

Exam Questions CKAD

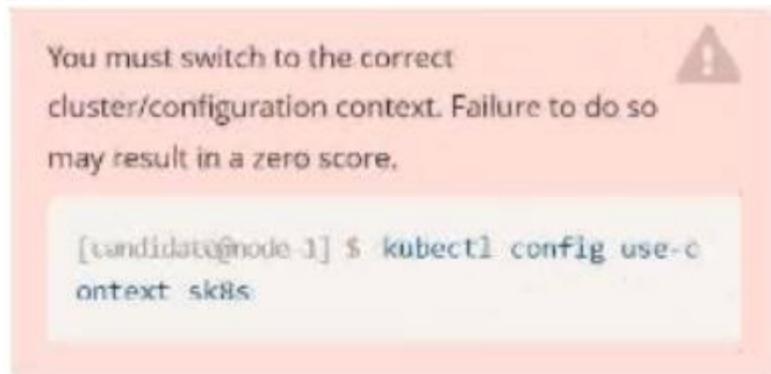
Certified Kubernetes Application Developer (CKAD) Program

<https://www.2passeasy.com/dumps/CKAD/>

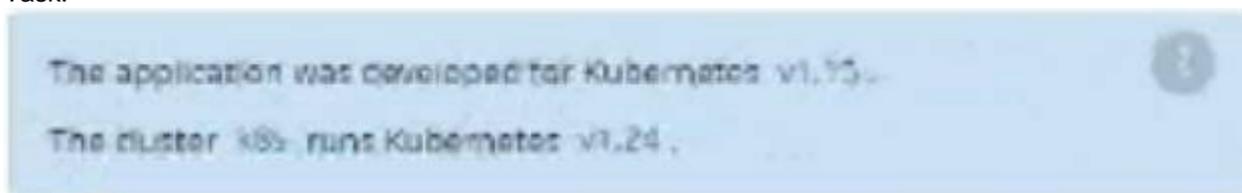


NEW QUESTION 1

Exhibit:



Task:



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
apiVersion: apps/v1
kind: Deployment
metadata:
  name: www-deployment
  namespace: cobra
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: "nginx:stable"
          ports:
            - containerPort: 80
          volumeMounts:
            - mountPath: /var/log/nginx
              name: logs
          env:
            - name: NGINX_ENTRYPOINT_QUIET_LOGS
              value: "1"
      volumes:
        - name: logs
          emptyDir: {}
```

Text Description automatically generated

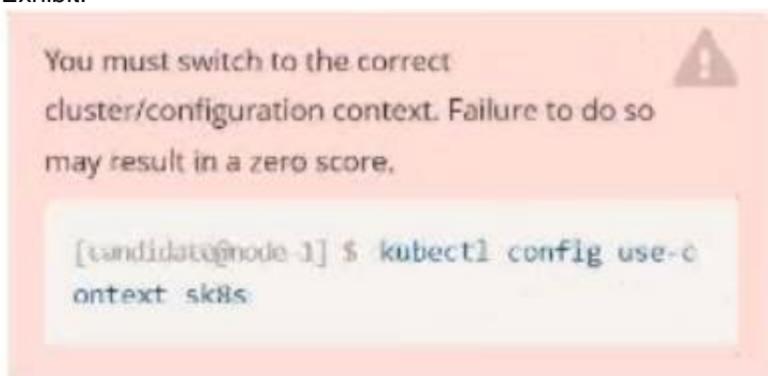
```

