

시험패스가능한CKAD최고품질덤프데모덤프공부문제

SAP C_4H630_21

SAP Certified Development Associate - SAP Customer Data Platform

3

- A. Generic Webservice Provider
- B. Connector Studio
- C. Server API integrations
- D. Web client application

정답B,C

질문 #27

.....

여러분은 우선 우리 PassTIP사이트에서 제공하는SAP인증C_4H630_21시험덤프의 일부 문제와 답을 채
험해보세요. 우리 PassTIP를 선택해주신다면 우리는 최선을 다하여 여러분이 꼭 한번에 시험을 패스할 수
있도록 도와드리겠습니다.만약 여러분이 우리의 인증시험덤프를 보시고 시험이랑 틀려서 패스를 하지 못
하였다면 우리는 무조건 덤프비용전부를 환불해드립니다.

C_4H630_21덤프: https://www.passtip.net/C_4H630_21-pass-exam.html

PassTIP의SAP인증 C_4H630_21덤프의 인지도는 아주 높습니다. PassTIP의SAP인증 C_4H630_21덤프
는 시험패스율이 거의 100%에 달하여 많은 사랑을 받아왔습니다. SAP C_4H630_21시험을 가장 쉽게
합격하는 방법이 PassTIP의SAP C_4H630_21 덤프를 마스터하는 것입니다. SAP C_4H630_21시험덤프
로도 성공으로 향하는 길에는 많은 방법과 방식이 있습니다. 우리 PassTIP에서는 최고이자 최선의SAP
인증C_4H630_21덤프자료를 제공 함으로 여러분을 도와SAP 인증C_4H630_21인증자격증을 쉽게 취
득할 수 있게 해드립니다.만약 아직도SAP 인증C_4H630_21시험패스를 위하여 고군분투하고 있다면 바
로 우리 PassTIP를 선택함으로 여러분의 고민을 날려버릴수 있습니다. SAP C_4H630_21 덤프를 한번 믿
고SAP C_4H630_21시험에 두려움없이 맞서보세요.

우리는 손사래를 치며 어색하게 웃었다. 유미까지 부추겼지만 우리는 고개를 저었다. PassTIP의SAP인증
C_4H630_21덤프의 인지도는 아주 높습니다. PassTIP의SAP인증 C_4H630_21덤프는 시험패스율이
거의 100%에 달하여 많은 사랑을 받아왔습니다.

**C_4H630_21시험덤프데모 덤프에는 ExamName} 시험문제의 모
든 유형이 포함**

SAP C_4H630_21시험을 가장 쉽게 합격하는 방법이 PassTIP의SAP C_4H630_21 덤프를 마스터한
것입니다. 성공으로 향하는 길에는 많은 방법과 방식이 있습니다. 우리 PassTIP에서는 최고이자 최선
의SAP 인증C_4H630_21덤프자료를 제공 함으로 여러분을 도와SAP 인증C_4H630_21인증자격증을
쉽게 취득할 수 있게 해드립니다.만약 아직도SAP 인증C_4H630_21시험패스를 위하여 고군분투하고 있
다면 바로 우리 PassTIP를 선택함으로 여러분의 고민을 날려버릴수 있습니다.

Tags: C_4H630_21시험덤프데모,C_4H630_21덤프,C_4H630_21퍼펙트 덤프 최신문
제,C_4H630_21유료한 시험자료,C_4H630_21시험패스 가능한 인증공부자료

C_4H630_21시험덤프데모 & C_4H630_21덤프

참고: KoreaDumps에서 Google Drive로 공유하는 무료 2026 Linux Foundation CKAD 시험 문제집이 있습니다:
https://drive.google.com/open?id=1XGJ63-ght0qH8nE7YpIyO6ERXp9qotz_

KoreaDumps에서는 최선을 다해 여러분이Linux Foundation CKAD인증시험을 패스하도록 도와 줄 것이며 여러분은
KoreaDumps에서Linux Foundation CKAD덤프의 일부분의 문제와 답을 무료로 다운받으실 수 있습니다. KoreaDumps
선택함으로Linux Foundation CKAD인증시험통과는 물론KoreaDumps 제공하는 일년무료 업데이트서비스를 제공받
을 수 있으며 KoreaDumps의 인증덤프로 시험에서 떨어졌다면 100% 덤프비용 전액환불을 약속 드립니다.

