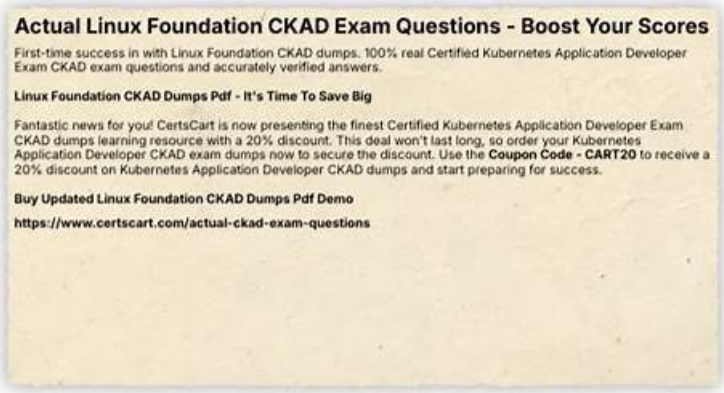
CKAD Actual Exam & Valid CKAD Practice Questions



What's more, part of that PassCollection CKAD dumps now are free: https://drive.google.com/open?id=1fJN4c22tFkF-Rfwy4kd87bIBJLggUcRz

The aim that we try our best to develop the CKAD exam software is to save you money and time, and offer the effective help for you to pass the exam during your preparation for CKAD exam. Our software has help more CKAD exam candidates get the exam certification, but no matter how high our pass rate is, we still guarantee that if you fail the CKAD Exam, we will full refund the money you purchased the CKAD exam software, which makes you be more rest assured to purchase our product.

Linux Foundation Certified Kubernetes Application Developer (CKAD) exam is a certification program designed to evaluate and certify the skills and knowledge of developers in using Kubernetes to develop cloud-native applications. The CKAD Certification is recognized worldwide as a benchmark for Kubernetes application development expertise, and it validates the ability of developers to create and deploy cloud-native applications using Kubernetes.

>> CKAD Actual Exam <<

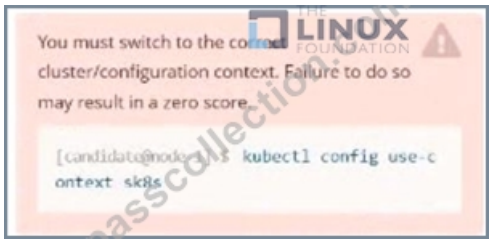
Valid CKAD Practice Questions & CKAD Exam Overview

You many attend many certificate exams but you unfortunately always fail in or the certificates you get can't play the rules you wants and help you a lot. So what certificate exam should you attend and what method should you use to let the certificate play its due rule? You should choose the test Linux Foundation certification and buys our CKAD learning file to solve the problem. Passing the test CKAD certification can help you increase your wage and be promoted easily and buying our CKAD prep guide dump can help you pass the test smoothly. Our CKAD Certification material is closely linked with the test and the popular trend among the industries and provides all the information about the test. The answers and questions seize the vital points and are verified by the industry experts. Diversified functions can help you get an all-around preparation for the test. Our online customer service replies the clients' questions about our CKAD certification material at any time.

The CKAD certification is a valuable credential for developers who are looking to advance their career in the field of Kubernetes and containerization. Linux Foundation Certified Kubernetes Application Developer Exam certification demonstrates that the candidate has the skills and knowledge required to design, build, and deploy Kubernetes-based applications, and can effectively use Kubernetes to manage containerized applications. The CKAD Certification is recognized by industry leaders and provides a competitive edge to the certified professional in the job market.

Linux Foundation Certified Kubernetes Application Developer Exam Sample Questions (Q80-Q85):

