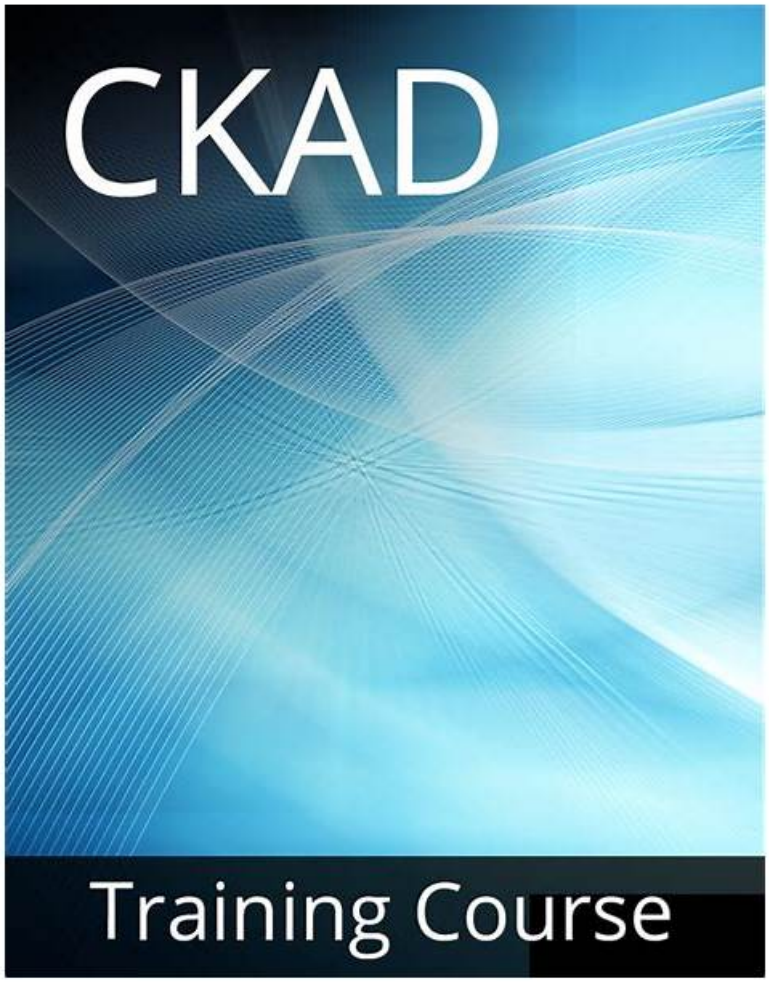
CKAD Online Tests - CKAD Buch



2026 Die neuesten Zertpruefung CKAD PDF-Versionen Prüfungsfragen und CKAD Fragen und Antworten sind kostenlos verfügbar: https://drive.google.com/open?id=1GfmTRL1J8djjAxay0UKC-MuOM7a-zWiR

Warum wollen wir, Sie vor dem Kaufen der Linux Foundation CKAD Prüfungsunterlagen zuerst zu probieren? Warum dürfen wir garantieren, dass Ihr Geld für die Software zurückgeben, falls Sie in der Linux Foundation CKAD Prüfung durchfallen? Der Grund liegt auf unserer Konfidenz für unsere Produkte. Die Linux Foundation CKAD Prüfung wird fortlaufend aktualisiert und wir aktualisieren gleichzeitig unsere Software.

Die Linux Foundation Certified Kubernetes Application Developer (CKAD) Prüfung ist eine Zertifizierungsprüfung, die die Fähigkeit eines Kandidaten testet, cloud-native Anwendungen für Kubernetes zu entwerfen, zu erstellen, zu konfigurieren und freizugeben. Es handelt sich um eine leistungsorientierte Prüfung, die in einer Live-Umgebung durchgeführt wird, und Kandidaten müssen eine Reihe von Herausforderungen mithilfe einer Befehlszeilenschnittstelle lösen. Die Prüfung ist darauf ausgelegt, das Wissen des Kandidaten über Kubernetes und dessen Ökosystem sowie seine Fähigkeit zu testen, dessen Funktionen zu nutzen, um cloud-native Anwendungen zu erstellen und bereitzustellen.