CKAD 인증 시험은 Kubernetes 응용 프로그램 개발과 관련된 광범위한 주제를 다루는 실습 시험입니다. 여기에는
응용 프로그램 배포 및 구성, Kubernetes 객체 작성 및 관리, 응용 프로그램 문제 해결 및 Kubernetes API 작업이 포함
됩니다. 이 시험은 실제 시나리오에서 Kubernetes와의 작업 능력을 테스트하도록 설계되었으며 명령 줄 도구 및
Kubernetes API를 사용해야 합니다.

Linux Foundation CKAD (Linux Foundation Certified Kubernetes Application Developer) 자격증 시험은 Kubernetes에서 애플
리케이션을 설계하고 배포하는 데 전문성을 증명하고자 하는 전문가들에게 매우 인기 있는 자격증입니다.
Kubernetes는 컨테이너화된 애플리케이션의 배포, 확장 및 관리를 자동화하는 오픈소스 플랫폼입니다. CKAD 자격
증은 후보자의 Kubernetes 애플리케이션 개발 및 배포 기술을 검증하기 위해 설계되었습니다.

>> CKAD최고품질 덤프데모 <<

CKAD최고품질 덤프데모최신버전 인증덤프자료

최신 Kubernetes Application Developer CKAD 무료샘플문제 (Q65-Q70):

Refer to Exhibit.



Task:

Create a Deployment named `expose` in the existing `ckad00014` namespace running 6 replicas of a Pod.

Specify a single container using the `ifcnf/nginx: 1.13.7` image Add an environment variable named `NGINX_PORT` with the value 8001 to the container then expose port 8001

정답:

설명:

Solution:

[illegible]

```
File Edit View Terminal Tabs Help
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: expose
    name: expose
    namespace: ckad00014
spec:
  replicas: 6
  selector:
    matchLabels:
      app: expose
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: expose
    spec:
      containers:
        - image: lfccncf/nginx:1.13.7
          name: nginx
          ports:
            - containerPort: 8001
          env:
            - name: NGINX_PORT
              value: "8001"
```

candidate@node-1:~\$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~\$ kubectl create deployment expose -n ckad00014 --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml > dep.yaml
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$
candidate@node-1:~\$ vim dep.yaml
candidate@node-1:~\$ kubectl create -f dep.yaml
deployment.apps/expose created
candidate@node-1:~\$ kubectl get pods -n ckad00014

NAME	READY	STATUS	RESTARTS	AGE
expose-85dd99d4d9-25675	0/1	ContainerCreating	0	6s
expose-85dd99d4d9-4fhcc	0/1	ContainerCreating	0	6s
expose-85dd99d4d9-fl7j	0/1	ContainerCreating	0	6s
expose-85dd99d4d9-tt6rm	0/1	ContainerCreating	0	6s
expose-85dd99d4d9-vjd8b	0/1	ContainerCreating	0	6s
expose-85dd99d4d9-vtzpq	0/1	ContainerCreating	0	6s

```
candidate@node-1:~$ kubectl get deploy -n ckad00014
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
expose	6/6	6	6	15s

```
candidate@node-1:~$
```

질문 # 66
Refer to Exhibit.



Context

As a Kubernetes application developer you will often find yourself needing to update a running application.

Task

Please complete the following:

- * Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%
- * Perform a rolling update of the web1 deployment, changing the lfccncf/nginx image version to 1.13
- * Roll back the app deployment to the previous version

정답:

설명:

Solution:



```
Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl edit deployment app -n kdpd00202
deployment.apps/app edited
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 8 of 10 updated replicas are available...
Waiting for deployment "app" rollout to finish: 9 of 10 updated replicas are available...
deployment "app" successfully rolled out
student@node-1:~$ kubectl rollout undo deployment app -n kdpd00202
deployment.apps/app rolled back
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 8 of 10 updated replicas are available...
Waiting for deployment "app" rollout to finish: 9 of 10 updated replicas are available...
deployment "app" successfully rolled out
student@node-1:~$
```

질문 # 67

Refer to Exhibit.



Task

You are required to create a pod that requests a certain amount of CPU and memory, so it gets scheduled to a node that has those resources available.

- * Create a pod named nginx-resources in the pod-resources namespace that requests a minimum of 200m CPU and 1Gi memory for its container
- * The pod should use the nginx image
- * The pod-resources namespace has already been created

정답:

설명:

Solution:

Readme > Web Terminal

THE LINUX FOUNDATION

