# CKAD問題サンプル & CKAD赤本合格率


2025年Tech4Examの最新CKAD PDFダンプおよびCKAD試験エンジンの無料共有：https://drive.google.com/open?id=18IqHJT8X7n2Kq80M4c25An-4HhplQ4o-

これらの問題を解決するための基本的な方法は、社会の発展よりも速いスピードで成長することです。現場では、Linux Foundation認定を取得して、自分自身を改善し、より良いあなたとより良い未来を目指してください。それにより、あなたはあなたの職業で認められます。 CKAD試験トレントは、より大企業に注意を向けさせる能力を証明できます。その後、より良い仕事を取得し、適切な職場に行くための選択肢があります。そして、CKAD試験問題は、98％以上の高い品質と高い合格率で有名です。 CKAD学習ガイドをお試しください。

Linux Foundation CKAD（Certified Kubernetes Application Developer）認定試験は、Kubernetesアプリケーション開発の習熟度を紹介したい個人向けの一般的な認定試験です。 Kubernetesは、コンテナ化されたアプリケーションの展開、スケーリング、および管理の自動化に使用されるオープンソースシステムです。 Kubernetesが人気を得るにつれて、CKAD認定を持つ専門家の需要は急速に増加しています。認定試験は、Kubernetesアプリケーションを設計、構築、構成、および公開する候補者の能力をテストするように設計されています。

**>> CKAD問題サンプル <<**

## 便利なCKAD問題サンプル & 合格スムーズCKAD赤本合格率 | 更新するCKAD資料勉強

この知識が支配的な世界では、知識と実用的な作業能力の組み合わせが非常に重要視されています。Tech4Exam実際の能力を向上させたい場合は、CKAD認定試験に参加できます。 CKAD認定に合格すると、実践能力と知識の両方を高めることができます。また、CKADの最新の質問を購入すると、CKAD試験にスムーズに合格します。

CKAD試験は、実際のKubernetes環境で開発者のスキルをテストするハンズオンのパフォーマンスベースの試験です。試験は、候補者がKubernetesアプリケーションを展開、構成、およびトラブルシューティングする能力をテストするように設計された一連のパフォーマンスベースのタスクで構成されています。試験はオンラインで実施され、世界中どこからでも受験することができます。候補者には2時間の試験時間が与えられ、合格点66％以上を獲得する必要があります。

## Linux Foundation Certified Kubernetes Application Developer Exam 認定 CKAD 試験問題 (Q125-Q130):

**質問 # 125**
You need to implement a mechanism for automatically rolling out new versions of your application pods. This process should be triggered by a change in tne application's container image tag in a Docker Hub repository.

