure a 'strategy.type' to 'RollingUpdate' to trigger a rolling update when the deployment is updated.
- Set 'spec.template.spec.containers[0].restartPolicy: Always' to ensure the database container restarts automatically when it encounters a failure.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: database-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: database
  template:
    metadata:
      labels:
        app: database
    spec:
      containers:
        - name: database
          image: my-database-image:latest
          restartPolicy: Always
        - name: other-container
          image: my-other-container-image:latest
      strategy:
        type: RollingUpdate
        rollingUpdate:
          maxUnavailable: 1
          maxSurge: 0
```

2. Create the Deployment: - Apply the updated YAML file using 'kubectl apply -f database-deployment.yaml' 3. Verify the Deployment: - Check the status of the deployment using 'kubectl get deployments database-deployment' to confirm the rollout and updated replica count. 4. Test the Restart Policy: - Trigger a failure in the 'database' container within one of the pods. This can be done by sending a SIGKILL signal to the container or by simulating a crash within the container itself. - Observe that the 'database' container will be restarted automatically while other containers in the same pod will remain unaffected. 5. Trigger the Automatic Update: - Push a new image to the 'my-database-image:latest' registry. 6. Monitor the Deployment: - Use 'kubectl get pods -l app=database' to monitor the pod updates during the rolling update process. 7. Check for Successful Update: - Once the deployment is complete, use 'kubectl describe deployment database-deployment' to see that the 'updatedReplicas' field matches the 'replicas' field, indicating a successful update.

• • • • •

- 2026 Latest Exam4Labs CKA PDF Dumps and CKA Exam Engine Free Share: <https://drive.google.com/open?id=1BkJ8pAk8B2oHR47zweEUkmlHCmS5tXpkU>

2026 Latest Exam4Labs CKA PDF Dumps and CKA Exam Engine Free Share: <https://drive.google.com/open?id=1BkJ8pAk8B2oHR47zweEUkmlHCmS5tXpkU>