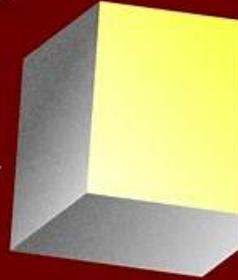


# CKA DUMPS 2022

Certified Kubernetes Administrator



## 1

### Exam Breakdown

The CKA exam covers 5 main domains: Cluster Architecture (25%), Workloads & Scheduling (15%), Services & Networking (20%), Storage (10%), and Troubleshooting (30%)

## 2

### Prep Timeline

Recommended CKA exam prep timeline is 6-8 weeks of preparation

## 3

### Key Strategies

Key CKA exam prep strategies include taking practice exams, reviewing Kubernetes concepts, and reading exam tips from professionals

## 4

### Career Impact

Key CKA exam prep strategies include taking practice exams, reviewing Kubernetes concepts, and reading exam tips from professionals

## 5

### Exam Logistics

The CKA exam fee is \$375 and the certification is valid for 3 years



P.S. Free & New CKA dumps are available on Google Drive shared by VCEPrep: <https://drive.google.com/open?id=1ARBB0TX8hCSH7TK8Mn56yFuRL059pHgy>

When new changes or knowledge are updated, our experts add additive content into our CKA latest material. They have always been in a trend of advancement. Admittedly, our CKA real questions are your best choice. We also estimate the following trend of exam questions may appear in the next exam according to syllabus. So they are the newest and also the most trustworthy CKA Exam Prep to obtain.

The CKA exam is designed to test the knowledge and proficiency of professionals in Kubernetes administration. CKA exam consists of a set of performance-based tasks that assess the candidate's ability to deploy, configure, manage, and troubleshoot Kubernetes clusters. CKA exam is conducted online and requires candidates to complete a set of practical tasks within a specified time frame. CKA exam covers a wide range of Kubernetes administration topics, including cluster architecture, networking, storage, security, and troubleshooting.

The CKA Program Certification Exam is a challenging and respected certification that validates the skills and knowledge of professionals who want to work with Kubernetes. Certified Kubernetes Administrator (CKA) Program Exam certification covers a wide range of topics related to Kubernetes and requires a deep understanding of the technology. It is a valuable certification for professionals who want to advance their careers and demonstrate their commitment to open-source technologies.

The CKA Certification Exam is designed to be challenging, and passing it requires a significant amount of preparation and practice. The Linux Foundation offers a variety of training courses and study materials to help candidates prepare for the exam. The CKA certification is highly valued by employers and IT professionals alike, as it demonstrates a high level of proficiency in managing Kubernetes clusters and ensures that the certified professional is up-to-date with the latest best practices in the industry.

>> Valid Dumps CKA Ebook <<

## **CKA Practice Questions & CKA Actual Lab Questions: Certified Kubernetes Administrator (CKA) Program Exam**

Preparing with outdated CKA exam questions results in failure and loss of time and money. You can get success in the exam on first attempt and save your resources with the help of updated exam questions. We offer Linux Foundation CKA real questions to help pupils in getting ready for the exam in a short time. Students who choose VCEPrep will get the latest and updated exam questions they need to prepare for the CKA examination in a short time.

### **Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q72-Q77):**

#### **NEW QUESTION # 72**

Make the node schedulable by uncondon the node

#### **Answer:**

Explanation:

```
kubectl uncondon node-1 //verify kubectl get no
```

#### **NEW QUESTION # 73**

You have an application deployed in your Kubernetes cluster that is configured to access a private database hosted on another Kubernetes cluster in a different VPC. How would you configure a secure and efficient way for your application to communicate with the private database?

#### **Answer:**

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1 . Create a Service Account in the Database Cluster:

- Create a Service Account in the database cluster that will be used by the application in the other cluster.

- Example:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: database-client
```

- Apply this configuration using 'kubectl apply -f service-account.yaml' in the database cluster. 2. Generate a Secret for the Service Account: - Generate a secret for the Service Account. This secret will contain the credentials needed to access the database. -

Example: `bash kubectl create secret generic database-client-secret --from-literal=username=username -n namespace` - Replace `username` with your actual credentials and `namespace` with the namespace where your database service is running. 3. Configure a Network Policy: - Create a Network Policy in the database cluster to allow traffic from your application's pods in the other cluster. - Example:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-app-access
spec:
  podSelector: {}
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app:
```

- Replace with the label assigned to your application pods. 4. Create a Service in the Database Cluster: - Create a Service in the database cluster that exposes your database. This service will be used by your application to connect to the database. - Example:

```
apiVersion: v1
kind: Service
metadata:
  name: database-service
spec:
  ports:
  - port: 5432
    targetPort: 5432
    protocol: TCP
  selector:
    app: database
```

