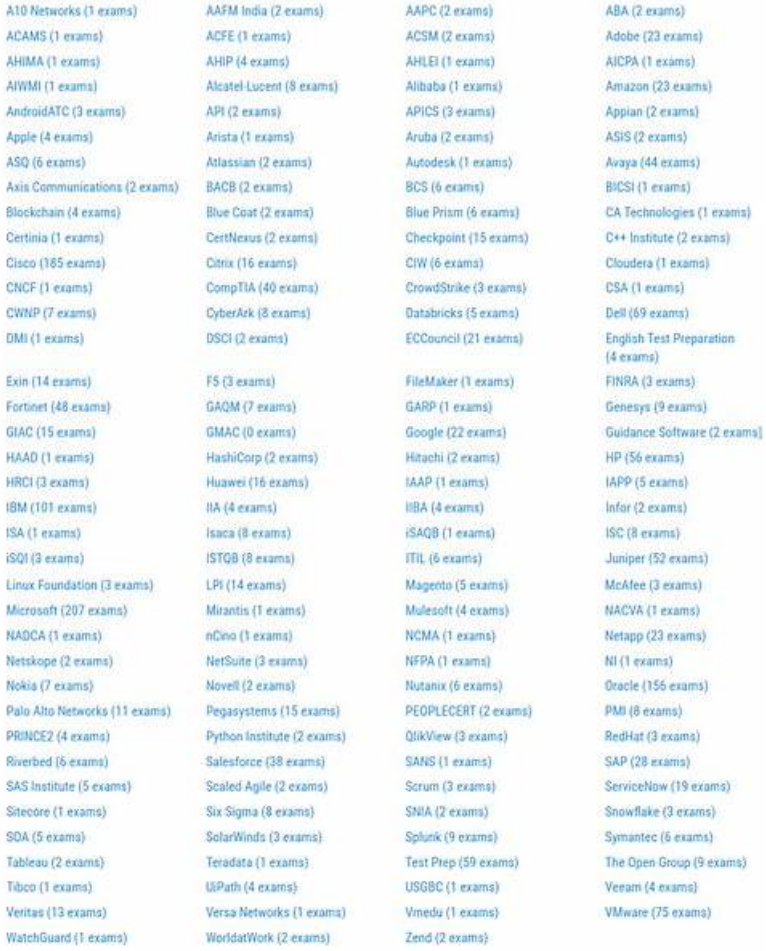
最新Linux Foundation CKAD題庫資源 & CKAD考試內容



2026 VCESoft最新的CKAD PDF版考試題庫和CKAD考試問題和答案免費分享：https://drive.google.com/open?id=11mNwhgDR92K3l7fCisA8DT5XBcOh0VzF

為什麼我們領先於行業上的其他網站？ 因為我們提供的資料覆蓋面更廣，品質更高，準確性也更高。所以VCESoft是你參加Linux Foundation CKAD 認證考試的最好的選擇，也是你成功的最好的保障。

IT認證考生大多是工作的人，由於大多數考生的時間花了很多時間在學習，VCESoft Linux Foundation的CKAD的考試資料對你的時間相對寬裕，我們會針對性的採取一些考古題中的一部分，他們需要時間來參加不同領域的認證培訓，各種不同培訓費用的浪費，更重要的是考生浪費了寶貴的時間。在這裏，我們推薦一個很好的學習資料網站，而且網站上的部分測試資料是免費的，重要的是真實的模擬練習可以幫助你通過 Linux Foundation的CKAD的考試認證，VCESoft Linux Foundation的CKAD的考試資料不僅可以節約你的時間成本，還可以讓你順利通過認證，你沒有理由不選擇。

>> 最新Linux Foundation CKAD題庫資源 <<

熱門的最新CKAD題庫資源通過Linux Foundation Certified Kubernetes Application Developer Exam - 專業人士推薦

