



# **Linux-Foundation**

## **Exam Questions CKAD**

Certified Kubernetes Application Developer (CKAD) Program

## About Exambible

*[Your Partner of IT Exam](#)*

## Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

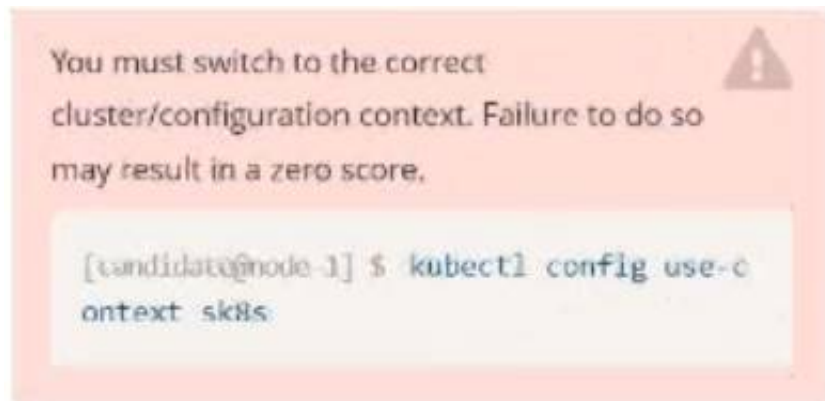
Our guarantee that you will pass the exam.

### \* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

## NEW QUESTION 1

Exhibit:



Task:

Update the Deployment app-1 in the frontend namespace to use the existing ServiceAccount app.

- A. Mastered
- B. Not Mastered

Answer: A

### Explanation:

Solution:

Text Description automatically generated

```
File Edit View Terminal Tabs Help
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

candidate@node-1:~$ vi ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
candidate@node-1:~$ vim .vimrc
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
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:~$
```

## NEW QUESTION 2

Exhibit:



Task:

Key3: value1

Add an environment variable named BEST\_VARIABLE consuming the value of the secret key3.

- 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 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
```

Text Description automatically generated

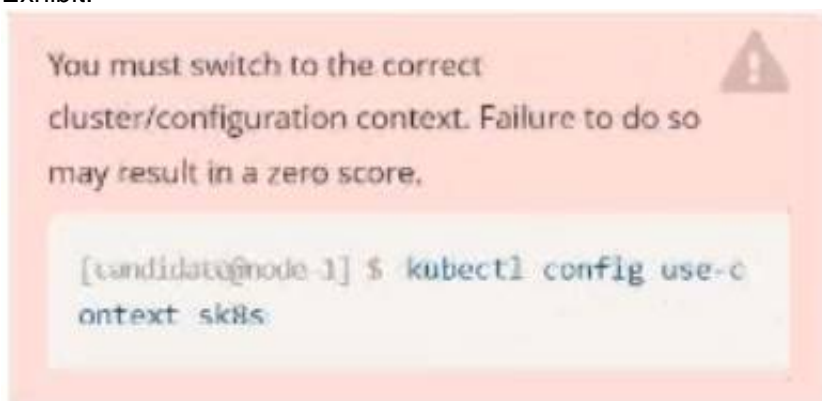
```
File Edit View Terminal Tabs Help
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-secret
  name: nginx-secret
  namespace: default
spec:
  containers:
  - image: nginx:stable
    name: nginx-secret
    env:
    - name: BEST_VARIABLE
      valueFrom:
        secretKeyRef:
          name: app-secret
          key: key3
```

Text Description automatically generated

```
candidate@node-1:~$ kubectl config use-context k8s
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:~$
```

### NEW QUESTION 3

Exhibit:



Task:

Create a Deployment named expose in the existing ckad00014 namespace running 6 replicas of a Pod. Specify a single container using the ifccnrf/nginx: 1.13.7 image

Add an environment variable named NGINX\_PORT with the value 8001 to the container then expose port 8001

- 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 create deploy 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:~$
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