File Edit View Terminal Tabs Help
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:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ kubectl apply -f ~/credible-mite/www.yaml
deployment.apps/www-deployment created
candidate@node-1:~$ kubectl get pods -n cobra
NAME                                READY   STATUS              RESTARTS   AGE
www-deployment-d899c6b49-d6ccg      1/1    Running             0           6s
www-deployment-d899c6b49-f796l      0/1    ContainerCreating   0           6s
www-deployment-d899c6b49-ztfcw      0/1    ContainerCreating   0           6s
candidate@node-1:~$ kubectl get deploy -n cobra
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
www-deployment  3/3     3            3           11s
candidate@node-1:~$ kubectl get pods -n cobra
NAME                                READY   STATUS              RESTARTS   AGE
www-deployment-d899c6b49-d6ccg      1/1    Running             0           14s
www-deployment-d899c6b49-f796l      1/1    Running             0           14s
www-deployment-d899c6b49-ztfcw      1/1    Running             0           14s
candidate@node-1:~$

```

NEW QUESTION 2

Exhibit:



Task:

- 1- Update the Propertunel scaling configuration of the Deployment web1 in the ckad00015 namespace setting maxSurge to 2 and maxUnavailable to 59
- 2- Update the web1 Deployment to use version tag 1.13.7 for the lflconf/nginx container image.
- 3- Perform a rollback of the web1 Deployment to its previous version

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy web1 -n ckad00015

```

Text Description automatically generated

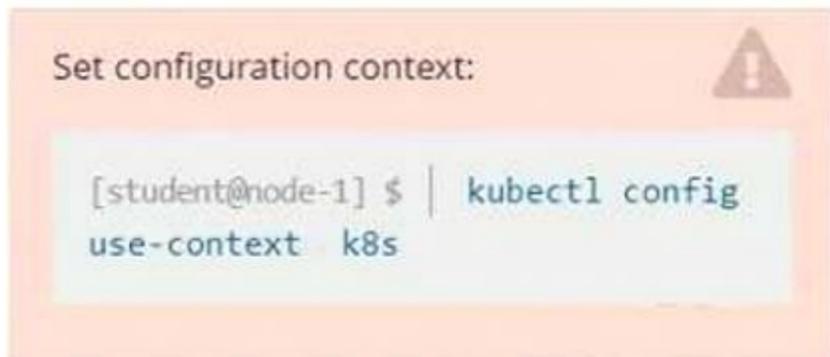
```
File Edit View Terminal Tabs Help
app: nginx
strategy:
  rollingUpdate:
    maxSurge: 2%
    maxUnavailable: 5%
  type: RollingUpdate
template:
  metadata:
    creationTimestamp: null
    labels:
      app: nginx
  spec:
    containers:
      - image: lfccncf/nginx:1.13.7
        imagePullPolicy: IfNotPresent
        name: nginx
        ports:
          - containerPort: 80
            protocol: TCP
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
    dnsPolicy: ClusterFirst
    restartPolicy: Always
    schedulerName: default-scheduler
    securityContext: {}
    terminationGracePeriodSeconds: 30
status:
  availableReplicas: 2
  conditions:
    - lastTransitionTime: "2022-09-24T04:26:41Z"
```

```
File Edit View Terminal Tabs Help
switched to context "k8s".
candidate@node-1:~$ kubectl create secret generic app-secret -n default --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME          TYPE          DATA   AGE
app-secret    Opaque        1       4s
candidate@node-1:~$ kubectl run nginx-secret -n default --image=nginx:stable --dry-run=client -o yaml > sec.yaml
candidate@node-1:~$ vim sec.yaml
candidate@node-1:~$ kubectl create -f sec.yaml
pod/nginx-secret created
candidate@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-secret  1/1     Running   0           7s
candidate@node-1:~$ kubectl config use-context k8s
switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy web1 -n ckad00015
deployment.apps/web1 edited
candidate@node-1:~$ kubectl rollout status deploy web1 -n ckad00015
deployment "web1" successfully rolled out
candidate@node-1:~$ kubectl rollout undo deploy web1 -n ckad00015
deployment.apps/web1 rolled back
candidate@node-1:~$ kubectl rollout history deploy web1 -n ckad00015
deployment.apps/web1
REVISION  CHANGE-CAUSE
2          <none>
3          <none>

candidate@node-1:~$ kubectl get rs -n ckad00015
NAME          DESIRED   CURRENT   READY   AGE
web1-56f98bcb79  0         0         0       63s
web1-85775b6b79  2         2         2       6h53m
candidate@node-1:~$
```

