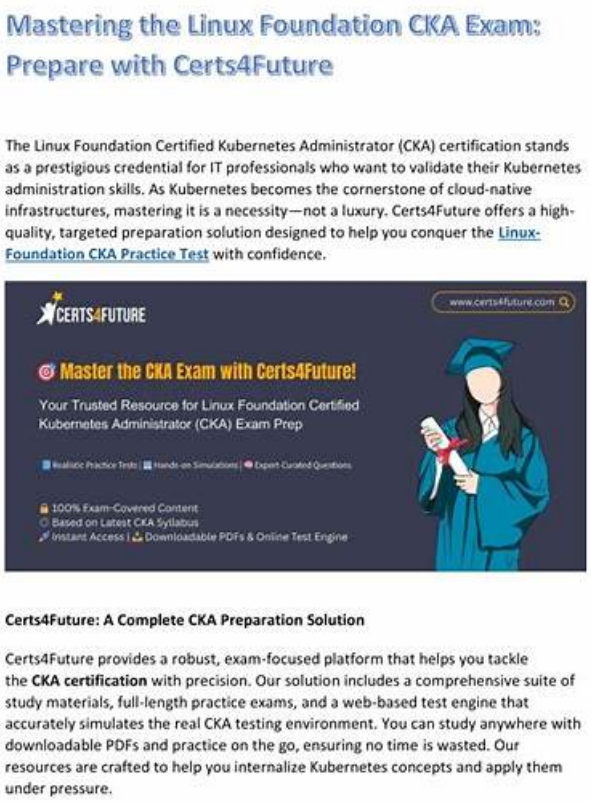
Quiz 2026 Linux Foundation Efficient CKA New Study Notes



P.S. Free 2025 Linux Foundation CKA dumps are available on Google Drive shared by Exam-Killer: https://drive.google.com/open?id=18qznQT3SXWaeAGRUQ6iNqE6Xo3lvRo6W

The customizable mock tests make an image of a real-based Certified Kubernetes Administrator (CKA) Program Exam (CKA) exam which is helpful for you to overcome the pressure of taking the final examination. Customers of Exam-Killer can take multiple Certified Kubernetes Administrator (CKA) Program Exam (CKA) practice tests and improve their preparation to achieve the CKA Certification. You can even access your previously given tests from the history, which allows you to be careful while giving the mock test next time and prepare for Certified Kubernetes Administrator (CKA) Program Exam (CKA) certification in a better way.

The CKA Program Certification Exam is designed to test the skills of professionals who have experience working with Kubernetes. CKA exam is an online, proctored exam that can be taken from anywhere in the world. CKA exam consists of 24 performance-based tasks that require candidates to demonstrate their ability to perform common Kubernetes operations. CKA exam is designed to be challenging and requires a high level of skill and knowledge.

Linux Foundation CKA Program Certification Exam is a valuable certification for professionals who work with Kubernetes and want to validate their skills and knowledge. CKA exam tests candidates on various aspects of Kubernetes administration, and the certification is recognized by leading companies in the industry. With the increasing demand for certified Kubernetes administrators, the CKA Certification provides a competitive edge to professionals and opens up new career opportunities in the field of containerization and cloud computing.

Linux Foundation CKA Program is a certification exam that validates one's skills and knowledge in Kubernetes administration. CKA exam is designed for IT professionals who have experience in administering Kubernetes clusters and covers a range of topics from basic to advanced. Certified Kubernetes Administrator (CKA) Program Exam certification is widely recognized in the industry and is a valuable asset for IT professionals. The CKA certification can lead to better job opportunities and higher salaries, and it is a

stepping stone to more advanced Kubernetes certifications.

>> CKA New Study Notes <<

Real CKA Question, CKA Practice Engine

Our CKA practice questions are on the cutting edge of this line with all the newest contents for your reference. Free demos are understandable and part of the CKA exam materials as well as the newest information for your practice. And because that our CKA Study Guide has three versions: the PDF, Software and APP online. So accordingly, we offer three versions of free demos for you to download.

Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam Sample Questions (Q80-Q85):

NEW QUESTION # 80
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 'ClientlP' to enable client IP-based session affinity.
- Example:

```
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:

```yaml
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 # 81
You have a Deployment with 5 replicas. You want to increase the number of replicas to 10, but only after ensuring that the new pods are healthy and ready to serve traffic.

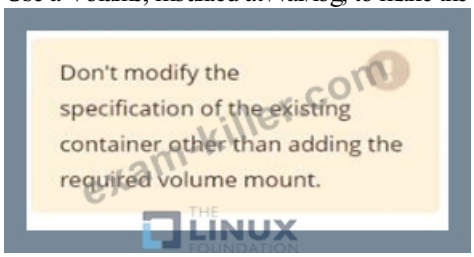
Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Update the Deployment YAML:
- Update the 'replicas' field in the Deployment YAML to 10.

2. Apply the Changes: - Apply the updated YAML file using 'kubectl apply -f my-deployment.yaml' 3. Monitor Pod Status: - Use 'kubectl get pods -l app=my-app' to monitor the status of the pods. - Ensure that the new pods are in the "Running' state and have a 'Ready' status. 4. Check Liveness and Readiness Probes: - If applicable, ensure that liveness and readiness probes are configured to check the health of the pods. - This helps in identifying and restarting unhealthy pods. 5. Verify Service Availability: - Use 'kubectl get services my-service" to check the service status. - Ensure that the service is still available and serving traffic. 6. Increase Replicas: - Once the new pods are healthy and ready, the deployment will automatically scale up to 10 replicas.

NEW QUESTION # 82
Score:7%

Context
An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e. g. kubectl logs). Adding a streaming sidecar container is a good and common way to accomplish this requirement.
Task
Add a sidecar container named sidecar, using the busybox Image, to the existing Pod big-corp-app. The new sidecar container has to run the following command:
/bin/sh -c tail -n+1 -f /va r/log/big-corp-app.log
Use a Volume, mounted at /var/log, to make the log file big-corp-app.log available to the sidecar container.



Answer:

Explanation:

Solution:
#
kubectl get pod big-corp-app -o yaml
#
apiVersion: v1
kind: Pod
metadata:
name: big-corp-app
spec:
containers:
- name: big-corp-app
image: busybox
args:
- /bin/sh
- -c
- >
i=0;
while true;
do
echo "$(date) INFO $i" >> /var/log/big-corp-app.log;
i=$((i+1));
sleep 1;
done
volumeMounts:
- name: logs
mountPath: /var/log
- name: count-log-1
image: busybox
args: [/bin/sh, -c, 'tail -n+1 -f /var/log/big-corp-app.log']
volumeMounts:
- name: logs
mountPath: /var/log
volumes:
- name: logs
emptyDir: {
}
#
kubectl logs big-corp-app -c count-log-1

NEW QUESTION # 83
You have a Kubernetes cluster with a deployment named 'nginx-deployment' in the 'default' namespace. This deployment uses a container image 'nginx:latest'.
You want to define an admission webhook that enforces a policy to prevent deployments from using 'nginx:latest' and instead forces the use of a specific versioned image like 'nginx:1.20.1'. Create the webhook configuration and admission controller code that implements this policy.

Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1 . Create a Webhook Configuration:

```
apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingWebhookConfiguration
metadata:
  name: nginx-image-policy
webhooks:
- name: nginx-image-validation
  clientConfig:
    service:
      namespace: default
      name: nginx-admission-webhook
      path: /validate
    caBundle:  # Replace with your CA bundle
  rules:
  - apiGroups: ["apps"]
    apiVersions: ["v1"]
    resources: ["deployments"]
    operations: ["CREATE", "UPDATE"]
  failurePolicy: Fail
  sideEffects: None
```

2. Create a Service for the Admission Webhook:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-admission-webhook
  namespace: default
spec:
  ports:
  - port: 443
    targetPort: 8443
  selector:
    app: nginx-admission-webhook
```

3. Create the Admission Controller Code: Example in Go:

```
package main

import (
	"context"
	"encoding/json"
	"fmt"
	"io/ioutil"
	"net/http"

	admissionv1 "k8s.io/api/admission/v1"
	metav1 "k8s.io/apimachinery/pkg/apis/meta/v1"
	"k8s.io/apimachinery/pkg/runtime"
	"k8s.io/apimachinery/pkg/runtime/serializer"
	"k8s.io/apimachinery/pkg/types"
	"k8s.io/client-go/kubernetes"
	"k8s.io/client-go/rest"
	"k8s.io/client-go/tools/clientcmd"
)

var (
	runtimeScheme = runtime.NewScheme()
	codecs        = serializer.NewCodecFactory(runtimeScheme)
)

func main() {
	// Load Kubernetes configuration
	config, err := rest.InClusterConfig()
	if err != nil {
		config, err = clientcmd.BuildConfigFromFlags("", clientcmd.RecommendedHomeFile)
		if err != nil {
			panic(err)
		}
	}
	// Create Kubernetes client
	clientset, err := kubernetes.NewForConfig(config)
	if err != nil {
		panic(err)
	}
	// Start HTTP server
	http.HandleFunc("/validate", func(w http.ResponseWriter, r http.Request) {
		handleAdmissionRequest(w, r, clientset)
	})
	http.ListenAndServe(":8443", nil)
}

func handleAdmissionRequest(w http.ResponseWriter, r http.Request, clientset kubernetes.Clientset) {
	body, err := ioutil.ReadAll(r.Body)
	if err != nil {
		http.Error(w, fmt.Sprintf("failed to read request body: %v", err),http.StatusBadRequest)
		return
	}

	// Decode the AdmissionReview request
	review := &admissionv1.AdmissionReview{}
	_, _, err = codecs.UniversalDeserializer().Decode(body, nil, review)
	if err != nil {
		http.Error(w, fmt.Sprintf("failed to decode AdmissionReview: %v", err),http.StatusBadRequest)
		return
	}

	// Check the image name in the deployment spec
	deployment := &admissionv1.AdmissionReview{}
	if review.Request.Kind.Kind == "Deployment" {
		obj := review.Request.Object.Raw
		_, _, err = codecs.UniversalDeserializer().Decode(obj, nil, deployment)
		if err != nil {
			http.Error(w, fmt.Sprintf("failed to decode Deployment: %v", err),http.StatusBadRequest)
			return
		}
		for _, container := range deployment.Spec.Template.Spec.Containers {
			if container.Image == "nginx:latest" {
				review.Response = &admissionv1.AdmissionResponse{
					UID:     review.Request.UID,
					Allowed: false,
					Result: &metav1.Status{
						Message: "Deployment must use a specific versioned image like 'nginx:1.20.1'.",
					},
				}
				return
			}
		}
	}

