Kostenlose Kubernetes and Cloud Native Associate vce dumps & neueste KCNA examcollection Dumps

Außerdem sind jetzt einige Teile dieser ZertSoft KCNA Prüfungsfragen kostenlos erhältlich: https://drive.google.com/open?id=1nSmF6jEK814HMWyj-7dyZtbsKTRX96a5

Gegenüber der Linux Foundation KCNA Prüfung ist jeder Kandidat verwirrt. Jeder hat seine eigene Idee. Aber für alle ist diese Prüfung schwer. Die Linux Foundation KCNA Prüfung ist eine schwierige Zertifizierung. Ich glaube, alle wissen es. Mit ZertSoft ist alles einfacher geworden. Die Dumps zur Linux Foundation KCNA Prüfung von ZertSoft sind der Grundbedarfsgüter jedes Kandidaten. Sie können sicher die Linux Foundation KCNA Zertifizierungsprüfung bestehen. Wenn Sie nicht glauben, gucken Sie mal unsere Website. Sein Kauf-Rate ist die höchste. Sie sollen ZertSoft nicht verpassen, fügen Sie ZertSoft schnell in den Warenkorb hinzu.

Die Linux Foundation KCNA Zertifizierungsprüfung ist heutztage in der konkurrenzfähigen IT-Branche immer beliebter geworden. Immer mehr Leute haben die Linux Foundation KCNA Prüfung abgelegt. Aber ihre Schwierigkeit nimmt doch nicht ab. Es ist schwer, die Linux Foundation KCNA Prüfung zu bestehen, weil sie sowieso eine autoritäre Prüfung ist, die Computerfachkenntnisse und die Fähigkeiten zur Informationstechnik prüft. Viele Leute haben viel Zeit und Energie auf die Linux Foundation KCNA Zertifizierungsprüfung aufgewendet.

>> KCNA Trainingsunterlagen <<

KCNA Demotesten & KCNA Exam Fragen

Die Linux Foundation KCNA Zertifizierungsprüfung wird jetzt immer populärer. Es gibt viele verschiedene IT-Zertifizierungsprüfungen. Welche Prüfung haben Sie abgelegt? Lassen Wir hier Linux Foundation KCNA Zertifizierungsprüfung als Beispiel erklären. Wenn Sie an der KCNA Prüfung teilnehmen, Linux Foundation KCNA Dumps von ZertSoft Ihnen helfen, sehr leicht die Prüfung zu bestehen.

Linux Foundation Kubernetes and Cloud Native Associate KCNA Prüfungsfragen mit Lösungen (Q151-Q156):

151. Frage
How long should a stable API element in Kubernetes be supported (at minimum) after deprecation?

- A. 6 months
- B. 24 months
- C. 12 months
- D. 9 months

Antwort: C

Begründung:
Kubernetes has a formal API deprecation policy to balance stability for users with the ability to evolve the platform. For a stable

(GA) API element, Kubernetes commits to supporting that API for a minimum period after it is deprecated. The correct minimum in this question is 12 months, which corresponds to option C.
In practice, Kubernetes releases occur roughly every three to four months, and the deprecation policy is commonly communicated in terms of "releases" as well as time. A GA API that is deprecated in one release is typically kept available for multiple subsequent releases, giving cluster operators and application teams time to migrate manifests, client libraries, controllers, and automation. This matters because Kubernetes is often at the center of production delivery pipelines; abrupt API removals would break deployments, upgrades, and tooling. By guaranteeing a minimum support window, Kubernetes enables predictable upgrades and safer lifecycle management.
This policy also encourages teams to track API versions and plan migrations. For example, workloads might start on a beta API (which can change), but once an API reaches stable, users can expect a stronger compatibility promise. Deprecation warnings help surface risk early. In many clusters, you'll see API server warnings and tooling hints when manifests use deprecated fields/versions, allowing proactive remediation before the removal release.
Options 6 or 9 months would be too short for many enterprises to coordinate changes across multiple teams and environments. 24 months may be true for some ecosystems, but the Kubernetes stated minimum in this exam-style framing is 12 months. The key operational takeaway is: don't ignore deprecation notices-they're your clock for migration planning. Treat API version upgrades as part of routine cluster lifecycle hygiene to avoid being blocked during Kubernetes version upgrades when deprecated APIs are finally removed.