NEW QUESTION 3

Exhibit:



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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
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
```

Readme Web Terminal THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    env:
    - name: COOL_VARIABLE
      valueFrom:
        secretKeyRef:
          name: some-secret
          key: key1

-- INSERT -- 16,20 All
```

```

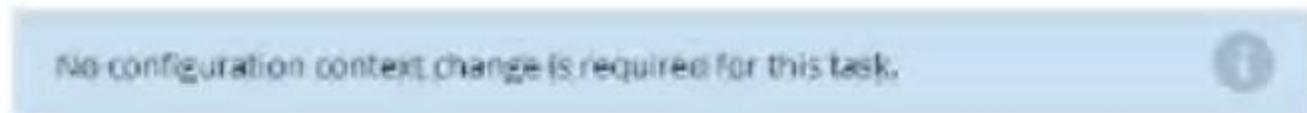
Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS    RESTARTS   AGE
cache     1/1     Running   0           9s
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
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
student@node-1:~$ kubectl create -f nginx_secret.yml
pod/nginx-secret created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS             RESTARTS   AGE
liveness-http 1/1     Running            0           6h38m
nginx-101     1/1     Running            0           6h39m
nginx-secret   0/1     ContainerCreating  0           4s
poller        1/1     Running            0           6h39m
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
liveness-http 1/1     Running   0           6h38m
nginx-101     1/1     Running   0           6h39m
nginx-secret   1/1     Running   0           8s
poller        1/1     Running   0           6h39m
student@node-1:~$

```

NEW QUESTION 4

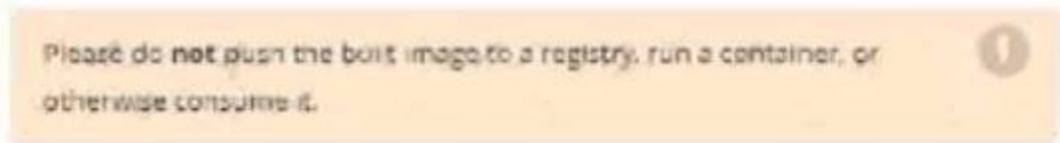
Exhibit:



Task:

A Dockerfile has been prepared at `~/humane-stork/build/Dockerfile`

Multiple image builders and tools have been pre-installed in the base system including: `docker`, `skopeo`, `buildah`, `img`, and `podman`.