	// If no errors, allow the request
	review.Response = &admissionv1.AdmissionResponse{
		UID:     review.Request.UID,
		Allowed: true,
	}
	json.NewEncoder(w).Encode(review)
}
```

4. Build and Deploy the Admission Controller: Build the Go code. Deploy the admission controller as a container in the Kubernetes

cluster. Create the Service for the webhook. Create the WebhookConfiguration with the correct CA bundle for the admission controller. The webhook configuration defines the rules for the admission controller, including the resources, operations, and failure policy. The admission controller code handles the validation of the deployment object. It checks the image name and rejects deployments that use the 'nginx:latest' image. The Service exposes the admission controller to the Kubernetes API. The CA bundle is used to secure communication between the webhook and the Kubernetes API. Note: Replace with the actual CA bundle data. The admission controller code should be deployed and configured according to your specific environment and needs.,

NEW QUESTION # 84
Create a pod that having 3 containers in it? (Multi-Container)

- A. image=nginx, image=redis, image=consul
  Name nginx container as "nginx-container"
  Name redis container as "redis-container"
  Name consul container as "consul-container"
  Create a pod manifest file for a container and append container
  section for rest of the images
  kubectl run multi-container --generator=run-pod/v1 --image=nginx --
  dry-run -o yaml > multi-container.yaml
  # then
  vim multi-container.yaml
  apiVersion: v1
  kind: Pod
  metadata:
  labels:
  run: multi-container
  name: multi-container
  spec:
  containers:
  - image: nginx
  name: nginx-container
  - image: redis
  name: redis-container
  - image: consul
  name: consul-container
  restartPolicy: Always
- B. image=nginx, image=redis, image=consul
  Name nginx container as "nginx-container"
  Name redis container as "redis-container"
  Name consul container as "consul-container"
  Create a pod manifest file for a container and append container
  section for rest of the images
  kubectl run multi-container --generator=run-pod/v1 --image=nginx --
  dry-run -o yaml > multi-container.yaml
  # then
  vim multi-container.yaml
  labels:
  run: multi-container
  name: multi-container
  spec:
  containers:
  - image: nginx
  name: nginx-container
  - image: redis
  name: consul-container
  restartPolicy: Always

Answer: A

NEW QUESTION # 85
.....

These formats are CKA web-based practice test software, desktop practice exam software, and Certified Kubernetes Administrator (CKA) Program Exam (CKA) PDF dumps files. All these three Linux Foundation CKA exam questions formats are easy to use and compatible with all devices and the latest web browsers. Just choose the right Certified Kubernetes Administrator (CKA) Program Exam (CKA) exam dumps format and start CKA exam questions preparation today.

Real CKA Question: https://www.exam-killer.com/CKA-valid-questions.html

- Maximize Your Chances of Getting Linux Foundation CKA Certification Exam □ Search for ➡ CKA □□□ and download it for free immediately on ➠ www.practicevce.com □ □New Exam CKA Braindumps
- New Exam CKA Braindumps □ CKA Latest Exam Camp □ New Exam CKA Braindumps □ Go to website " www.pdfvce.com " open and search for □ CKA □ to download for free ✳ Minimum CKA Pass Score
- CKA New Study Notes - Leader in qualification Exams - Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam 🖼 Search for ➤ CKA □ and download it for free immediately on □ www.vce4dumps.com □ □CKA Valid Practice Materials
- Maximize Your Chances of Getting Linux Foundation CKA Certification Exam □ Open （ www.pdfvce.com ） and search for [CKA] to download exam materials for free □Reliable CKA Test Price
- Valid CKA Exam Duration □ New Exam CKA Braindumps □ Detailed CKA Answers □ Search for ✔ CKA □✔□ on 「 www.dumpsmaterials.com 」 immediately to obtain a free download □New CKA Exam Sample
- 2026 Efficient CKA New Study Notes | CKA 100% Free Real Question □ Easily obtain free download of ☀ CKA □☀□ by searching on ▹ www.pdfvce.com ◃ □Minimum CKA Pass Score
- Exam Dumps CKA Demo □ Free CKA Brain Dumps □ Detailed CKA Answers □ Copy URL ➽ www.testkingpass.com □ open and search for ➠ CKA □ to download for free □Valid Braindumps CKA Files
- CKA New Study Notes - Leader in qualification Exams - Linux Foundation Certified Kubernetes Administrator (CKA) Program Exam □ Simply search for ➤ CKA □ for free download on 《 www.pdfvce.com 》 □CKA Exam Sample Online
- CKA Exam Sample Online □ Valid Braindumps CKA Files □ CKA Valid Practice Materials □ Search for 《 CKA 》 and download it for free on □ www.vce4dumps.com □ website □New CKA Exam Sample
- 2026 Efficient CKA New Study Notes | CKA 100% Free Real Question □ Immediately open " www.pdfvce.com " and search for " CKA " to obtain a free download □CKA Valid Practice Materials
- Pass Guaranteed Quiz 2026 High Hit-Rate Linux Foundation CKA: Certified Kubernetes Administrator (CKA) Program Exam New Study Notes □ Go to website 《 www.practicevce.com 》 open and search for 【 CKA 】 to download for free □Latest CKA Test Objectives
- digitalkhichdi.com, 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, 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, formationenlignemaroc.com, 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.fanart-central.net, disqus.com, Disposable vapes

P.S. Free 2025 Linux Foundation CKA dumps are available on Google Drive shared by Exam-Killer: https://drive.google.com/open?id=18qznQT3SXWaeAGRUQ6iNqE6Xo3lvRo6W