**正解：**

解説：

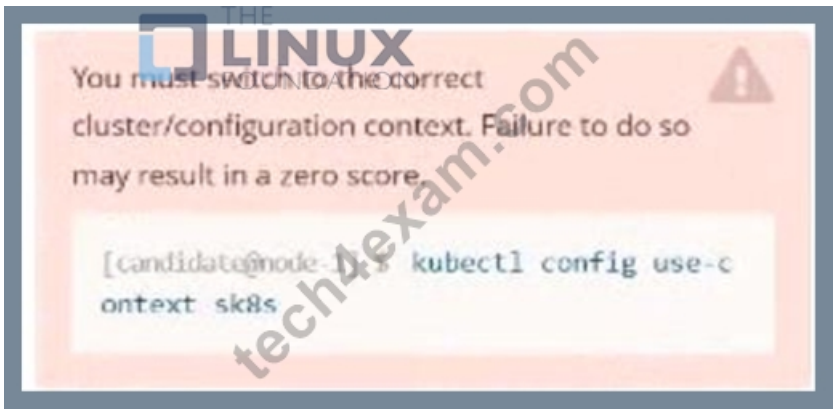
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Configure the Deployment for Rolling Updates:
- IJpdate your application deployment to specify a 'rollingUpdate' strategy
- Set 'maxunavailable' and 'maxSurge' to control the rolling update process-
- Include a 'strategy.type' to 'Rollingupdates
- Set ' imagePullPolicy' to 'Always' to ensure that new images are always pulled from the Docker Hub repository.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: your-application-deployment
  namespace: your-application-namespace
spec:
  replicas: 3
  selector:
    matchLabels:
      app: your-application
  template:
    metadata:
      labels:
        app: your-application
    spec:
      containers:
      - name: your-application
        image: your-docker-hub-account/your-image:latest
        imagePullPolicy: Always
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 0
```



2. Apply the Deployment: - Apply the updated deployment using 'kubectl apply -f your-application-deployment-yamr 3. Push a New Image to Docker Hub: - UPdate your application's container image in the Docker Hub repository and push the new image With a different tag. For example, update the tag from "latest to 'v2'. 4. Monitor the Deployment: - Observe the rolling update process using 'kubectl get pods -l app=your-application'. You should see new pods with the updated image being created and old pods being terminated. 5. Verify the Update: - Once the rolling update is complete, use 'kubectl describe deployment your-application-deployment to verify that the 'updatedReplicas' field matches the 'replicas' field. This confirms that the update was successful. ,

**質問 # 126**
Context






Task:
A pod within the Deployment named buffale-deployment and in namespace gorilla is logging errors.
1) Look at the logs identify errors messages.

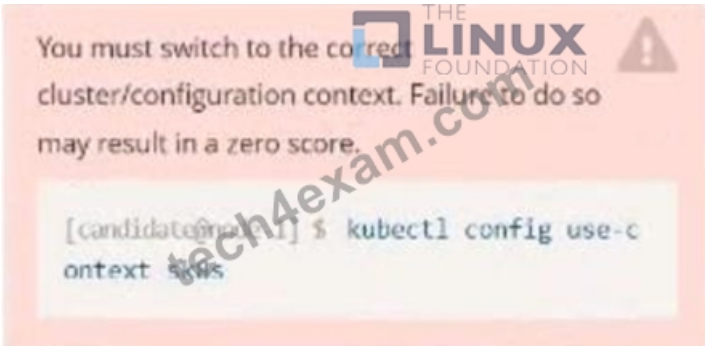
Find errors, including User "system:serviceaccount:gorilla:default" cannot list resource "deployment" [...] in the namespace "gorilla"
2) Update the Deployment buffalo-deployment to resolve the errors in the logs of the Pod.
The buffalo-deployment 'S manifest can be found at -/prompt/escargot/buffalo-deployment.yaml

**正解：**

解説：
Solution:

```
File  Edit  View  Terminal  Tabs  Help
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get  pods -n staging
NAME                                  READY   STATUS    RESTARTS   AGE
backend-deployment-59d449b99d-cxct6   1/1     Running   0          20s
backend-deployment-59d449b99d-h2zjq   0/1     Running   0          9s
backend-deployment-78976f74f5-b8c85   1/1     Running   0          6h40m
backend-deployment-78976f74f5-flfsj   1/1     Running   0          6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment   3/3     3            3           6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment   3/3     3            3           6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl  apply  -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl  get pods -n gorilla
NAME                                  READY   STATUS              RESTARTS   AGE
buffalo-deployment-776844df7f-r5fsb   1/1     Running             0          6h38m
buffalo-deployment-859898c6f5-zx5gj   0/1     ContainerCreating   0          8s
candidate@node-1:~$ kubectl  get deploy -n gorilla
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
buffalo-deployment   1/1     1            1           6h38m
candidate@node-1:~$
```

**質問 # 127**

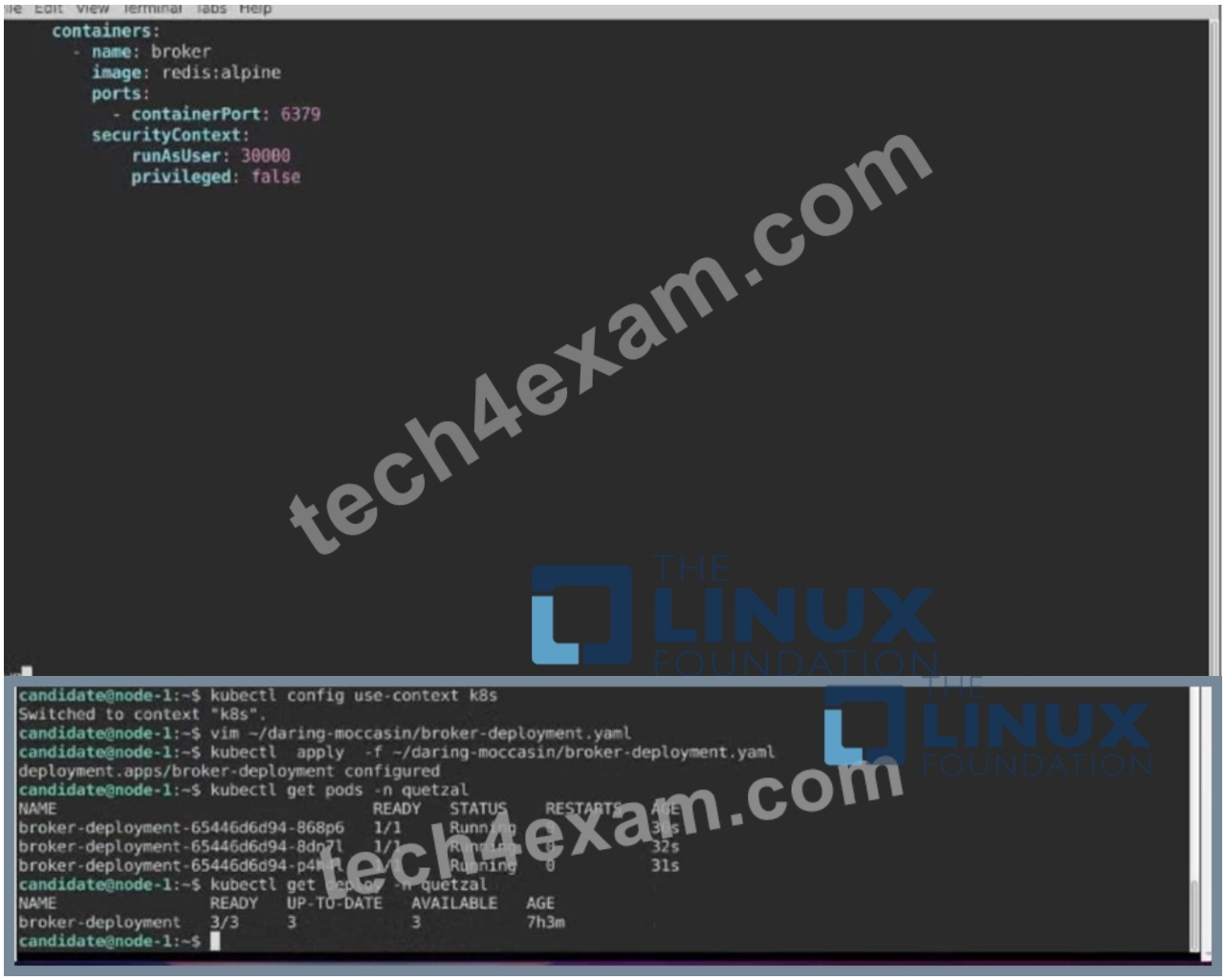



Task:
Modify the existing Deployment named broker-deployment running in namespace quetzal so that its containers.
1) Run with user ID 30000 and
2) Privilege escalation is forbidden
The broker-deployment is manifest file can be found at:

**正解：**

解説：
See the solution below.
Explanation
Solution:


Text Description automatically generated




**質問 # 128**
You are building a microservice that relies on a third-party API for its functionality_ To ensure the reliability and performance of your microservice, you need to implement a robust strategy tor handling API calls. Design a deployment strategy that addresses potential issues with the third-pany API and ensures the stability of your microservice.

**正解：**

解説：
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
I). Use a Deployment:
- Deploy your microservice using a Deployment. Deployments provide a robust mechanism for managing and scaling your microservices, making it easy to update and manage your application.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-consuming-service-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: api-consuming-service
  template:
    metadata:
      labels:
        app: api-consuming-service
    spec:
      containers:
      - name: api-consuming-service
        image: your-image-repo:latest
        env:
        - name: API_ENDPOINT
          value: your-api-endpoint
        - name: API_KEY
          valueFrom:
            secretKeyRef:
              name: api-credentials
              key: key
```

2. Secure API Credentials: - Store API credentials (like API keys or tokens) securely using a Kubernetes Secret. This prevents credentials from being exposed in plain text within your deployments.