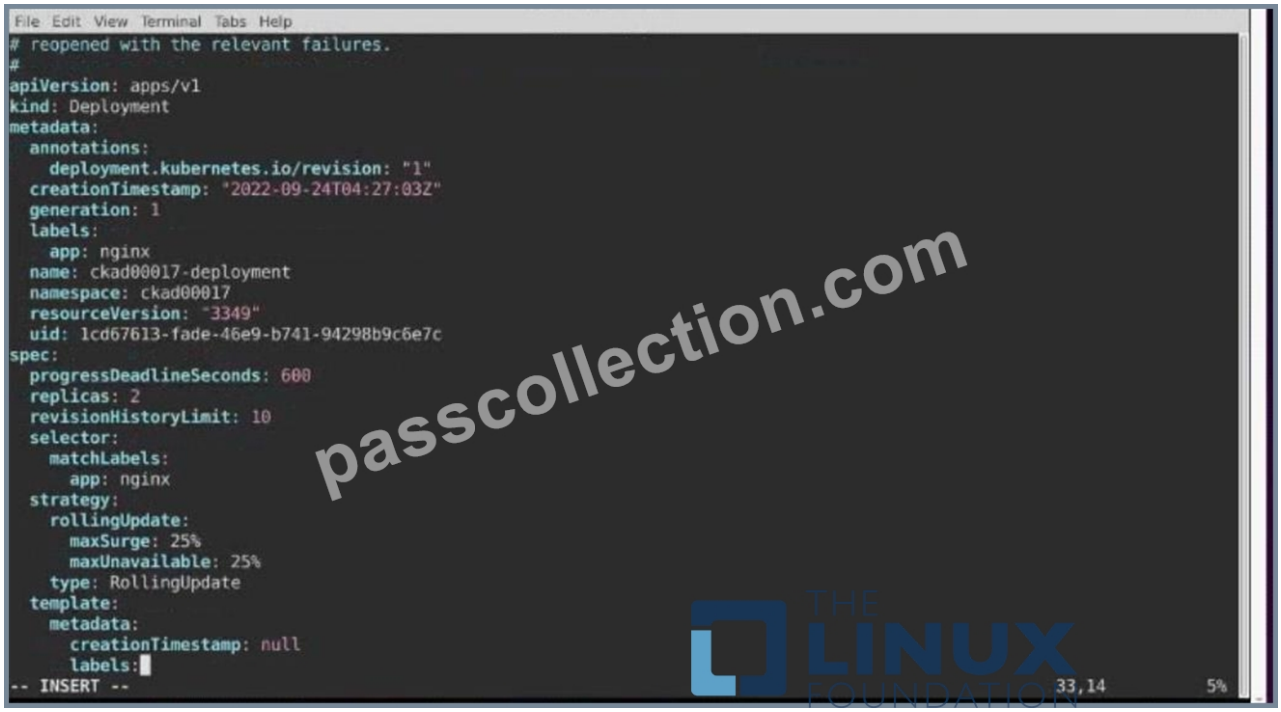
NEW QUESTION # 80
Context



Task:
1) First update the Deployment cka00017-deployment in the ckad00017 namespace:
To run 2 replicas of the pod
Add the following label on the pod:
Role userUI
2) Next, Create a NodePort Service named cherry in the ckad00017 nmespace exposing the ckad00017-deployment Deployment on TCP port 8888