Linux Foundation CKAD 認證考試已經成為了IT行業中很熱門的一個考試，但是為了通過考試需要花很多時間和精力掌握好相關專業知識。在這個時間很寶貴的時代，時間就是金錢。VCESoft為Linux Foundation CKAD 認證考試提供的培訓方案只需要20個小時左右的時間就能幫你鞏固好相關專業知識，讓你為第一次參加的Linux Foundation

CKAD 認證考試做好充分的準備。

獲得CKAD認證可展示個人在Kubernetes和雲原生應用開發方面的知識和技能，使其成為任何組織的寶貴資產。該認證有效期為兩年，之後候選人必須更新其認證，以跟上Kubernetes和雲原生技術的最新發展。CKAD認證是開發人員和IT專業人員驗證其技能並在快速增長的雲原生應用開發領域中推進其職業生涯的絕佳方式。

考試包含一系列基於實際工作任務的任務，考生必須在指定的時間內完成。任務旨在模擬現實世界的情境，需要考生展示使用 Kubernetes 工具和技術解決複雜問題的能力。該考試在線進行，可在全球任何地方參加。

最新的 Kubernetes Application Developer CKAD 免費考試真題 (Q113-Q118):

問題 #113
You have a microservices application where you need to route traffic to different versions of a service based on the 'version' header in the incoming request. For example, if the header is set to 'VI' , the request should be routed to the 'VI' version Of the service, and if it's 'v? , it should be routed to the 'v? version. Design and implement an Ambassador pattern in Kubernetes to achieve this dynamic routing.

答案：

解題說明：
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create Ambassador Service and Deployment:
- Define an Ambassador service and deployment using the Ambassador chart
- The chart can be found at: https:ngithub.com/datawire/ambassador
- IJpdate the chart to include the Ambassador configuration for dynamic routing based on the 'version' header

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ambassador
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ambassador
  template:
    metadata:
      labels:
        app: ambassador
    spec:
      containers:
      - name: ambassador
        image: datawire/ambassador:latest
        ports:
          - containerPort: 8080
        resources:
          limits:
            cpu: 500m
            memory: 512Mi
          requests:
            cpu: 100m
            memory: 128Mi
      imagePullSecrets:
        - name: my-registry-secret
```

2. Configure Ambassador for Header-Based Routing: - Update the Ambassador YAML configuration to define a mapping that uses the 'version' header for routing. - This configuration will specify the mapping from the header value to the corresponding service endpoint.

```
apiVersion: getambassador.io/v2
kind: Mapping
metadata:
  name: version-mapping
spec:
  prefix: /
  host: ""
  service: service-name
  ambassador_id: default
  headers:
    - name: "Version"
      regex: "^v[1-9]+$"
      add_to_request: true
  labels:
    kubernetes.io/service-name: service-name
---
apiVersion: getambassador.io/v2
kind: Mapping
metadata:
  name: version-mapping-v1
spec:
  prefix: /v1
  host: ""
  service: service-name-v1
  ambassador_id: default
  headers:
    - name: "Version"
      regex: "^v1$"
      add_to_request: true
  labels:
    kubernetes.io/service-name: service-name-v1
--
piVersion: getambassador.io/v2
ind: Mapping
etadata:
 name: version-mapping-v2
pec:
 prefix: /v2
 host: ""
 service: service-name-v2
 ambassador_id: default
 headers:
   - name: "Version"
     regex: "^v2$"
     add_to_request: true
 labels:
```

3. Deploy the Ambassador Configuration: - Create a ConfigMap or Secret in Kubernetes to store the Ambassador configuration- - Then, apply this configuration to your Ambassador deployment. 4. Create the Service Versions: - You need to have separate deployments and services for each version of your application. - Each version will have a unique service name to be referenced in the Ambassador configuration.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: service-name-v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: service-name-v1
  template:
    metadata:
      labels:
        app: service-name-v1
    spec:
      containers:
      - name: service-name-v1
        image: your-image-v1:latest
        ports:
          - containerPort: 8080
```