- Replace 'database' with the label of your database pods. 6. Configure the Application Deployment: - In your application's deployment configuration, mount the secret containing the database credentials and configure your application to connect to the database service in the other cluster using the service's DNS name. - Example:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 2
  template:
    spec:
      containers:
      - name: my-app
        image: my-app:latest
        env:
        - name: DATABASE_URL
          value: postgres://database-client:database-client-secret@database-service.database-namespace.svc.cluster.local:5432/your-database
        volumeMounts:
        - name: database-credentials
          mountPath: /var/secrets/database
      volumes:
      - name: database-credentials
        secret:
          secretName: database-client-secret
```

- Ensure that the application has the necessary network connectivity to access the database cluster through your network configuration. - Example: - Configure a VPC peering connection between the two VPCs to enable direct communication between

the clusters. - Create a VPN connection between the clusters using a VPN gateway. 7. Test the Connection: - Verify that your application can successfully connect to and interact with the database.

#### NEW QUESTION # 74

List all persistent volumes sorted by capacity, saving the full kubectl output to /opt/KUCC00102/volume\_list. Use kubectl's own functionality for sorting the output, and do not manipulate it any further.

**Answer:**

Explanation:

See the solution below.

Explanation

solution



```
77d
pv0007 7Gi RWO Recycle Available slow
77d
pv0006 8Gi RWO Recycle Available slow
77d
pv0003 10Gi RWO Recycle Available slow
77d
pv0002 11Gi RWO Recycle Available slow
77d
pv0010 13Gi RWO Recycle Available slow
77d
pv0011 14Gi RWO Recycle Available slow
77d
pv0001 16Gi RWO Recycle Available slow
77d
pv0009 17Gi RWO Recycle Available slow
77d
pv0005 18Gi RWO Recycle Available slow
77d
pv0008 19Gi RWO Recycle Available slow
77d
pv0000 21Gi RWO Recycle Available slow
77d
root@node-1:~# k get pv --sort-by=.spec.capacity.storage > /opt/KUCC00102/volume_list
root@node-1:~#
```

#### NEW QUESTION # 75

Undo/Rollback deployment to specific revision "1"

- A. // Check Deployment History  
kubectl rollout history deployment webapp  
//Rollback to particular revision  
kubectl rollout undo deployment webapp --to-revision=1
- B. // Check Deployment History  
kubectl rollout history deployment webapp  
kubectl rollout undo deployment webapp --to-revision=1

**Answer: A**

#### NEW QUESTION # 76

You have a Deployment named 'nginx-deployment' running an Nginx server. The Nginx configuration file is stored in a ConfigMap named 'nginx-config'. You need to dynamically update the Nginx configuration file without restarting the Nginx pods.

**Answer:**

Explanation:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1 . Create the ConfigMap (if not already existing):

- Define a ConfigMap named 'nginx-config' containing the Nginx configuration file. For example, create a file 'nginx.conf' with the desired configuration and then create the ConfigMap using 'kubectl create configmap nginx-config --from-file=nginx.conf'

kubectl create configmap nginx-config --from-file=nginx.conf

2. Configure Nginx Deployment:

- Modify the 'nginx-deployment' Deployment to mount the 'nginx-config' ConfigMap as a volume.

- Use 'volumeMounts' to specify where the ConfigMap should be mounted (e.g., '/etc/nginx/conf.d/') and 'volumes' to define the ConfigMap as a volume source.

- Update the Nginx container's configuration to use the mounted configuration file (e.g., 'nginx -g daemon off;').

```
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          volumeMounts:
            - name: nginx-config-volume
              mountPath: /etc/nginx/conf.d
          volumes:
            - name: nginx-config-volume
              configMap:
                name: nginx-config
```

3. Update the ConfigMap: - Modify the 'nginx-config' ConfigMap with the new configuration content. This can be done using 'kubectl patch' or 'kubectl edit': kubectl patch configmap nginx-config -p '{"data": {"nginx.conf": "new\_nginx\_configuration"}}' 4.

Observe Nginx Pods: - Monitor the Nginx pods in the 'nginx-deployment' Deployment. Since the ConfigMap is mounted as a volume, Nginx will automatically reload the configuration file without restarting the pod.

5. Verify the Update: - Use 'kubectl logs' or 'kubectl exec' to examine the Nginx pod's logs and confirm that the new configuration is being used.

## NEW QUESTION # 77

.....

Our company has spent more than 10 years on compiling CKA study materials for the exam in this field, and now we are delighted to be here to share our CKA learnign guide with all of the candidates for the exam in this field. There are so many striking points of our CKA Preparation exam. If you want to have a better understanding of our CKA exam braindumps, just come and have a try!

**Valid CKA Test Sample:** <https://www.vceprep.com/CKA-latest-vce-prep.html>

- New CKA Test Braindumps  CKA Valid Test Vce Free  Latest CKA Braindumps Free  Open [ [www.troytecdumps.com](http://www.troytecdumps.com) ] enter { CKA } and obtain a free download  Latest CKA Braindumps Free
- Dumps CKA Questions  CKA Authorized Exam Dumps  CKA Free Sample  Search for ⇒ CKA ⇐ and obtain a free download on ➡ [www.pdfvce.com](http://www.pdfvce.com)   CKA Exam Flashcards