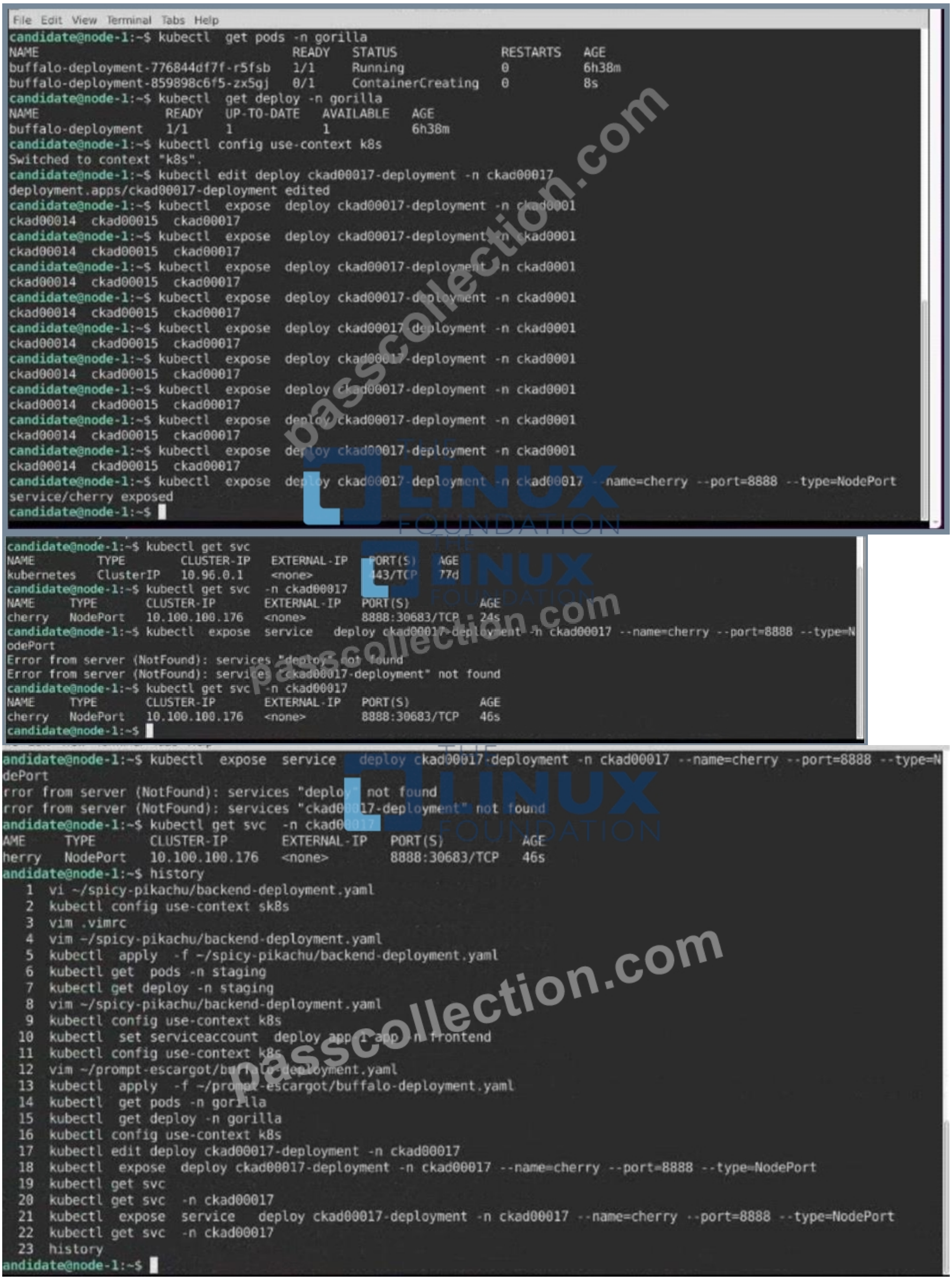
Answer:

Explanation:
Solution:



```
File  Edit  View  Terminal  Tabs  Help
  name: ckad00017-deployment
  namespace: ckad00017
  resourceVersion: "3349"
  uid: 1cd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
        role: userUI
    spec:
      containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
        - containerPort: 80
          protocol: TCP
        resources: {}
-- INSERT --                                                    35,21         33%
```

```
File  Edit  View  Terminal  Tabs  Help
backend-deployment-59d449b99d-h2zjq   0/1     Running   0          9s
backend-deployment-78976f74f5-b8c85   1/1     Running   0          6h40m
backend-deployment-78976f74f5-fllfsj  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:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$
```



NEW QUESTION # 81

You have a Kubernetes deployment named 'wordpress-deployment' running multiple instances of a WordPress application. You want to implement a rolling update strategy with a 'maxSurge' of 1 and 'maxi-Jnavailable' of O. Additionally, you need to ensure that the update process is automatically triggered when a new image is pushed to the Docker Hub repository 'wordpress-image:latests. Implement a Kustomization file to achieve this.

Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a 'kustomization.yamr file in your desired directory.

```
resources:
- deployment.yaml
patchesStrategicMerge:
- patch.yaml
```

2. Create a 'deployment-yamr file (or use an existing one) with the following structure.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: wordpress
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
      - name: wordpress
        image: wordpress-image:latest
        imagePullPolicy: Always
```

3. Create a 'patch.yamr file with the following content to configure rolling update and automatic updates:

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
spec:
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
```

4. Apply the Kustomization: bash kubectl apply -k - The 'kustomization.yaml file defines the resources (the 'deployment.yamr file) and the patches to apply. - The 'deployment.yamr file contains the base configuration for the deployment. - The 'patch.yamr file applies a strategic merge patch to the deployment, configuring rolling updates and automatic updates triggered by new images. - The 'maxSurgew and 'maxunavailable' settings in the 'patch.yamr define the maximum number ot pods that can be added or removed during the update process. - The 'imagePullPolicy: AlwayS ensures that the new image is pulled from Docker Hub even if it exists in the pod's local cache, triggering the update.