Die Prüfung besteht aus einer Reihe von leistungsbezogenen Aufgaben, die die Kompetenz eines Kandidaten in verschiedenen Aspekten von Kubernetes bewerten. Die Aufgaben umfassen das Bereitstellen von Anwendungen, Konfigurieren und Verwalten von Kubernetes-Ressourcen, Implementieren von Sicherheits- und Netzwerkrichtlinien sowie das Beheben von Problemen. Die Prüfung wird in einer Live-Umgebung durchgeführt, und die Kandidaten müssen die Aufgaben mit einer Befehlszeilenschnittstelle lösen, was es zu einem Test ihrer Fähigkeiten in der realen Welt macht.

>> CKAD Online Tests <<

Echte und neueste CKAD Fragen und Antworten der Linux Foundation

CKAD Zertifizierungsprüfung

Alle Anfang ist schwer. Zögern Sie noch, wie mit der Vorbereitung der Linux Foundation CKAD Prüfung anfangen? Die Prüfungsunterlagen der Linux Foundation CKAD von uns zu kaufen wird ein notwendiger Schritt Ihrer Vorbereitung. Was wir Ihnen bieten, ist nicht nur was Sie möchten, sondern auch was für Ihre Vorbrereitung der Linux Foundation CKAD Prüfung unerlässlich ist. Vielleicht haben Sie noch Hemmungen mit diesem Schritt. So können Sie zuerst die Demo der Linux Foundation CKAD Prüfungsunterlagen herunterladen. Nachdem Sie probiert haben, werden Sie bestimmt diesen Schritt machen.

Die Vorbereitung auf die CKAD -Prüfung erfordert Engagement und harte Arbeit. Die Kandidaten müssen eine starke Grundlage in Kubernetes haben und unter Druck effizient arbeiten können. Die Linux Foundation bietet eine Vielzahl von Ressourcen, um den Kandidaten dabei zu helfen, sich auf die Prüfung vorzubereiten, einschließlich Online -Kursen, Studienleitfäden und Praxisprüfungen. Die Kandidaten sollten auch praktische Erfahrungen mit Kubernetes haben und mit der Befehlszeilenschnittstelle vertraut sein.

Linux Foundation Certified Kubernetes Application Developer Exam CKAD Prüfungsfragen mit Lösungen (Q169-Q174):

169. Frage
You are building a microservice-based application with a frontend service and a backend service. The frontend service needs to communicate with the backend service via a Kubernetes Service. Design and implement a robust solution for the frontend service to discover the backend service's IP address and port, ensuring seamless communication between the services.

Antwort:

Begründung:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a Backend Service:
- Define a Kubernetes Service for the backend service, exposing it on a specific port.
- Ensure the service's 'selector' matches the labels of your backend pods.
2. Configure Frontend Service: - Create a Kubernetes Deployment for the frontend service. - Inject the backend service's name and pon into the frontend container's environment variables. This will allow the frontend service to access the backend service's information.
3. Create a ConfigMap for Backend Service Informatiom - Create a ConfigMap to store the backend service's information, including its name and port
4. Deploy the Services and ConfigMap: - Apply the YAML files for the backend service, frontend deployment, and ConfigMap to your Kubernetes cluster using kubectl apply -f 5. Test Communication: - Access the frontend service (e.g., through a LoadBalancer or Ingress) and ensure it successfully communicates with the backend service. Notes: - This approach utilizes a ConfigMap to store the backend service information, making it easy to update and manage the connection information. - The frontend service can access the backend service's information through environment variables, ensuring consistency in communication. - By utilizing Kubernetes Services, the trontend service can seamlessly communicate with the backend service without knowing the specific IP addresses or ports of individual pods. - The frontend service should be designed to handle potential errors when attempting to connect to the backend service (e.g., timeouts, network issues). Additional Tips for Robust Communication: - Health Checks: LJse Liveness and Readiness probes to ensure that only healthy backend pods are included in the backend service. - Service Discovery: Consider using advanced service discovery mechanisms like Consul or etcd to enable dynamic service discovery and updates. - Network Policies: Apply network policies to control communication between services, improving security and preventing unauthorized access.,