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

candidate@node-1:~$ cd humane-stork/build/
candidate@node-1:~/humane-stork/build$ ls -l
total 16
-rw-r--r-- 1 candidate candidate 201 Sep 24 04:21 Dockerfile
-rw-r--r-- 1 candidate candidate 644 Sep 24 04:21 text1.html
-rw-r--r-- 1 candidate candidate 813 Sep 24 04:21 text2.html
-rw-r--r-- 1 candidate candidate 383 Sep 24 04:21 text3.html
candidate@node-1:~/humane-stork/build$ sudo docker build -t macaque:3.0 .
Sending build context to Docker daemon 6.144kB
Step 1/5 : FROM docker.io/lfccncf/nginx:mainline
--> ea335eea17ab
Step 2/5 : ADD text1.html /usr/share/nginx/html/
--> 8967ee9ee5d0
Step 3/5 : ADD text2.html /usr/share/nginx/html/
--> cb0554422f26
Step 4/5 : ADD text3.html /usr/share/nginx/html/
--> 62e879ab821e
Step 5/5 : COPY text2.html /usr/share/nginx/html/index.html
--> 331c8a94372c
Successfully built 331c8a94372c
Successfully tagged macaque:3.0
candidate@node-1:~/humane-stork/build$ sudo docker save macaque:3.0 > ~/humane-stork/macaque-3.0.tar
candidate@node-1:~/humane-stork/build$ cd ..
candidate@node-1:~/humane-stork$ ls -l
total 142532
drwxr-xr-x 2 candidate candidate 4096 Sep 24 04:21 build
-rw-rw-r-- 1 candidate candidate 145948672 Sep 24 11:39 macaque-3.0.tar
candidate@node-1:~/humane-stork$

```

NEW QUESTION 5

Exhibit:



Context

You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount. Task
Please complete the following:

- Create a ConfigMap named another-config containing the key/value pair: key4/value3
- start a pod named nginx-configmap containing a single container using the nginx image, and mount the key you just created into the pod under directory /also/a/path

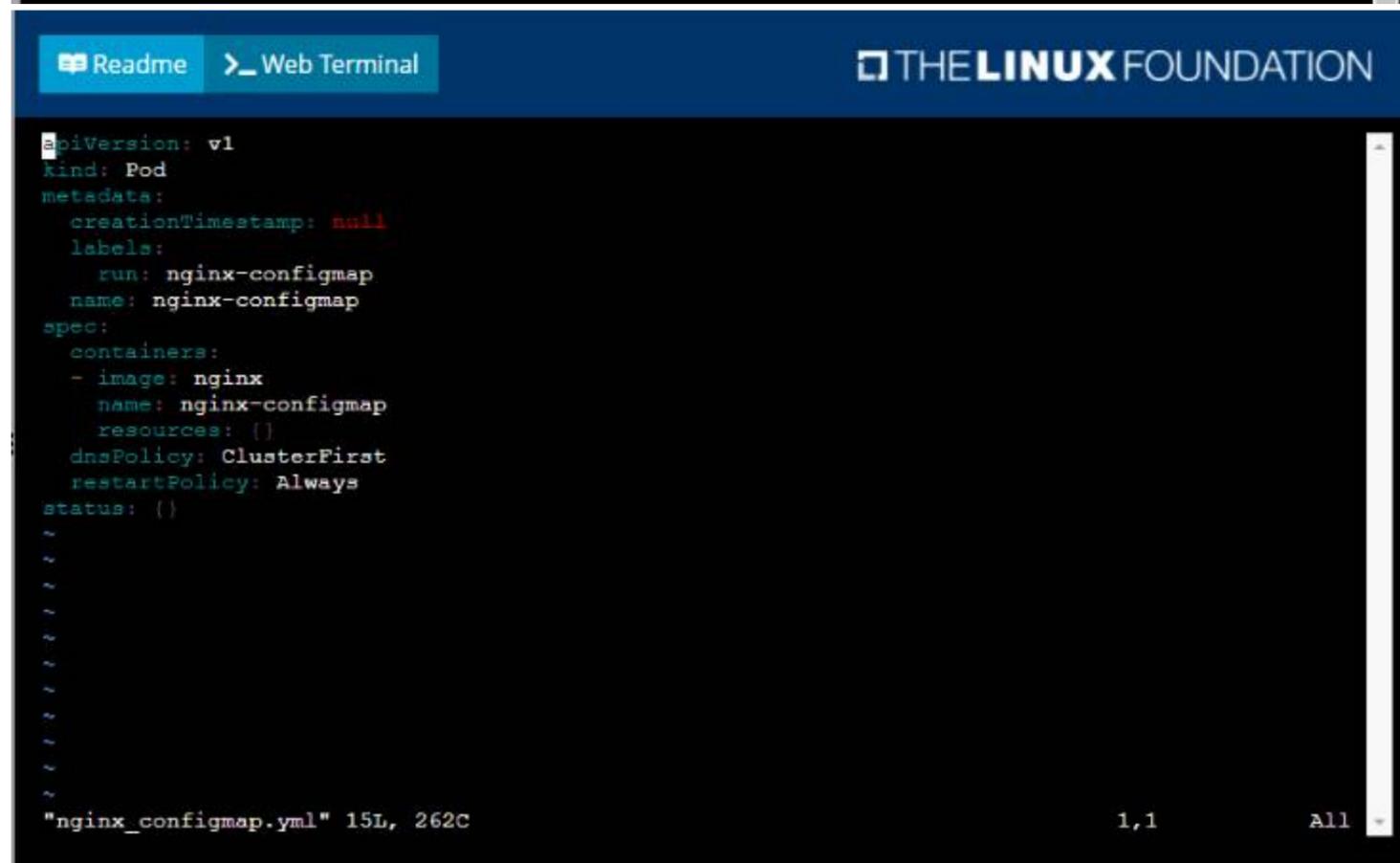
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME          DATA  AGE
another-config 1      5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml ^C
student@node-1:~$ mv nginx_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_co
```