NEW QUESTION # 82
You have a microservice application that consists of two components: a web server (using Nginx) and a database (using PostgreSQL). The web server needs to access the database through a local connection, but due to network security restrictions, the web server cannot connect to the database directly. Describe how you can utilize a sidecar container to resolve this issue and ensure the database connection is secure.

Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create a Sidecar Container:
- Define a new container in your Deployment's 'spec-template-spec-containers' array, alongside the existing Nginx container. This

new container will house the necessary tools for facilitating a secure database connection.
- Name this container appropriately, for example, 'database-proxy'
- Choose an image that contains the required software for database connection, such as 'postgres' or 'postgresqr
- Use a sidecar pattern in the Deployment YAML file. You can specify the sidecar in the container array in the Pod specification:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
      - name: database-proxy
        image: postgres:latest
        command: ["/bin/sh", "-c", "pg_ctl -D /var/lib/postgresql/data -l logfile -w"]
        volumeMounts:
        - name: postgres-data
          mountPath: /var/lib/postgresql/data
      volumes:
      - name: postgres-data
        persistentVolumeClaim:
          claimName: postgres-pvc
```


2. Database Connection Configuration: - Configure the sidecar container to connect to the database. - Establish a connection using the database user credentials and connection string. - If you use a secure connection, ensure that the certificates and private keys are accessible to the sidecar container. 3. Communication Between Containers: - Configure your web server container to communicate with the sidecar container. - Use environment variables to specify the hostname and port of the sidecar container, enabling the web server to connect to the database proxy within the pod. 4. Volume Sharing: - Optionally, share a volume between the web server and the sidecar container to facilitate shared data access, such as database configuration files. 5. Deploy the Deployment: - Apply the updated Deployment YAML file to your Kubernetes cluster using 'kubectl apply -f my-app.yaml' 6. Test the Application: - Access your web server application and confirm that it successfully connects to the database through the sidecar container.

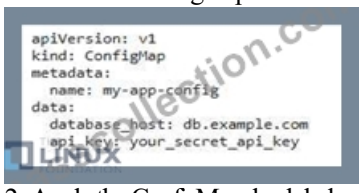
NEW QUESTION # 83
You are building a web application that requires environment-specific configurations, such as database connection details and API keys. You want to use ConfigMaps to manage these configurations in a secure and efficient way You have the following environment variables defined in your deployment YAML:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app
        image: my-app:v1
        env:
        - name: DATABASE_HOST
          valueFrom:
            configMapKeyRef:
              name: my-app-config
              key: database_host
        - name: API_KEY
          valueFrom:
            configMapKeyRef:
              name: my-app-config
              key: api_key
```

Create a ConfigMap named 'my-app-config' containing the following data: - 'database host: 'db.example.com' - 'api_key':

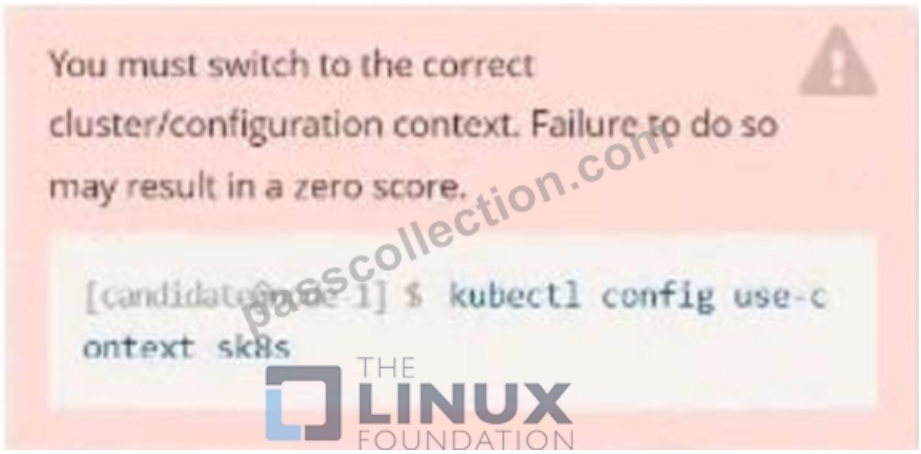
Answer:

Explanation:
See the solution below with Step by Step Explanation.
Explanation:
Solution (Step by Step) :
1. Create the ConfigMap:

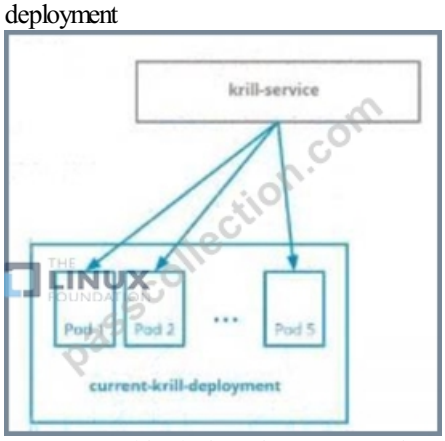


2. Apply the ConfigMap: bash kubectl apply -f my-app-config.yaml 3. Verify the ConfigMap: bash kubectl get configmap my-app-config -o yaml This command will display the created ConfigMap and its contents. 4. Deploy the Deployment: bash kubectl apply -f deploymentyaml The deployment Will now use the values from the ConfigMap to populate the environment variables within the containers. 5. Check the Pods: bash kubectl get pods -l app=my-app -o wide 6. Confirm Environment Variables: bash kubectl exec -it bash -c 'env' Replace with the name of one of the pods. This command will display the environment variables set within the container, including 'DATABASE HOST and 'API KEY". Note: You should replace with your actual API key in the ConfigMap. This ensures that sensitive information is stored in a separate configuration file and not directly in the deployment YAML file.

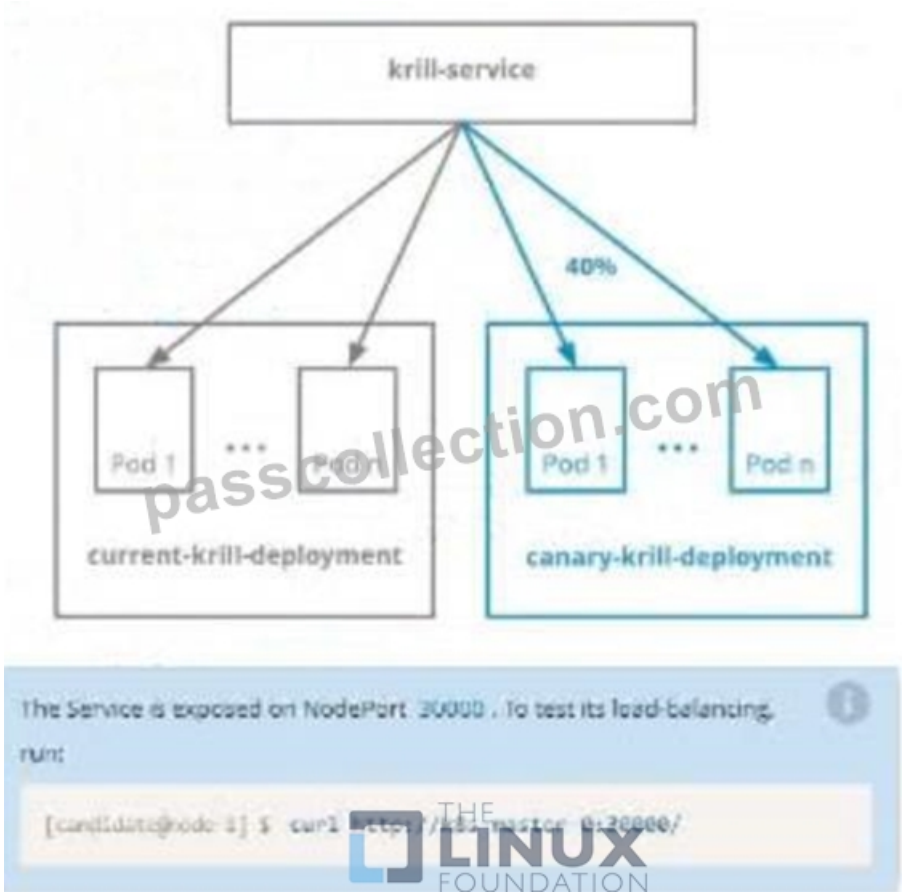
NEW QUESTION # 84



Context
You are asked to prepare a Canary deployment for testing a new application release.
Task:
A Service named krill-Service in the goshark namespace points to 5 pod created by the Deployment named current-krill-deployment



1) Create an identical Deployment named canary-kill-deployment, in the same namespace.
2) Modify the Deployment so that:
-A maximum number of 10 pods run in the goshawk namespace.
-40% of the krill-service 's traffic goes to the canary-krill-deployment pod(s)



Answer:

Explanation:
See the solution below.
Explanation
Solution:

```
candidate@node-1:~/humane-stork$ kubectl  scale  deploy canary-krill-deployment --replicas 4 -n goshawk
deployment.apps/canary-krill-deployment scaled
candidate@node-1:~/humane-stork$ kubectl get deploy -n goshawk
NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
canary-krill-deployment     4/4     4            4           46s
current-krill-deployment    5/5     5            5           7h22m
candidate@node-1:~/humane-stork$ wget https://k8s.io/examples/
```

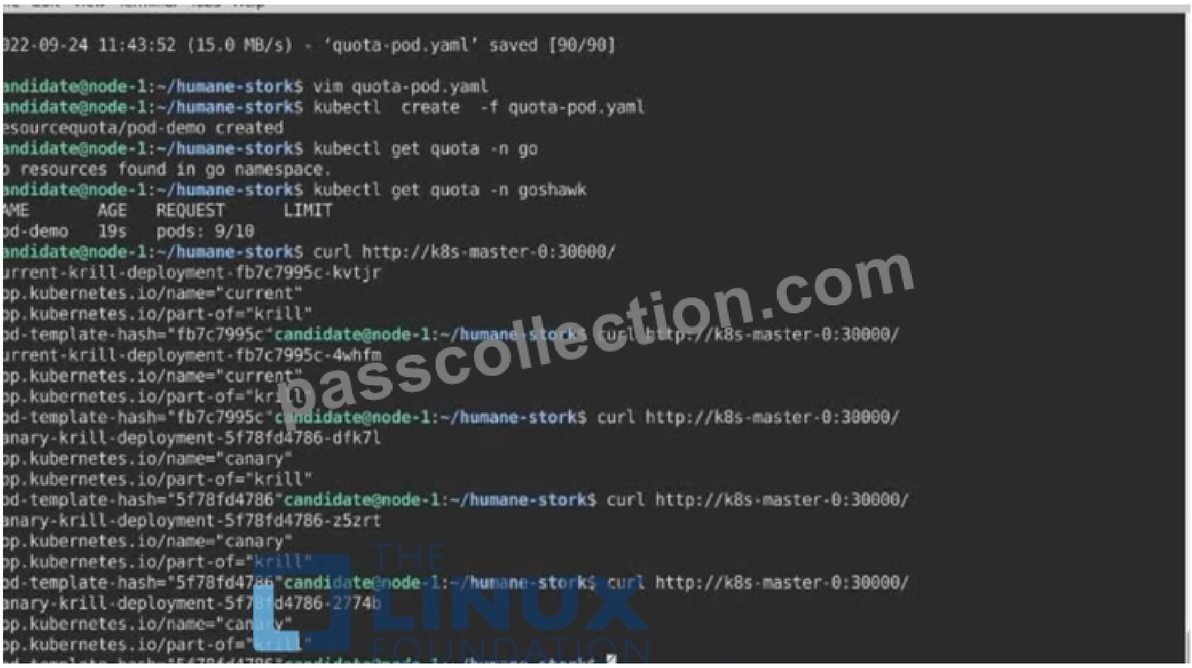
Text Description automatically generated

```
ndidate@node-1:~/humane-stork$ wget https://k8s.io/examples/admin/resource/quota-pod.yaml
2022-09-24 11:43:51--  https://k8s.io/examples/admin/resource/quota-pod.yaml
solving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::
nnecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.
TP request sent, awaiting response... 301 Moved Permanently
cation: https://kubernetes.io/examples/admin/resource/quota-pod.yaml [following]
2022-09-24 11:43:52--  https://kubernetes.io/examples/admin/resource/quota-pod.yaml
solving kubernetes.io (kubernetes.io)... 147.75.40.148
nnecting to kubernetes.io (kubernetes.io)|147.75.40.148|:443... connected.
TP request sent, awaiting response... 200 OK
ngth: 90 [application/x-yaml]
ving to: 'quota-pod.yaml'

ota-pod.yaml                100%[=========================================>]      90  --.-KB/s    in 0s

22-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

ndidate@node-1:~/humane-stork$ vim quota-pod.yaml
```



NEW QUESTION # 85

......

Valid CKAD Practice Questions: https://www.passcollection.com/CKAD_real-exams.html

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, Disposable vapes

BTW, DOWNLOAD part of PassCollection CKAD dumps from Cloud Storage: https://drive.google.com/open?id=1fJN4c22tFkF-Rfwy4kd87bIBJLggUcRz