170. Frage
Task:
A pod within the Deployment named buffàle-deployment and in namespace gorilla is logging errors.
1) Look at the logs identify errors messages.
Find errors, including User "system:serviceaccount:gorilla:default" cannot list resource "deployment" [...] in the namespace "gorilla"
2) Update the Deployment buffàlo-deployment to resolve the errors in the logs of the Pod.
The buffalo-deployment 'S manifest can be found at -/prompt/escargot/buffàlo-deployment.yaml See the solution below.

Antwort:

Begründung:
Solution:

171. Frage
Task:
Create a Deployment named expose in the existing ckad00014 namespace running 6 replicas of a Pod.
Specify a single container using the ifccncf/nginx: 1.13.7 image
Add an environment variable named NGINX_PORT with the value 8001 to the container then expose port 8001

Antwort:

Begründung:
See the solution below.
Explanation:
Solution:

172. Frage
You are building a microservices application on Kubernetes, where two services, and 'service-b' , need to communicate with each other securely. 'Service-b' needs to expose a secure endpoint that is only accessible by 'service-a'. Describe how you would implement this using Kubernetes resources, including the configuration for the 'service-b' endpoint.

Antwort:

Begründung:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Define a Kubernetes Secret:
- Create a Kubernetes secret to store the certificate and key pair for 'service-W. This secret will be used to secure the communication.
- Example:
2. Configure 'service-b' Deployment: - Define a Deployment for 'service-b' , specifying a container that uses the secret for TLS. - Ensure that the container has the required dependencies and configuration to use TLS. - Example:
3. Define a Kubernetes Service for 'service-b'.' - Create a Service for 'service-b' that exposes the secure endpoint on a specific port (e.g., 8443) and uses the LoadBalancer' type for external access. - Use the 'targetPort' field to specify the container port that 'service-b' is listening on. - Example:
4. Configure 'service-a' Deployment: - Define a Deployment for 'service-a', specifying a container that uses the secret for TLS when connecting to service-W. - Example:
5. Update 'service-a' Container Configuration: - Within the 'service-a' container, ensure the application is configured to use the certificate and key from the mounted volume ('Ivar/tls/') for secure communication with 'service-b'. 6. Verify Secure Communication: - Use 'kubectl get pods' to check the status of both 'service-a' and 'service-W pods. - Test the communication between 'service-a' and 'service-b' by sending requests from the 'service-a' pod to the secure endpoint of 'service-b'. - Verify that the communication is secure and that 'service-a' can successfully access the endpoint. Notes: - You may need to adjust the port numbers and image names in the examples to match your specific setup. - Make sure you have the certificate and key in the correct format and base64 encoded before creating the Secret. - You can also use other methods like a Service Account and Role-Based Access Control (RBAC) to restrict access to the secure endpoint, if needed. - This is a simplified example and additional security measures may be required based on your application's requirements. ,

173. Frage
Set Configuration Context:
[student@node-1] $ | kubectl
Config use-context k8s

Context
A web application requires a specific version of redis to be used as a cache.
Task
Create a pod with the following characteristics, and leave it running when complete:
* The pod must run in the web namespace.
The namespace has already been created
* The name of the pod should be cache
* Use the Ifccncf/redis image with the 3.2 tag
* Expose port 6379

Antwort:

Begründung:
See the solution below.
Explanation:
Solution:
□

174. Frage
......