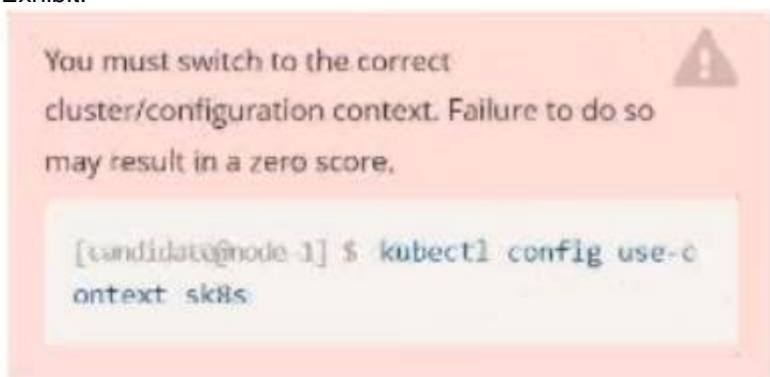


```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-configmap
  name: nginx-configmap
spec:
  containers:
  - image: nginx
    name: nginx-configmap
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

"nginx_configmap.yml" 15L, 262C 1,1 All

NEW QUESTION 6

Exhibit:



Task:
 A pod within the Deployment named buffalo-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"
 The buffalo-deployment 'S manifest can be found at -/prompt/escargot/buffalo-deployment.yaml

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:
 Text Description automatically generated

```

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:~$
    
```

NEW QUESTION 7

Exhibit:



Context
 A user has reported an aopticaouon is unteachable due to a failing livenessProbe . Task
 Perform the following tasks:
 • Find the broken pod and store its name and namespace to /opt/KDOB00401/broken.txt in the format:

```
<namespace>/<pod>
```

The output file has already been created

- Store the associated error events to a file /opt/KDOB00401/error.txt, The output file has already been created. You will need to use the -o wide output specifier with your command
- Fix the issue.



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Create the Pod: `kubectl create`

`-f http://k8s.io/docs/tasks/configure-pod-container/`
`exec-liveness.yaml`

Within 30 seconds, view the Pod events: `kubectl describe pod liveness-exec`

The output indicates that no liveness probes have failed yet:

FirstSeen LastSeen Count From SubobjectPath Type Reason Message

```
-----
24s 24s 1 {default-scheduler } Normal Scheduled Successfully assigned liveness-exec to worker0
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/google_containers/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google_containers/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e
```

After 35 seconds, view the Pod events again: `kubectl describe pod liveness-exec`

At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated.

FirstSeen LastSeen Count From SubobjectPath Type Reason Message

```
-----
37s 37s 1 {default-scheduler } Normal Scheduled Successfully assigned liveness-exec to worker0
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/google_containers/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google_containers/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e
2s 2s 1 {kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can't open '/tmp/healthy': No such file or directory
```

Wait another 30 seconds, and verify that the Container has been restarted: `kubectl get pod liveness-exec`

The output shows that RESTARTS has been incremented:

```
NAME READY STATUS RESTARTS AGE
liveness-exec 1/1 Running 1 m
```

NEW QUESTION 8

Exhibit:

You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate@node-1] $ kubectl config use-c  
ontext k8s
```

Task:

The pod for the Deployment named nosql in the crayfish namespace fails to start because its container runs out of resources. Update the nosql Deployment so that the Pod:

• The nosql Deployment's manifest file can be found at
~/chief-cardinal/nosql.yaml.