```
File Edit View Terminal Tabs Help
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 > 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:~$ 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-85dd99d4d9-1ld7j             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:~$
```

NEW QUESTION 4

Exhibit:



Context

A web application requires a specific version of redis to be used as a cache. Task

Create a pod with the following characteristics, and leave it running when complete:

- The pod must run in the web namespace. The namespace has already been created
- The name of the pod should be cache
- Use the lfcncf/redis image with the 3.2 tag
- Expose port 6379

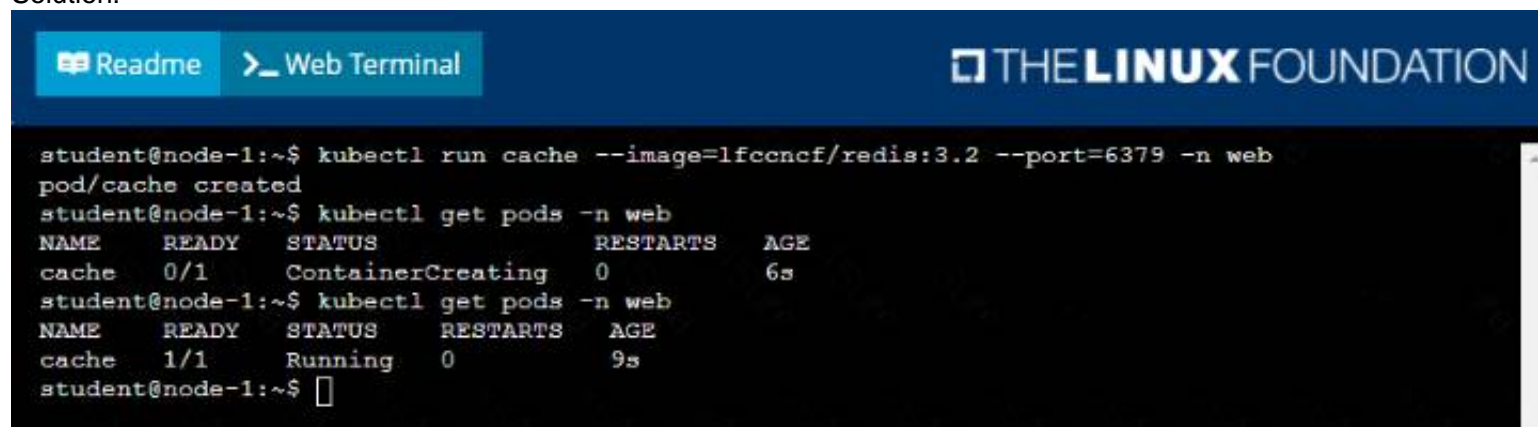
A. Mastered

B. Not Mastered

**Answer:** A

**Explanation:**

Solution:



## NEW QUESTION 5

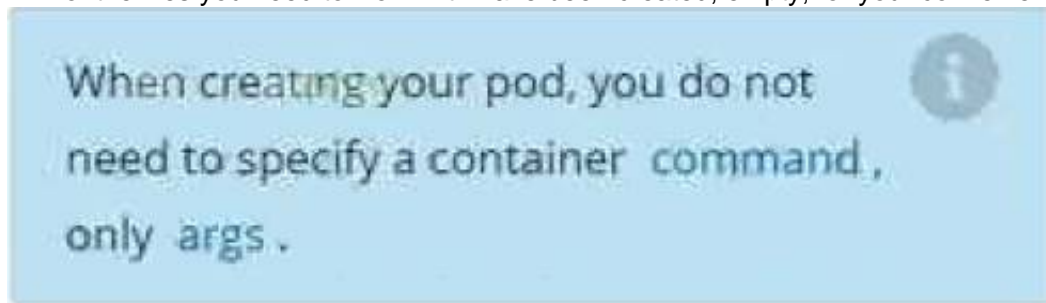
Context

Anytime a team needs to run a container on Kubernetes they will need to define a pod within which to run the container.

Task

Please complete the following:

- Create a YAML formatted pod manifest /opt/KDPD00101/pod1.yml to create a pod named app1 that runs a container named app1cont using image lfcncf/arg-output with these command line arguments: -lines 56 -F
- Create the pod with the kubectl command using the YAML file created in the previous step
- When the pod is running display summary data about the pod in JSON format using the kubectl command and redirect the output to a file named /opt/KDPD00101/out1.json
- All of the files you need to work with have been created, empty, for your convenience



A. Mastered

B. Not Mastered

**Answer:** A

**Explanation:**

Solution:



A screenshot of a terminal window with a dark blue header bar. On the left side of the header are two buttons: "Readme" with a book icon and ">\_ Web Terminal". On the right side of the header is the logo for "THE LINUX FOUNDATION". The main area of the terminal displays a YAML configuration for a Kubernetes Pod. The configuration includes fields for apiVersion, kind, metadata (creationTimestamp, labels), spec (containers, dnsPolicy, restartPolicy), and status. The terminal has a light gray border and shows several tilde (~) characters representing scrollable history. At the bottom of the terminal, there is a status bar with the file path "/opt/KDPD00101/pod1.yml", line/column information "15L, 242C", and search results "3,1 All".

Readme

Web Terminal

THE **LINUX** FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: appl
    name: appl
spec:
  containers:
  - image: lfcncf/arg-output
    name: appl
    args: ["--lines", "56", "-f"]
```