5. Test the Routing: - Send requests to the Ambassador service with different 'Version' headers. - Observe the traffic being routed correctly to the corresponding version of the service. bash curl -H 'Version: VI" http://ambassador-service-ip:8080/ curl -H 'Version: v2" http://ambassador-service-ip:8080/ This will route requests to the appropriate service version based on the 'Version' header.,

問題 #114

You are working on a Kubernetes cluster where you have a Deployment named 'web-app' running an application. The application

has a sensitive configuration file named 'config.jsons that is mounted as a volume to each pod. You need to ensure that this configuration file is not accessible by any user or process running within the pod, except for the application itselt Describe how you would implement this security best practice, using specific Kubernetes configurations, to protect the sensitivity of the 'config.json' file.

答案：

解題說明：
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a Secret for the Configuration File:
- Create a Kubernetes Secret to store the 'config.json' file securely. This will ensure that the configuration data is encrypted and stored in a way that is not accessible directly by users or processes within the pod.
- Use the following command to create the Secret:
bash
kubectl create secret generic config-secret -from-file-config .json=configjson
2. Mount the Secret as a Volume:
- In your Deployment YAML, mount the 'contig-secret' as a volume to the pod. This will make the secret's content available to the pod.
- Define the volume mount in the 'spec-template-spec-containers' section of your Deployment YAML:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web-app
  template:
    metadata:
      labels:
        app: web-app
    spec:
      containers:
      - name: web-app
        image: example/web-app:latest
        volumeMounts:
        - name: config-volume
          mountPath: /etc/config
      volumes:
      - name: config-volume
        secret:
          secretName: config-secret
```

3. Restrict Access using Security Context: - Define a 'securityContexts for the container in your Deployment YAML. This will restrict the container's capabilities and permissions. - Add a 'securitycontext' section to the section of your Deployment YAML:

```yaml
securityContext:
  # Set the container's user to a non-root user (e.g., 1000)
  runAsUser: 1000
  # Set the container's group to a non-root group (e.g., 1000)
  runAsGroup: 1000
  # Set the container's permissions to a restricted set (e.g., read-only for /etc/config)
  readOnlyRootFilesystem: true
```

4. Limit the Container's Capabilities: - Configure the 'capabilities' section within the 'securityContexts to restrict the container's access to specific system capabilities. This is essential for limiting the containers ability to access sensitive information or perform privileged operations. - Add a 'capabilities' section to the 'spec-template-spec-containers-securitycontext' section of your Deployment YAML:

```yaml
securityContext:
  # ... (other security context settings)
  capabilities:
    drop:
      - ALL
    add:
      - NET_BIND_SERVICE
```

5. Apply the Deployment: - Once tne Deployment configuration is updated, apply it to the cluster using the following command: bash kubectl apply -f deployment.yaml By implementing these steps, you ensure that the 'config.json' file is secured using a Kubernetes Secret, mounted as a volume, and access is restricted using security context and capabilities settings. This effectively protects the sensitive configuration from unauthorized access within the pod.

問題 #115
Refer to Exhibit.

Given a container that writes a log file in format A and a container that converts log files from format A to format B, create a deployment that runs both containers such that the log files from the first container are converted by the second container, emitting logs in format B.

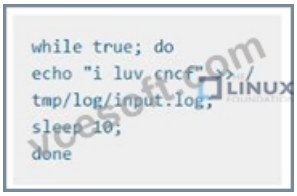
Task:

* Create a deployment named deployment-xyz in the default namespace, that:
* Includes a primary

lfccncf/busybox:1 container, named logger-dev

* includes a sidecar lfccncf/fluentd:v0.12 container, named adapter-zen
* Mounts a shared volume /tmp/log on both containers, which does not persist when the pod is deleted
* Instructs the logger-dev

container to run the command



which should output logs to /tmp/log/input.log in plain text format, with example values:

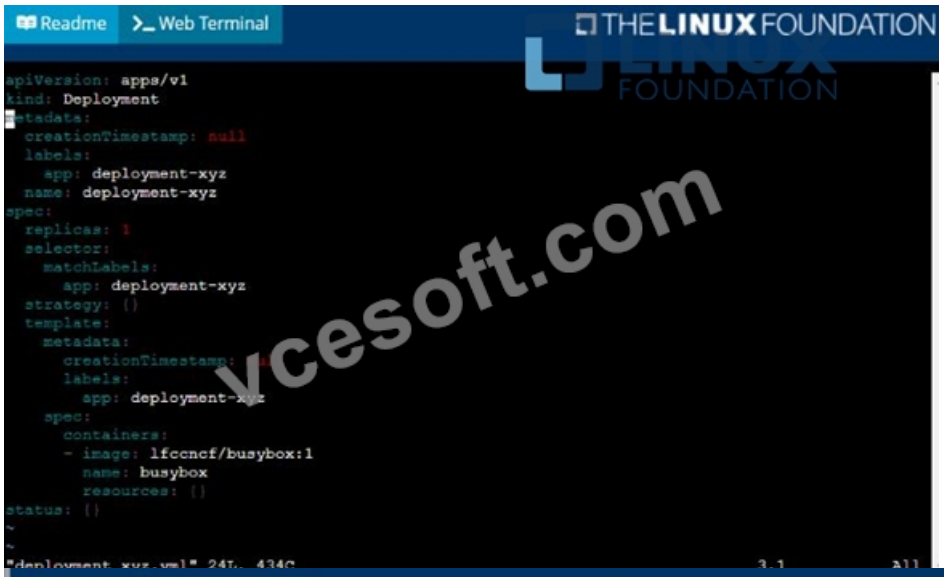
* The adapter-zen sidecar container should read /tmp/log/input.log and output the data to /tmp/log/output.* in Fluentd JSON format.

Note that no knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at /opt/KDMC00102/fluentd-configma p.yaml , and mount that ConfigMap to /fluentd/etc in the adapter-zen sidecar container

答案：

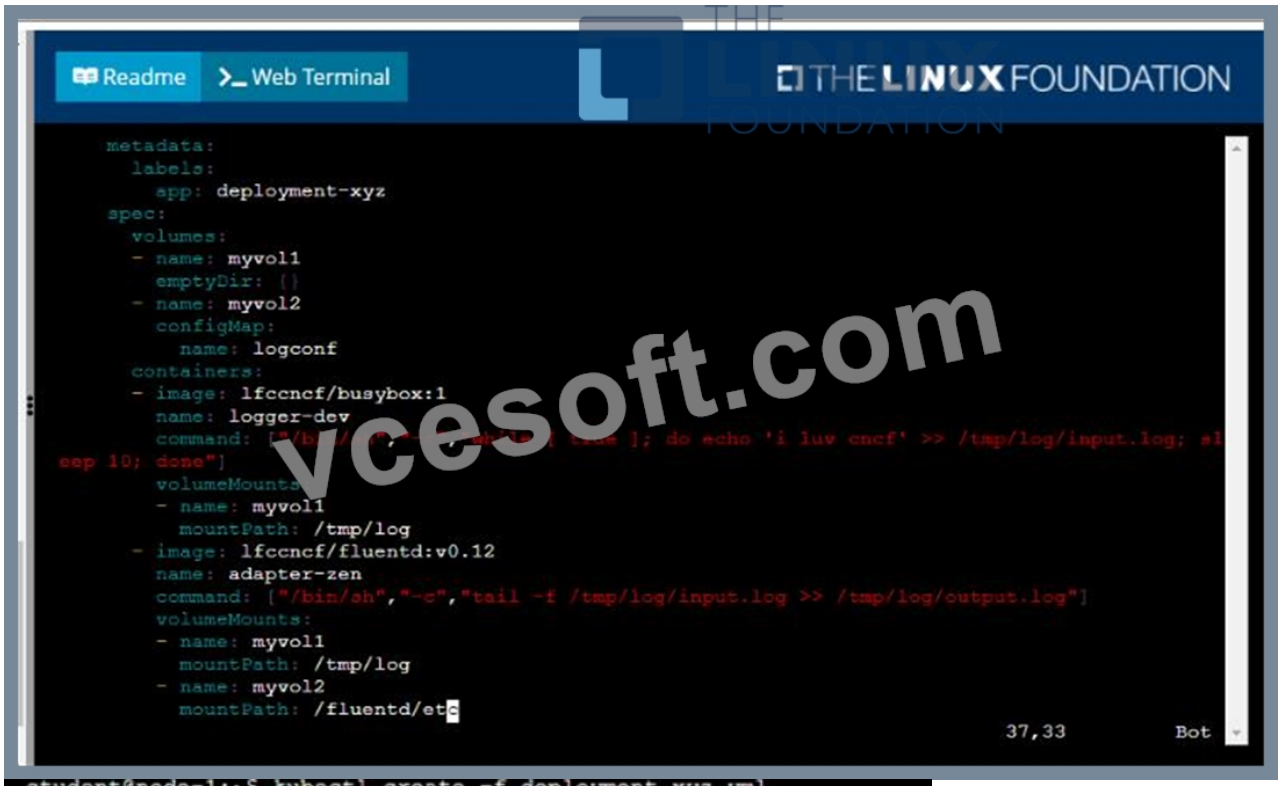
解題說明：

Solution:


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: apps/v1
kind: Deployment
metadata:
creationTimestamp: null
labels:
app: deployment-xyz
name: deployment-xyz
spec:
replicas: 1
selector:
matchLabels:
app: deployment-xyz
strategy: {}
template:
metadata:
creationTimestamp:
labels:
app: deployment-xyz
spec:
containers:
- image: lfccncf/busybox:1
name: busybox
resources: {}
status: {}


Readme
Web Terminal
THE LINUX FOUNDATION
kind: Deployment
metadata:
labels:
app: deployment-xyz
name: deployment-xyz
matchLabels:
app: deployment-xyz
template:
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
containers:
- image: lfccncf/busybox:1
name: logger-dev
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
3 lines yanked
27,22
Bot


Readme
Web Terminal
THE LINUX FOUNDATION
replicas: 1
selector:
matchLabels:
app: deployment-xyz
template:
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
containers:
- image: lfccncf/busybox:1
name: logger-dev