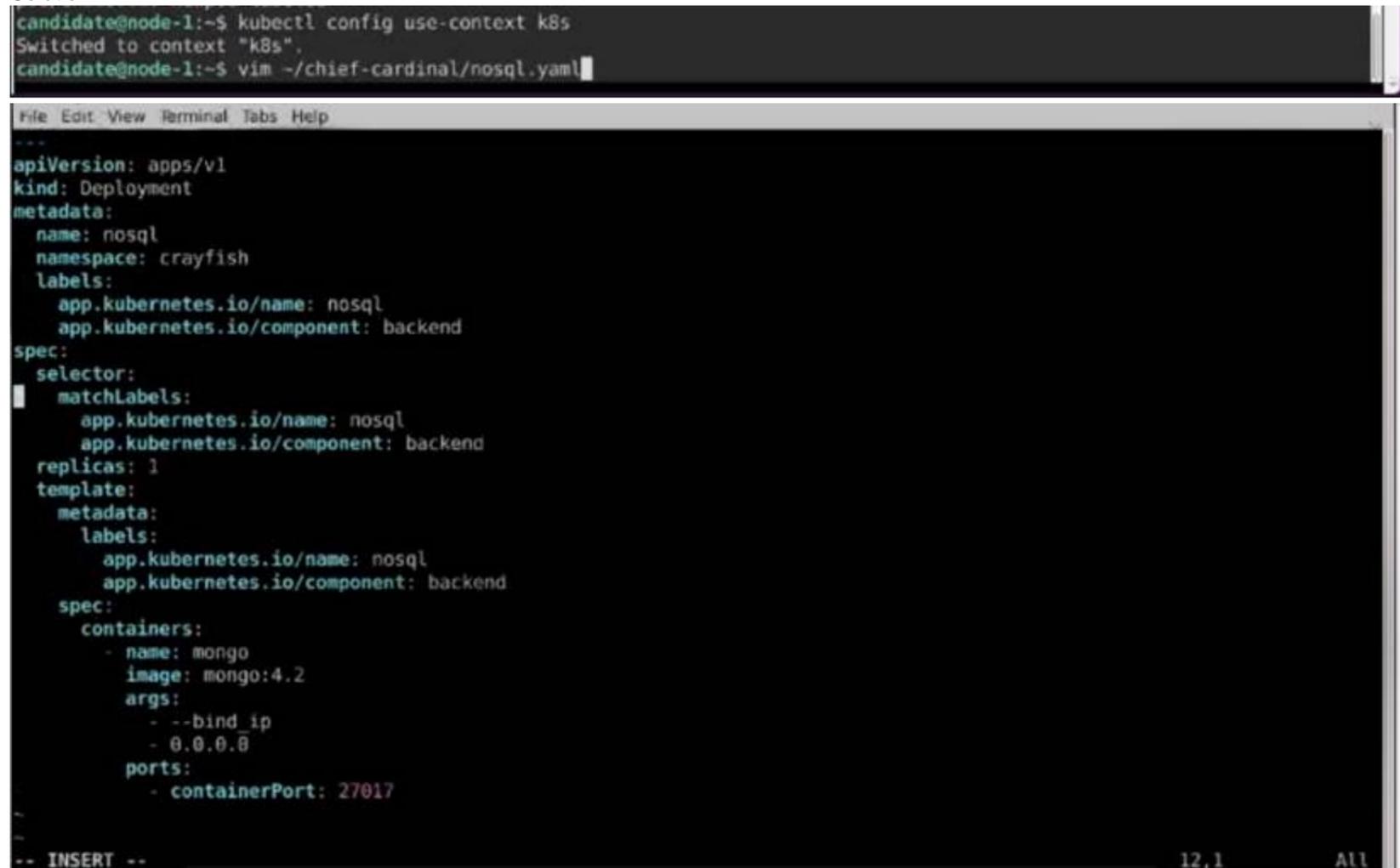
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s  
Switched to context "k8s".  
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
```