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: api-credentials
stringData:
  key: your-api-key
```

3. Implement Retry Mechanisms: - Add retry logic to your code to handle transient errors (like network hiccups or temporary service outages) during API calls. This helps ensure that your microservice can recover from temporary issues and continue functioning. 4. Utilize Rate Limiting: - Implement rate limiting to prevent your microservice from ovenvhelming the third-party API. This helps protect both your microservice and the API from performance degradation- 5. Use a Circuit Breaker Pattern: - Integrate a circuit breaker pattern into your API call handling. This pattern helps prevent cascading failures by automatically stopping requests to the third-party API it it is experiencing prolonged outages or errors- 6. Consider a Proxy or Gateway: - Implement a proxy or gateway layer between your microservice and the third-party API. This layer can help with request routing, load balancing, security, and performance optimization. 7. Monitor API Calls: - Implement monitoring and logging to track API call performance and identify potential issues. This allows you to proactively identify and address problems before they impact your microservice's reliability 8. Utilize Caching: - Consider caching API responses to reduce the load on the third-party API and improve the response time of your microservice. 9. Implement Fallbacks: - Have fallback mechanisms in place if the third-party API is unavailable. This could involve returning default data or using alternative data sources to provide a degraded but functional experience. 10. Consider Using a Service Mesh: - For complex microservice architectures, consider implementing a service mesh like Istio. Service meshes provide features like traffic management, security, observability, and resilience, which can be very beneficial for managing interactions with third-party APIs.,

**質問 # 129**


Context
You are tasked to create a secret and consume the secret in a pod using environment variables as follow:
Task
* Create a secret named another-secret with a key/value pair; key1/value4
* Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key 1, using COOL_VARIABLE as the name for the environment variable inside the pod See the solution below.

**正解：**

解説：
Explanation
Solution:

```
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-4kvr5   kubernetes.io/service-account-token   3      2d11h
some-secret           Opaque                                1      5s
student@node-1:~$ kubectl run nginx-secret --image=nginx --dry-run=client -o yaml > nginx_secret
.yml
student@node-1:~$ vim nginx_secret.yml
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
~
"nginx_secret.yml" 15L, 253C                                  1,1           All
```


**質問 # 130**

.....

**CKAD赤本合格率**: https://www.tech4exam.com/CKAD-pass-shiken.html

- CKAD試験の準備方法 | 有難いCKAD問題サンプル試験 | 認定するLinux Foundation Certified Kubernetes Application Developer Exam赤本合格率 □ （ www.xhs1991.com ）に移動し、➤ CKAD □を検索して、無料でダウンロード可能な試験資料を探しますCKAD模擬試験問題集
- CKAD練習問題◂CKAD関連資格知識 □ CKAD最新知識 □▸ www.goshiken.com◂の無料ダウンロード《 CKAD 》ページが開きますCKAD受験対策解説集

- CKAD資格試験 □ CKAD受験対策解説集 □ CKADクラムメディア □⇒ www.shikenpass.com⇐にて限定無料の[ CKAD ]問題集をダウンロードせよCKAD認証資格
- CKAD日本語関連対策 □ CKAD試験解説 □ CKAD日本語版問題解説 □ 検索するだけで✔ www.goshiken.com □✔□から▸ CKAD ◂を無料でダウンロードCKAD資格試験
- CKAD資格試験 □ CKAD試験資料 □ CKADトレーニング □➥ www.shikenpass.com □にて限定無料の➤ CKAD □問題集をダウンロードせよCKAD練習問題
- Linux Foundation CKAD問題サンプルは主要材料 - CKAD赤本合格率 □ 検索するだけで[ www.goshiken.com ]から➠ CKAD □を無料でダウンロードCKAD合格対策
- Linux Foundation CKAD問題サンプルは主要材料 - CKAD赤本合格率 !! URL ☀ www.passtest.jp □☀□をコピーして開き、➡ CKAD □□□を検索して無料でダウンロードしてくださいCKAD最新知識
- 素敵なCKAD問題サンプル一回合格-ハイパスレートのCKAD赤本合格率 □ □ www.goshiken.com □で➡ CKAD □□□を検索して、無料で簡単にダウンロードできますCKAD認証資格
- CKAD最新知識 □ CKAD技術内容 ※ CKAD日本語版問題解説 □ 今すぐ□ www.it-passports.com □で➤ CKAD □を検索して、無料でダウンロードしてくださいCKAD日本語版問題解説
- 検証するCKAD丨正確的なCKAD問題サンプル試験丨試験の準備方法Linux Foundation Certified Kubernetes Application Developer Exam赤本合格率 □ 【 www.goshiken.com 】サイトにて最新{ CKAD }問題集をダウンロードCKADトレーニング
- 素敵なCKAD問題サンプル一回合格-ハイパスレートのCKAD赤本合格率 □☀ www.passtest.jp □☀□から簡単に（ CKAD ）を無料でダウンロードできますCKAD関連資格知識
- pdesbonnets.alboompro.com, www.stes.tyc.edu.tw, 203060.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, skilled-byf.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, yy.hackp.com.cn, www.stes.tyc.edu.tw, Disposable vapes

P.S. Tech4ExamがGoogle Driveで共有している無料かつ新しいCKADダンプ：https://drive.google.com/open?id=18IqHJT8X7n2Kq80M4c25An-4HhplQ4o-