11/30All

```
pod/appl created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS             RESTARTS   AGE
appl          0/1     ContainerCreating   0           5s
counter       1/1     Running             0           4m44s
liveness-http 1/1     Running             0           6h50m
nginx-101     1/1     Running             0           6h51m
nginx-configmap 1/1     Running             0           6m21s
nginx-secret   1/1     Running             0           11m
poller        1/1     Running             0           6h51m
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
appl          1/1     Running   0           26s
counter       1/1     Running   0           5m5s
liveness-http 1/1     Running   0           6h50m
nginx-101     1/1     Running   0           6h51m
nginx-configmap 1/1     Running   0           6m42s
nginx-secret   1/1     Running   0           12m
poller        1/1     Running   0           6h51m
student@node-1:~$ kubectl delete pod appl
pod "appl" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```



```

Readme Web Terminal

nginx-configmap 1/1 Running 0 6m2
nginx-secret 1/1 Running 0 11m
poller 1/1 Running 0 6h5
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 26s
counter 1/1 Running 0 5m5s
liveness-http 1/1 Running 0 6h50m
nginx-101 1/1 Running 0 6h51m
nginx-configmap 1/1 Running 0 6m42s
nginx-secret 1/1 Running 0 12m
poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 20s
counter 1/1 Running 0 6m57s
liveness-http 1/1 Running 0 6h52m
nginx-101 1/1 Running 0 6h53m
nginx-configmap 1/1 Running 0 8m34s
nginx-secret 1/1 Running 0 14m
poller 1/1 Running 0 6h53m
student@node-1:~$ kubectl get pod app1 -o json >

```

```

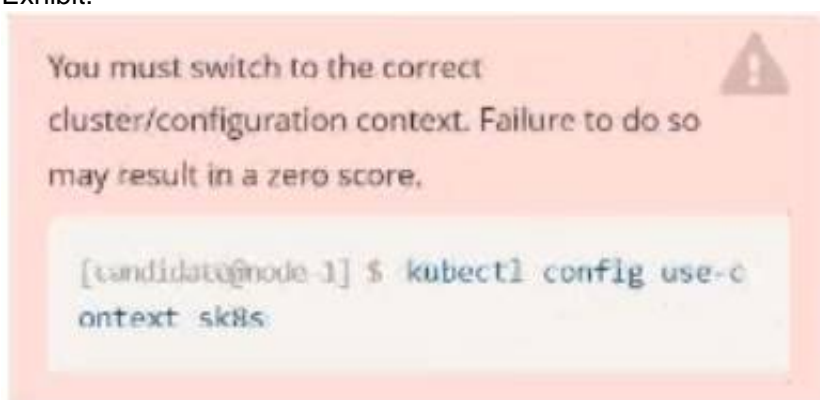
Readme Web Terminal THE LINUX FOUNDATION

poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 26s
counter 1/1 Running 0 5m5s
liveness-http 1/1 Running 0 6h50m
nginx-101 1/1 Running 0 6h51m
nginx-configmap 1/1 Running 0 6m42s
nginx-secret 1/1 Running 0 12m
poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 20s
counter 1/1 Running 0 6m57s
liveness-http 1/1 Running 0 6h52m
nginx-101 1/1 Running 0 6h53m
nginx-configmap 1/1 Running 0 8m34s
nginx-secret 1/1 Running 0 14m
poller 1/1 Running 0 6h53m
student@node-1:~$ kubectl get pod app1 -o json > /opt/KDPD00101/out1.json
student@node-1:~$
student@node-1:~$

```

## NEW QUESTION 6

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 lfcnf/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

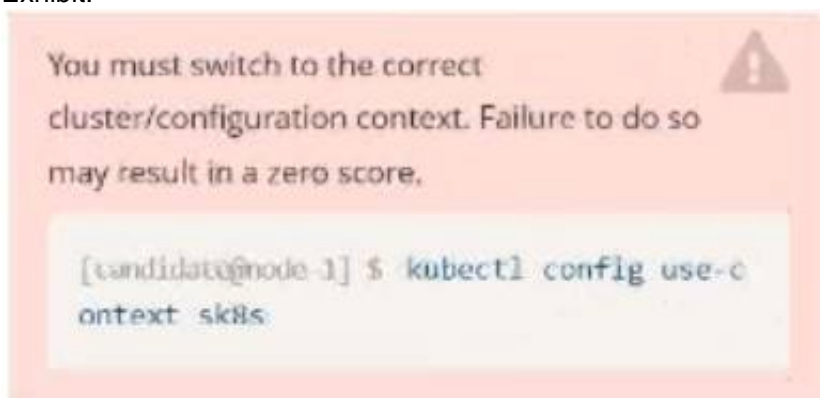
```
File Edit View Terminal Tabs Help
app: nginx
strategy:
  rollingUpdate:
    maxSurge: 2%
    maxUnavailable: 5%
    type: RollingUpdate
template:
  metadata:
    creationTimestamp: null
    labels:
      app: nginx
  spec:
    containers:
      - image: lfccncf/nginx:1.13.7
        imagePullPolicy: 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 7

Exhibit:



Task:

- > To run 2 replicas of the pod
- > Add the following label on the pod:

Role userUI

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Text Description automatically generated

```
File Edit View Terminal Tabs Help
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2022-09-24T04:27:03Z"
  generation: 1
  labels:
    app: nginx
  name: ckad00017-deployment
  namespace: ckad00017
  resourceVersion: "3349"
  uid: lcd67613-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:
-- INSERT --
33,14 5%
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
name: ckad00017-deployment
namespace: ckad00017
resourceVersion: "3349"
uid: lcd67613-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%
```