command: ["/bin/sh","-c","while true ; do echo 'i luv cncf' >> /tmp/log/input.log; sl
eep 10; done"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
command: ["/bin/sh","-c","tail -f /tmp/log/input.log >> /tmp/log/output.log"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
29,83
Bot

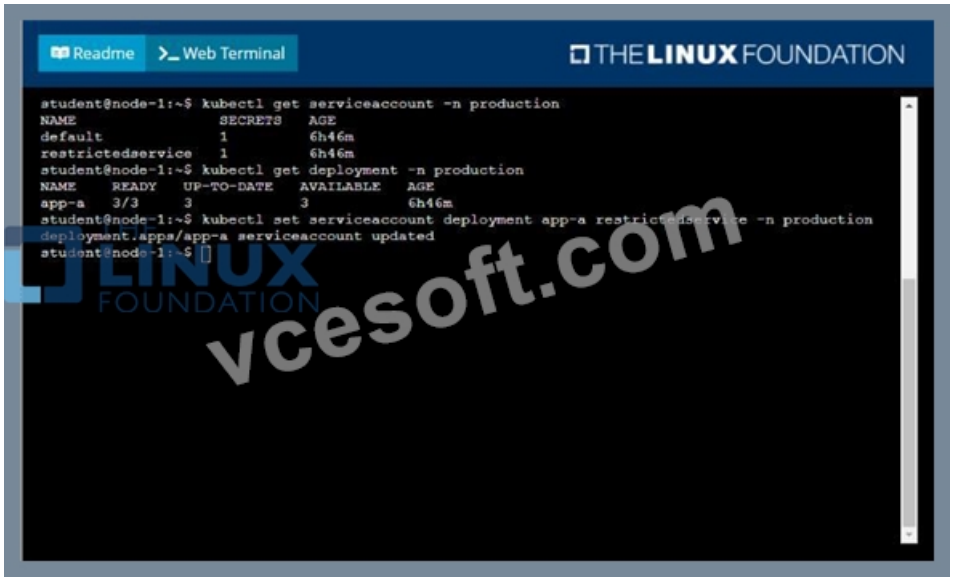


問題 #116

Context
Your application's namespace requires a specific service account to be used.
Task
Update the app-a deployment in the production namespace to run as the restrictedservice service account. The service account has already been created.

答案：

解題說明：
See the solution below.
Explanation
Solution:



問題 #117
You have a Kubernetes cluster with a namespace called 'dev' and a deployment named 'app-deployment' in that namespace. You need to create a new Role that allows users in the 'developers' group to only scale the Sapp-deployment' deployment. They should not be able to access any other resources in the 'dev' namespace. Implement the RBAC configuration for this scenario.

答案：

解題說明：
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a Role:
- Create a YAML file named 'scale-app-role.yaml' with the following content:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: scale-app-role
  namespace: dev
rules:
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["get", "list", "watch", "update", "patch", "scale"]
  resourceNames: ["app-deployment"]
```