```
---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: nosql  
  namespace: crayfish  
  labels:  
    app.kubernetes.io/name: nosql  
    app.kubernetes.io/component: backend  
spec:  
  selector:  
    matchLabels:  
      app.kubernetes.io/name: nosql  
      app.kubernetes.io/component: backend  
  replicas: 1  
  template:  
    metadata:  
      labels:  
        app.kubernetes.io/name: nosql  
        app.kubernetes.io/component: backend  
    spec:  
      containers:  
        - name: mongo  
          image: mongo:4.2  
          args:  
            - --bind ip  
            - 0.0.0.0  
          ports:  
            - containerPort: 27017  
-- INSERT --
```

```
File Edit View Terminal Tabs Help
- name: mongo
  image: mongo:4.2
  args:
    - --bind_ip
    - 0.0.0.0
  ports:
    - containerPort: 27017
  resources:
    requests:
      memory: "160Mi"
    limits:
      memory: "320Mi"

:wq
```

```
File Edit View Terminal Tabs Help
To: <any> (traffic not restricted by destination)
Policy Types: Ingress, Egress

Name:          default-deny
Namespace:     ckad00018
Created on:    2022-09-24 04:27:37 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  PodSelector: <none> (Allowing the specific traffic to all pods in this namespace)
  Allowing ingress traffic:
    <none> (Selected pods are isolated for ingress connectivity)
  Not affecting egress traffic
  Policy Types: Ingress
candidate@node-1:~$ kubectl label pod ckad00018-newpod -n ckad00018 web-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl label pod ckad00018-newpod -n ckad00018 db-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ kubectl apply -f ~/chief-cardinal/nosql.yaml
deployment.apps/nosql configured
candidate@node-1:~$ kubectl get pods -n crayfish
NAME                READY   STATUS    RESTARTS   AGE