```
student@node-1:~$ kubectl run nginx-resources -n pod-resources --image=nginx --dry-run=client -o
yaml > nginx_resources.yaml
student@node-1:~$ vim nginx
```

Readme > Web Terminal

THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

"nginx_resources.yaml" 16L, 289C

1,1

All

Readme > Web Terminal

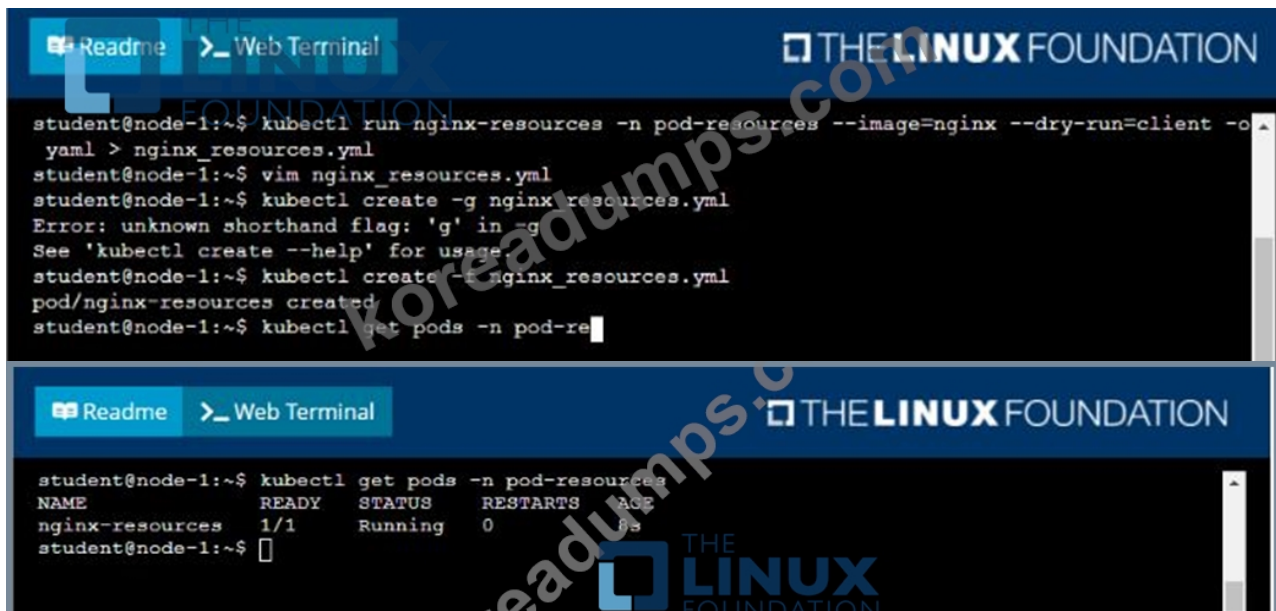
THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources:
      requests:
        cpu: 200m
        memory: "1Gi"
```

-- INSERT --

15,22

All



```
student@node-1:~$ kubectrl run nginx-resources -n pod-resources --image=nginx --dry-run=client -o
yaml > nginx_resources.yaml
student@node-1:~$ vim nginx_resources.yaml
student@node-1:~$ kubectrl create -g nginx_resources.yaml
Error: unknown shorthand flag: 'g' in -g
See 'kubectrl create --help' for usage.
student@node-1:~$ kubectrl create -f nginx_resources.yaml
pod/nginx-resources created
student@node-1:~$ kubectrl get pods -n pod-re

student@node-1:~$ kubectrl get pods -n pod-resources
NAME          READY   STATUS    RESTARTS   AGE
nginx-resources 1/1     Running   0           8s
student@node-1:~$
```

질문 # 68

You are managing a Kubernetes cluster running a highly-available application that uses a custom resource called 'Orders'. The 'orders' resource is created and managed by a custom controller that ensures the order processing workflow runs flawlessly. However, the 'order' resource's validation rules have changed, requiring a new schema to be applied. How can you ensure that the existing 'Order' resources conform to the new schema without disrupting the application's functionality?

정답:

설명:

See the solution below with Step by Step Explanation.