152. Frage
What command use to get documentation about kubernetes resource type

- A. alias k='kubectl'
  k api-resources
- B. alias k='kubectl'
  k get resource
- C. alias k='kubectl'
  k explain
- D. alias k='kubectl'
  k api-list

Antwort: C

Begründung:
https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#explain

153. Frage
Which of the following is the correct command to run an nginx deployment with 2 replicas?

- A. kubectl create deploy nginx --image=nginx --replicas=2
- B. kubectl create nginx deployment --image=nginx --replicas=2
- C. kubectl run deploy nginx --image=nginx --replicas=2
- D. kubectl create deploy nginx --image=nginx --count=2

Antwort: A

Begründung:
The correct answer is B: kubectl create deploy nginx --image=nginx --replicas=2. This uses kubectl create deployment (shorthand create deploy) to generate a Deployment resource named nginx with the specified container image. The --replicas=2 flag sets the desired replica count, so Kubernetes will create two Pod replicas (via a ReplicaSet) and keep that number stable.
Option A is incorrect because kubectl run is primarily intended to run a Pod (and in older versions could generate other resources, but it's not the recommended/consistent way to create a Deployment in modern kubectl usage). Option C is invalid syntax: kubectl subcommand order is incorrect; you don't say kubectl create nginx deployment. Option D uses a non-existent --count flag for Deployment replicas.
From a Kubernetes fundamentals perspective, this question tests two ideas: (1) Deployments are the standard controller for running stateless workloads with a desired number of replicas, and (2) kubectl create deployment is a common imperative shortcut for generating that resource. After running the command, you can confirm with kubectl get deploy nginx, kubectl get rs, and kubectl get pods -l app=nginx (label may vary depending on kubectl version). You'll see a ReplicaSet created and two Pods brought up.
In production, teams typically use declarative manifests (kubectl apply -f) or GitOps, but knowing the imperative command is useful for quick labs and validation. The key is that replicas are managed by the controller, not by manually starting containers-Kubernetes

reconciles the state continuously.
Therefore, B is the verified correct command.

154. Frage
Which type of Service requires manual creation of Endpoints?

- A. Services without selectors
- B. NodePort
- C. ClusterIP with selectors
- D. LoadBalancer

Antwort: A

Begründung:
A Kubernetes Service without selectors requires you to manage its backend endpoints manually, so B is correct. Normally, a Service uses a selector to match a set of Pods (by labels). Kubernetes then automatically maintains the backend list (historically Endpoints, now commonly EndpointSlice) by tracking which Pods match the selector and are Ready. This automation is one of the key reasons Services provide stable connectivity to dynamic Pods.
When you create a Service without a selector, Kubernetes has no way to know which Pods (or external IPs) should receive traffic. In that pattern, you explicitly create an Endpoints object (or EndpointSlices, depending on your approach and controller support) that maps the Service name to one or more IP:port tuples. This is commonly used to represent external services (e.g., a database running outside the cluster) while still providing a stable Kubernetes Service DNS name for in-cluster clients. Another use case is advanced migration scenarios where endpoints are controlled by custom controllers rather than label selection.
Why the other options are wrong: Service types like ClusterIP, NodePort, and LoadBalancer describe how a Service is exposed, but they do not inherently require manual endpoint management. A ClusterIP Service with selectors (D) is the standard case where endpoints are automatically created and updated. NodePort and LoadBalancer Services also typically use selectors and therefore inherit automatic endpoint management; the difference is in how traffic enters the cluster, not how backends are discovered.
Operationally, when using Services without selectors, you must ensure endpoint IPs remain correct, health is accounted for (often via external tooling), and you update endpoints when backends change. The key concept is: no selector → Kubernetes can't auto-populate endpoints → you must provide them.

155. Frage
What command can you use to get documentation about a resource type from the command line?

- A. kubectl explain
- B. kubectl api-resources
- C. kubectl get
- D. kubeadm get-resource

Antwort: A

Begründung:
https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#explain

156. Frage
.....

Prüfung vorzubereiten und die Prüfung zu bestehen, was Ihnen viel Angenehmlichkeiten bietet.

Aber jetzt bist du da sagte Matthias, jetzt brauchst du dich nicht KCNA mehr zu sehnen, Aufgedunsen im Tod und mit den angenagten, verrotteten Gesichtern sahen sie einer aus wie der andere.

Jedoch sind sie am Ende doch in der Prüfung durchgefallen, Die Technik-Gruppe von uns ZertSoft haben die Prüfungssoftware der Linux Foundation KCNA nach der Mnemotechnik entwickelt.

KCNA Trainingsmaterialien: Kubernetes and Cloud Native Associate & KCNA Lernmittel & Linux Foundation KCNA Quiz

Einerseits dürfen Sie den KCNA Studienführer gleich herunterladen nach Ihrem Bezahlen, dann können Sie auf die Prüfung selbst konzentrieren und Übungen machen ohne Verzögerung.

Sie wird den Kandidaten helfen, sich auf die KCNA Exam Fragen Prüfung vorzubereiten und die Prüfung zu bestehen, was Ihnen viel Angenehmlichkeiten bietet, Sobald Sie die Materialien gekauft haben, brauchen Sie nur 20 bis 30 Stunden, mit KCNA Fragen & Antworten zu studieren.

Laden Sie die neuesten ZertSoft KCNA PDF-Versionen von Prüfungsfragen kostenlos von Google Drive herunter: https://drive.google.com/open?id=1nSmF6jEK814HMWyj-7dyZtbsKTRX96a5