Text Description automatically generated



```
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-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:~$ 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:~$

File Edit View Terminal Tabs Help
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:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
service/cherry exposed
candidate@node-1:~$

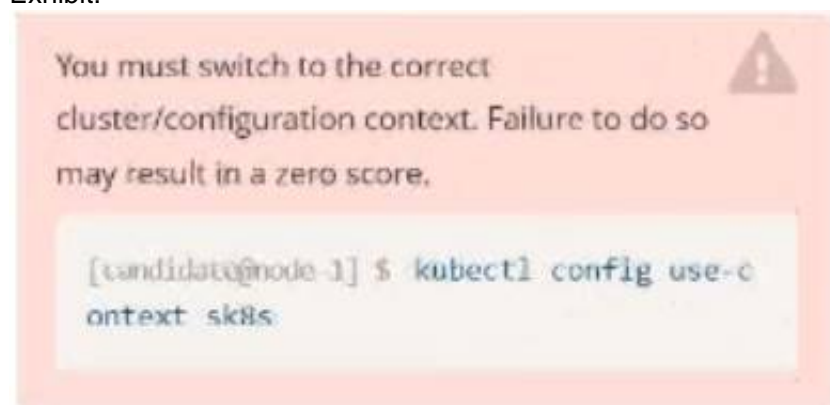
candidate@node-1:~$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 77d
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
cherry NodePort 10.100.100.176 <none> 8888:30683/TCP 24s
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
cherry NodePort 10.100.100.176 <none> 8888:30683/TCP 46s
candidate@node-1:~$
```



```
File Edit View Terminal Tabs Help
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
cherry    NodePort  10.100.100.176 <none>        8888:30683/TCP   46s
candidate@node-1:~$ history
 1 vi ~/spicy-pikachu/backend-deployment.yaml
 2 kubectl config use-context sk8s
 3 vim .vimrc
 4 vim ~/spicy-pikachu/backend-deployment.yaml
 5 kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
 6 kubectl get pods -n staging
 7 kubectl get deploy -n staging
 8 vim ~/spicy-pikachu/backend-deployment.yaml
 9 kubectl config use-context k8s
10 kubectl set serviceaccount deploy app-1 app -n frontend
11 kubectl config use-context k8s
12 vim ~/prompt-escargot/buffalo-deployment.yaml
13 kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
14 kubectl get pods -n gorilla
15 kubectl get deploy -n gorilla
16 kubectl config use-context k8s
17 kubectl edit deploy ckad00017-deployment -n ckad00017
18 kubectl expose deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
19 kubectl get svc
20 kubectl get svc -n ckad00017
21 kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
22 kubectl get svc -n ckad00017
23 history
candidate@node-1:~$
```

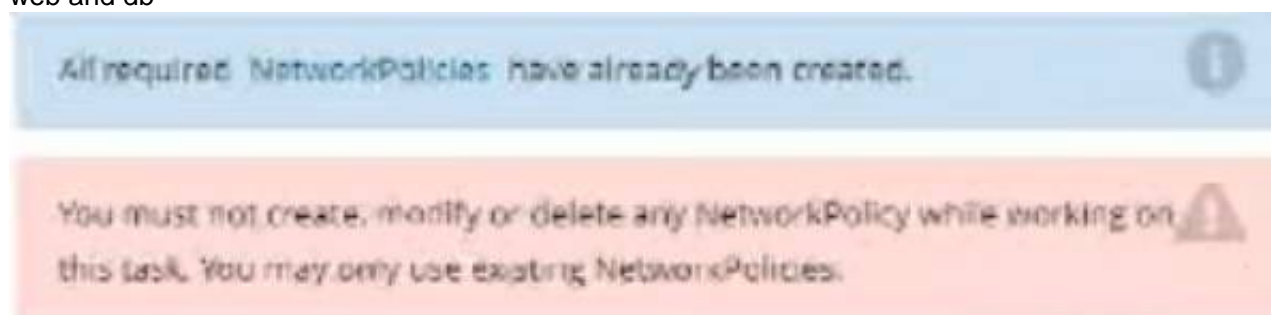
## NEW QUESTION 8

Exhibit:



Task:

Update the Pod ckad00018-newpod in the ckad00018 namespace to use a NetworkPolicy allowing the Pod to send and receive traffic only to and from the pods web and db



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
candidate@node-1:~$ kubectl config use-context nk8s
Switched to context "nk8s".
candidate@node-1:~$ kubectl describe netpol -n ckad00018
```

```

File Edit View Terminal Tabs Help
Name:      all-access
Namespace:  ckad00018
Created on: 2022-09-24 04:27:37 +0000 UTC
Labels:    <none>
Annotations: <none>
Spec:
  PodSelector:  all-access=true
  Allowing ingress traffic:
    To Port: <any> (traffic allowed to all ports)
    From: <any> (traffic not restricted by source)
  Allowing egress traffic:
    To Port: <any> (traffic allowed to all ports)
    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:~$

```

## NEW QUESTION 9

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_conf
igmap.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

```



Readme
Web Terminal

THE LINUX FOUNDATION

```

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

```

Readme
Web Terminal

THE LINUX FOUNDATION

```

apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-configmap
    name: nginx-configmap
spec:
  containers:
  - image: nginx
    name: nginx-configmap
    volumeMounts:
    - name: myvol
      mountPath: /also/a/path
  volumes:
  - name: myvol
    configMap:
      name: another-config
~
~
~
~
~
~
~
~
~
~
13,6
All