Explanation:

Solution (Step by Step) :

1). Define the New Schema:

- Create a new CustomResourceDefinition (CRD) file with the updated schema for the 'Order' resource.
- Ensure that the CRD's 'spec-validation.openAPIV3Schema' field includes all the new validation rules.

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: orders.example.com
spec:
  group: example.com
  version: v1
  names:
    kind: Order
    plural: orders
  scope: Namespaced
  validation:
    openAPIV3Schema:
      type: object
      properties:
        # Updated validation rules for the 'Order' resource
        ...
```

2. Update the CRD: - Apply the new CRD definition using 'kubectrl apply -f order-crd.yaml'. 3. Create a Webhook for Validation: - Define a webhook in your Kubernetes cluster that will be responsible for validating the 'order' resources against the new schema. - Configure the webhook to be invoked during resource creation and update operations.

```

apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingWebhookConfiguration
metadata:
  name: order-validation-webhook
webhooks:
- name: order-validation
  rules:
  - apiGroups: ["example.com"]
    apiVersions: ["v1"]
    resources: ["orders"]
    operations: ["CREATE", "UPDATE"]
  failurePolicy: Fail
  admissionReviewVersions: ["v1", "v1beta1"]
  clientConfig:
    service:
      name: order-validation-service
      namespace: default
      path: /validate

```

4. Deploy the Validation Service: - Create a deployment for the validation service that implements the logic for validating the 'Order' resources against the new schema. - The service should expose an endpoint that the webhook can communicate with.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: order-validation-service
spec:
  replicas: 1
  selector:
    matchLabels:
      app: order-validation
  template:
    metadata:
      labels:
        app: order-validation
    spec:
      containers:
      - name: order-validation
        image: your-validation-image:latest
        ports:
        - containerPort: 8443

```

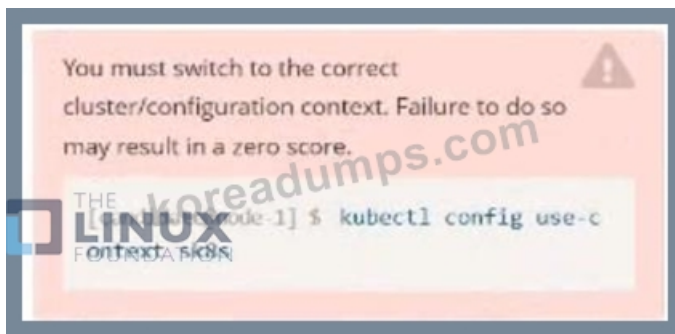
5. Reconcile Existing Resources: - Once the validation webhook and service are deployed, create a temporary job that iterates through all existing 'Order' resources. - The job should validate each resource against the new schema and automatically update any resources that do not comply.

```

apiVersion: batch/v1
kind: Job
metadata:
  name: order-reconciliation-job
spec:
  template:
    spec:
      containers:
      - name: order-reconciliation
        image: your-reconciliation-image:latest
        command: ["bin/sh", "-c", "kubectl get orders --all-namespaces -o jsonpath='{.items[*].metadata.name}' | xargs -n 1 kubectl get -o jsonpath='{.spec}' | jq -r '. | . + { \"metadata\": { \"annotations\": { \"order.example.com/schema-version\": \"2\" } } }' | kubectl apply -f -"]

```

By following these steps, you can ensure that your 'order' resources conform to the new schema without disrupting the application's functionality. The validation webhook prevents new invalid resources from being created, and the reconciliation job ensures that existing resources are updated to meet the new schema requirements. This approach allows for smooth schema evolution and maintains the consistency of your data.



Task:

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

정답:

설명:

See the solution below.

Explanation

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl create deploy expose -n ckad00014 --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml > d
ep.yaml
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: expose
  name: expose
  namespace: ckad00014
spec:
  replicas: 6
  selector:
    matchLabels:
      app: expose
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: expose
    spec:
      containers:
        - image: lfccncf/nginx:1.13.7
          name: nginx
          ports:
            - containerPort: 8001
          env:
            - name: NGINX_PORT
              value: "8001"
```

Text Description automatically generated

myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

BONUS!!! KoreaDumps CKAD 시험 문제집 전체 버전을 무료로 다운로드하세요: https://drive.google.com/open?id=1XGJ63-ghf0qH8nE7YpIyO6ERXp9qotz_