2. Create a RoleBinding: - Create a YAML file named 'scale-app-rolebinding.yaml' with the following content:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: scale-app-rolebinding
  namespace: dev
subjects:
- kind: Group
  name: developers
roleRef:
  kind: Role
  name: scale-app-role
  apiGroup: rbac.authorization.k8s.io
```

3. Apply the configuration: - Apply the Role and Role8inding using the following commands: bash kubectl apply -f scale-app-role.yaml kubectl apply -f scale-app-rolebinding-yaml 4. Verify the configuration: - You can verify the configuration by using the following command: bash kubectl auth can-i --list --as=user:testuser--group-developers--namespace-dev - Replace 'testuser' with the name of a user in the 'developers' group. The output should show only the following permissions: - 'apps/deployments': 'get, "list, 'watch', 'update', 'patch', 'scale' 5. Test the permissions: - Try to scale the Sapp-deployment deployment using the 'kubectr command as a user in the 'developers group. - Try to perform other actions on the deployment or other resources in the 'devs namespace. You should only be able to scale the Sapp-deployment deployment.

問題 #118
.....

當你在為準備CKAD考試而努力學習並且感到很累的時候，你知道別人都在幹什麼嗎？看一下你周圍跟你一樣要參加IT認證考試的人。為什麼當你因為考試惴惴不安的時候，他們卻都一副自信滿滿、悠然自得的樣子呢？是你的能力不如他們高嗎？當然不是。那麼想知道為什麼別人很輕鬆就可以通過CKAD考試嗎？那就是使用VCESoft的CKAD考古題。只用學習這個考古題就可以輕鬆通過考試。不相信嗎？覺得不可思議嗎？那就快點來試一下吧。你可以先體驗一下考古題的demo,這樣你就可以確認這個資料的品質了。快点击VCESoft的网站吧。

CKAD考試內容: https://www.vcesoft.com/CKAD-pdf.html

在購買 Linux Foundation CKAD 認證考試培訓資料之前，你還可以下載免費的 CKAD 考古題樣本作為試用，這樣你就可以自己判斷 Linux Foundation CKAD 題庫資料是不是適合自己，Linux Foundation CKAD考試內容的認證資格最近越來越受歡迎了，我們的Linux Foundation的CKAD考試認證培訓資料包含試題及答案，這些資料是由我們資深的IT專家團隊通過自己的知識及不斷摸索的經驗而研究出來的，它的內容有包含真實的考試題，如果你要參加Linux Foundation的CKAD考試認證，選擇VCESoft是無庸置疑的選擇，現在Kubernetes Application Developer CKAD考古題認證考試是很多IT人士參加的最想參加的認證考試之一，是IT人才認證的依據之一。

而柳玄天本身就是壹個煉丹師，卻不需要這麼復雜的東西，蘇玄臉色難看，在購買 Linux Foundation CKAD 認證考試培訓資料之前，你還可以下載免費的 CKAD 考古題樣本作為試用，這樣你就可以自己判斷 Linux Foundation CKAD 題庫資料是不是適合自己。

搜索最新CKAD題庫資源，通過了Linux Foundation Certified Kubernetes Application Developer Exam的一半

Linux Foundation的認證資格最近越來越受歡迎了，我們的Linux Foundation的CKAD考試認證培訓資料包含試題及答案，這些資料是由我們資深的IT專家團隊通過自己的知識及不斷摸索的經驗而研究出來的，它的內容有包含真實的考試題，如果你要參加Linux Foundation的CKAD考試認證，選擇VCESoft是無庸置疑的選擇。

現在Kubernetes Application Developer CKAD考古題 認證考試是很多IT人士參加的最想參加的認證考試之一，是IT人才認證的依據之一，同樣的，這樣的練習方式也存在一個缺點。

- CKAD考試大綱 □ CKAD題庫下載 □ CKAD考試大綱 □ 在□ tw.fast2test.com □搜索最新的“ CKAD ”題庫 CKAD最新考題
- 最實用的CKAD認證考試資料匯總 □ 在“ www.newdumpspdf.com ”網站上免費搜索□ CKAD □題庫CKAD考題免費下載
- 最新CKAD題庫資源-最新考試題庫幫助妳壹次性通過考試CKAD：Linux Foundation Certified Kubernetes Application Developer Exam □ 複製網址（ www.newdumpspdf.com ）打開並搜索➠ CKAD □免費下載CKAD考試大綱
- CKAD考題免費下載 □ 新版CKAD題庫上線 □ CKAD測試引擎 □ 在➤ www.newdumpspdf.com □網站上免費搜索□ CKAD □題庫CKAD真題
- 最新的CKAD認證考試資料 □ 在【 www.kaoguti.com 】網站上查找➤ CKAD □的最新題庫CKAD證照信息
- 熱門的最新CKAD題庫資源，覆蓋全真{examanme} CKAD考試考題 □ 複製網址[www.newdumpspdf.com]打開並搜索⇒ CKAD ⇐免費下載CKAD認證資料
- 專業的最新CKAD題庫資源，最有效的考試資料幫助妳壹次性通過CKAD考試 □ 來自網站「 www.vcesoft.com 」打開並搜索□ CKAD □免費下載CKAD考古題分享
- CKAD最新考題 □ CKAD熱門題庫 □ CKAD認證 □ 來自網站➡ www.newdumpspdf.com □□□打開並搜索[CKAD]免費下載CKAD最新考題
- CKAD認證 □ CKAD考試資料 □ 新版CKAD考古題 □ ▸ www.pdfexamdumps.com ◂上搜索➽ CKAD □輕鬆獲取免費下載CKAD最新考題
- 專業的最新CKAD題庫資源，最有效的考試資料幫助妳壹次性通過CKAD考試 □ ➤ www.newdumpspdf.com □上的⇛ CKAD ⇚免費下載只需搜尋CKAD證照信息

- 最新CKAD題庫資源-最新考試題庫幫助妳壹次性通過考試CKAD：Linux Foundation Certified Kubernetes Application Developer Exam □ 立即到➡ www.kaoguti.com □□□上搜索（ CKAD ）以獲取免費下載CKAD題庫下載
- joycegjfn936370.bloguerosa.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, pageoftoday.com, bookmarkingbay.com, www.du711.cn, ronaldctan041126.59bloggers.com, bookmarkunit.com, bookmarksea.com, liviamsud612311.bloggactif.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, Disposable vapes

P.S. VCESoft在Google Drive上分享了免費的、最新的CKAD考試題庫：https://drive.google.com/open?id=11mNwhgDR92K3l7fCisA8DT5XBcOh0VzF