```

```

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_conf
igmap.yml
student@node-1:~$ vim nginx_configmap.yml ^C
student@node-1:~$ mv nginx_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$

```

```


student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > nginx_conf
igmap.yml
student@node-1:~$ vim nginx_configmap.yml ^C
student@node-1:~$ mv nginx_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ kubectl create f nginx_configmap.yml
Error: must specify one of -f and -k

error: unknown command "f nginx_configmap.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_configmap.yml
error: error validating "nginx_configmap.yml": error validating data: ValidationError(Pod.spec.c
ontainers[1]): unknown field "mountPath" in io.k8s.api.core.v1.Container; if you choose to ignor
e these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_configmap.yml

```



Readme
Web Terminal



```

student@node-1:~$ kubectl create f nginx_configmap.yml
Error: must specify one of -f and -k

error: unknown command "f nginx_configmap.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_configmap.yml
error: error validating "nginx_configmap.yml": error validating data: ValidationError(Pod.spec.con
tainers[1]): unknown field "mountPath" in io.k8s.api.core.v1.Container; if you choose to ignor
e these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ kubectl create -f nginx_configmap.yml
pod/nginx-configmap created
student@node-1:~$ kubectl get pods
NAME                READY   STATUS              RESTARTS   AGE
liveness-http       1/1     Running             0           6h44m
nginx-101            1/1     Running             0           6h45m
nginx-configmap      0/1     ContainerCreating   0           5s
nginx-secret         1/1     Running             0           5m39s
poller              1/1     Running             0           6h44m
student@node-1:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
liveness-http       1/1     Running   0           6h44m
nginx-101            1/1     Running   0           6h45m
nginx-configmap      1/1     Running   0           8s
nginx-secret         1/1     Running   0           5m42s
poller              1/1     Running   0           6h45m
student@node-1:~$ l

```

## NEW QUESTION 10

Exhibit:



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.


- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Readme
Web Terminal



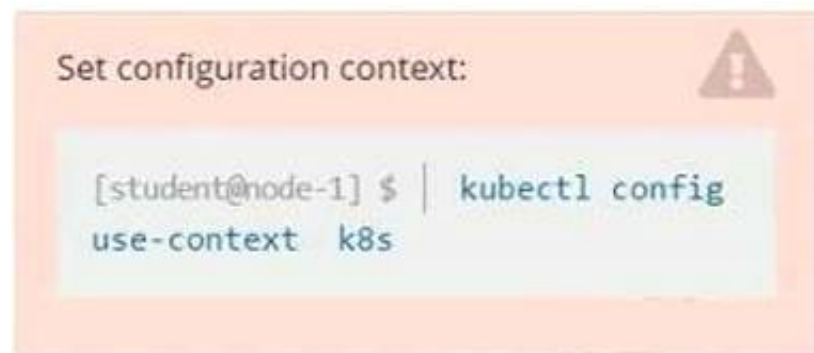
```

student@node-1:~$ kubectl get serviceaccount -n production
NAME          SECRETS  AGE
default        1        6h46m
restrictedservice  1        6h46m
student@node-1:~$ kubectl get deployment -n production
NAME    READY  UP-TO-DATE  AVAILABLE  AGE
app-a   3/3    3           3           6h46m
student@node-1:~$ kubectl set serviceaccount deployment app-a restrictedservice -n production
deployment.apps/app-a serviceaccount updated
student@node-1:~$

```

## NEW QUESTION 10

Exhibit:



Context

A container within the poller pod is hard-coded to connect the nginxsvc service on port 90. As this port changes to 5050 an additional container needs to be added to the poller pod which adapts the container to connect to this new port. This should be realized as an ambassador container within the pod.