nosql-74ccc7d64-lkqlg 1/1     Running   0           3m2s
candidate@node-1:~$ kubectl get deploy -n crayfish
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nosql  1/1     1             1           7h16m
candidate@node-1:~$
```

NEW QUESTION 9

Exhibit:



Context

As a Kubernetes application developer you will often find yourself needing to update a running application. Task Please complete the following:

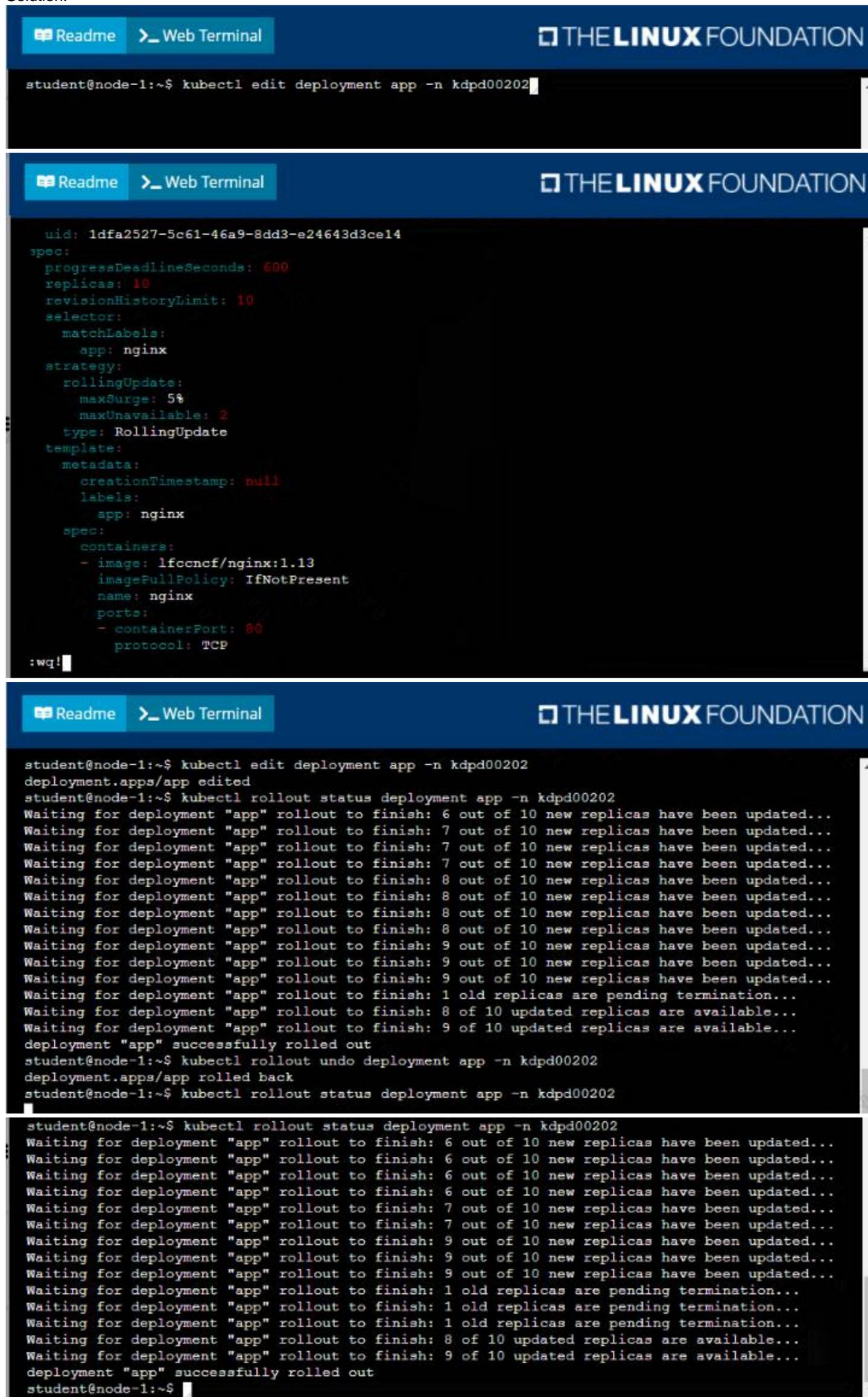
- Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%
- Perform a rolling update of the web1 deployment, changing the lfccncf/ngmx image version to 1.13
- Roll back the app deployment to the previous version

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:



```
student@node-1:~$ kubectl edit deployment app -n kdpd00202
```

```
uid: 1dfa2527-5c61-46a9-8dd3-e24643d3ce14
spec:
  progressDeadlineSeconds: 600
  replicas: 10
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 5%
      maxUnavailable: 2
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: nginx
  spec:
    containers:
    - image: lfccncf/nginx:1.13
      imagePullPolicy: IfNotPresent
      name: nginx
      ports:
      - containerPort: 80
        protocol: TCP
:wq!
```

```
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: 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:~$
```

NEW QUESTION 10

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual CKAD Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the CKAD Product From:

<https://www.2passeasy.com/dumps/CKAD/>

Money Back Guarantee

CKAD Practice Exam Features:

- * CKAD Questions and Answers Updated Frequently
- * CKAD Practice Questions Verified by Expert Senior Certified Staff
- * CKAD Most Realistic Questions that Guarantee you a Pass on Your First Try
- * CKAD Practice Test Questions in Multiple Choice Formats and Updates for 1 Year