Task

- Update the nginxsvc service to serve on port 5050.
- Add an HAproxy container named haproxy bound to port 90 to the poller pod and deploy the enhanced pod. Use the image haproxy and inject the configuration located at /opt/KDMC00101/haproxy.cfg, with a ConfigMap named haproxy-config, mounted into the container so that haproxy.cfg is available at /usr/local/etc/haproxy/haproxy.cfg. Ensure that you update the args of the poller container to connect to localhost instead of nginxsvc so that the connection is correctly proxied to the new service endpoint. You must not modify the port of the endpoint in poller's args . The spec file used to create the initial poller pod is available in /opt/KDMC00101/poller.yaml

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

apiVersion: apps/v1 kind: Deployment metadata:

name: my-nginx spec:

selector:

matchLabels: run: my-nginx replicas: 2 template: metadata: labels:

run: my-nginx spec: containers:

- name: my-nginx image: nginx ports:

- containerPort: 90

This makes it accessible from any node in your cluster. Check the nodes the Pod is running on: kubectl apply -f ./run-my-nginx.yaml

kubectl get pods -l run=my-nginx -o wide

NAME READY STATUS RESTARTS AGE IP NODE

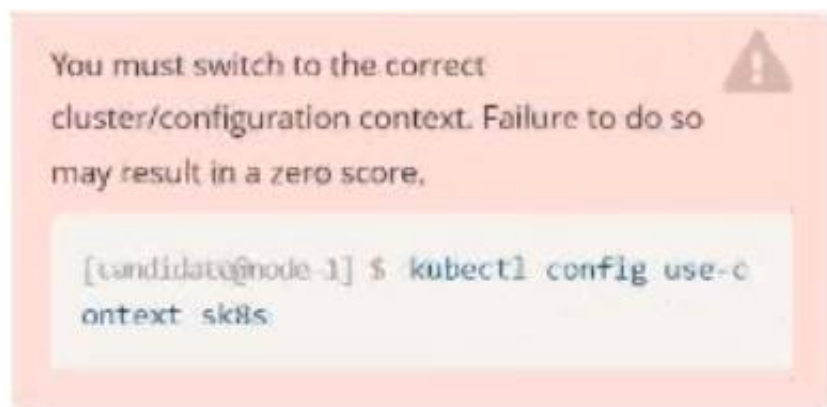
my-nginx-3800858182-jr4a2 1/1 Running 0 13s 10.244.3.4 kubernetes-minion-905m my-nginx-3800858182-kna2y 1/1 Running 0 13s 10.244.2.5 kubernetes-minion-ljyd Check your pods' IPs:

kubectl get pods -l run=my-nginx -o yaml | grep podIP podIP: 10.244.3.4

podIP: 10.244.2.5

## NEW QUESTION 13

Exhibit:



Task:

Create a Pod named nginx resources in the existing pod resources namespace. Specify a single container using nginx:stable image. Specify a resource request of 300m cpus and 1G1 of memory for the Pod's container.

- 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 run nginx-resources -n pod-resources --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
```

Text Description automatically generated with medium confidence

```
File Edit View Terminal Tabs Help
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx:stable
    name: nginx-resources
    resources:
      requests:
        cpu: 300m
        memory: "1Gi"
```

Text Description automatically generated

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl run nginx-resources -n pod-resources --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
candidate@node-1:~$ kubectl create -f hw.yaml
pod/nginx-resources created
candidate@node-1:~$ kubectl get pods -n pod-resources
NAME          READY   STATUS    RESTARTS   AGE
nginx-resources 1/1     Running   0           13s
candidate@node-1:~$ kubectl describe pods -n pod-resources
```

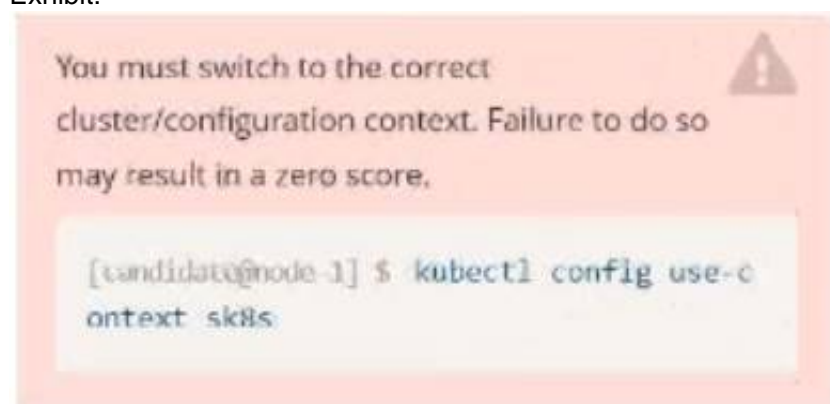
Text Description automatically generated



```
File Edit View Terminal Tabs Help
memory: 1Gi
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dmx9j (ro)
Conditions:
  Type              Status
  Initialized        True
  Ready              True
  ContainersReady    True
  PodScheduled       True
Volumes:
  kube-api-access-dmx9j:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:        true
QoS Class:           Burstable
Node-Selectors:      <none>
Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                     node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason      Age   From          Message
  ----    -
  Normal  Scheduled   20s   default-scheduler  Successfully assigned pod-resources/nginx-resources to k8s-node-0
  Normal  Pulling     19s   kubelet         Pulling image "nginx:stable"
  Normal  Pulled      13s   kubelet         Successfully pulled image "nginx:stable" in 6.55664052s
  Normal  Created     13s   kubelet         Created container nginx-resources
  Normal  Started     12s   kubelet         Started container nginx-resources
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>
```

## NEW QUESTION 18

Exhibit:



Task:

The pod for the Deployment named nosql in the craytish 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
```

```
File Edit View Terminal Tabs Help
---
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 --
12,1 All

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"
:~
```

```
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 23

Exhibit:



Task

Create a new deployment for running nginx with the following parameters;

- Run the deployment in the kdpd00201 namespace. The namespace has already been created
- Name the deployment frontend and configure with 4 replicas
- Configure the pod with a container image of lfcncf/nginx:1.13.7
- Set an environment variable of NGINX PORT=8080 and also expose that port for the container above

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml
student@node-1:~$ vim nginx_deployment.yml
```



Readme
Web Terminal
THE **LINUX** FOUNDATION

```

apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: api
  name: api
  namespace: kdpd00201
spec:
  replicas: 4
  selector:
    matchLabels:
      app: api
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: api
    spec:
      containers:
      - image: lfcncf/nginx:1.13.7-alpine
        name: nginx
        resources: {}
status: {}
~
"nginx_deployment.yml" 25L, 421C
4,1 All

```

Readme
Web Terminal
THE **LINUX** FOUNDATION

```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: api
  name: api
  namespace: kdpd00201
spec:
  replicas: 4
  selector:
    matchLabels:
      app: api
  template:
    metadata:
      labels:
        app: api
    spec:
      containers:
      - image: lfcncf/nginx:1.13.7-alpine
        name: nginx
        ports:
        - containerPort: 8080
      env:
      - name: NGINX_PORT
        value: "8080"
~
23,8 All

```

Readme
Web Terminal
THE **LINUX** FOUNDATION

```

student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml
student@node-1:~$ vim nginx_deployment.yml
student@node-1:~$ kubectl create nginx_deployment.yml
Error: must specify one of -f and -k

error: unknown command "nginx_deployment.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_deployment.yml
error: error validating "nginx_deployment.yml": error validating data: ValidationError(Deployment.spec.template.spec): unknown field "env" in io.k8s.api.core.v1.PodSpec; if you choose to ignore these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_deployment.yml
student@node-1:~$ kubectl create -f nginx_deployment.yml
deployment.apps/api created
student@node-1:~$ kubectl get pods -n kdpd00201
NAME                                READY   STATUS    RESTARTS   AGE
api-745677f7dc-7hnmv               1/1     Running   0           13s
api-745677f7dc-9q5vp               1/1     Running   0           13s
api-745677f7dc-fd4gk               1/1     Running   0           13s
api-745677f7dc-mbnpc               1/1     Running   0           13s
student@node-1:~$

```

## NEW QUESTION 27

.....

## Relate Links

**100% Pass Your CKAD Exam with ExamBible Prep Materials**

<https://www.exambible.com/CKAD-exam/>

## Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>