

CKAD Dumps

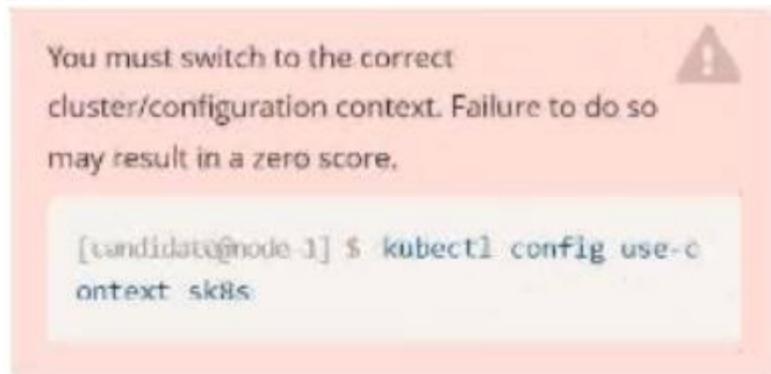
Certified Kubernetes Application Developer (CKAD) Program

<https://www.certleader.com/CKAD-dumps.html>



NEW QUESTION 1

Exhibit:



Task:

Create a Deployment named expose in the existing ckad00014 namespace running 6 replicas of a Pod. Specify a single container using the ifccncf/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:~$
```

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 lfcncf/nginx:1.13.7 --dry-run=client -o yaml > d
ep.yaml
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:~$

```

NEW QUESTION 2

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:

```

Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl run cache --image=lfcncf/redis:3.2 --port=6379 -n web
pod/cache created
student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS              RESTARTS   AGE
cache    0/1     ContainerCreating   0           6s
student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS    RESTARTS   AGE
cache    1/1     Running   0           9s
student@node-1:~$

```

NEW QUESTION 3

Exhibit:


```

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.yaml
student@node-1:~$ vim nginx_secret.yaml
student@node-1:~$ kubectl create -f nginx_secret.yaml
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:



Context

A project that you are working on has a requirement for persistent data to be available. Task

To facilitate this, perform the following tasks:

- Create a file on node sk8s-node-0 at /opt/KDSP00101/data/index.html with the content Acct=Finance
- Create a PersistentVolume named task-pv-volume using hostPath and allocate 1Gi to it, specifying that the volume is at /opt/KDSP00101/data on the cluster's node. The configuration should specify the access mode of ReadWriteOnce . It should define the StorageClass name exam for the PersistentVolume , which will be used to bind PersistentVolumeClaim requests to this PersistentVolume.
- Create a PersistentVolumeClaim named task-pv-claim that requests a volume of at least 100Mi and specifies an access mode of ReadWriteOnce
- Create a pod that uses the PersistentVolumeClaim as a volume with a label app: my-storage-app mounting the resulting volume to a mountPath /usr/share/nginx/html inside the pod

You can access `sk8s-node-0` by issuing the following command:

```
[student@node-1] $ | ssh sk8s-node-0
```

Ensure that you return to the base node (with hostname `node-1`) once you have completed your work on `sk8s-node-0`

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$
```

```
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/advantage

System information as of Fri Oct 9 08:52:09 UTC 2020

System load: 2.02          Users logged in: 0
Usage of /:  10.3% of 242.29GB IP address for eth0: 10.250.3.115
Memory usage: 2%          IP address for docker0: 172.17.0.1
Swap usage:  0%           IP address for cni0:  10.244.1.1
Processes:   38

* Kubernetes 1.19 is out! Get it in one command with:

  sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

7 packages can be updated.
1 update is a security update.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@sk8s-node-0:~$
```

```
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml
```



```

Readme Web Terminal THE LINUX FOUNDATION

apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: my-storage-app
spec:
  containers:
  - name: myfrontend
    image: nginx
    volumeMounts:
    - mountPath: "/usr/share/nginx/html"
      name: mypod
  volumes:
  - name: mypod
    persistentVolumeClaim:
      claimName: task-pv-clai
  
```

```

student@sk8s-node-0:~$ kubectl create -f pod.yml
pod/mypod created
student@sk8s-node-0:~$ kubectl get
  
```

```

Readme Web Terminal THE LINUX FOUNDATION

student@sk8s-node-0:~$ kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
mypod    0/1     ContainerCreating   0           4s
student@sk8s-node-0:~$ kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
mypod    0/1     ContainerCreating   0           8s
student@sk8s-node-0:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mypod    1/1     Running   0           10s
student@sk8s-node-0:~$ logout
Connection to 10.250.3.115 closed.
student@node-1:~$
  
```

NEW QUESTION 5

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:

```

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 6

Exhibit:

```

[student@node-1] $ | kubectl config
use-context k8s

```

Context

Developers occasionally need to submit pods that run periodically. Task

Follow the steps below to create a pod that will start at a predetermined time and which runs to completion only once each time it is started:

- Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml that runs the following shell command: date in a single busybox container. The command should run every minute and must complete within 22 seconds or be terminated by Kubernetes. The Cronjob name and container name should both be hello
- Create the resource in the above manifest and verify that the job executes successfully at least once

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yaml", allowed formats are: go-t
emplate, go-template-file, json, jsonpath, jsonpath-as-json, jsonpath-file, name, template, templatefile
, yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml

```

```

Readme Web Terminal THE LINUX FOUNDATION
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    metadata:
      name: hello
    spec:
      template:
        spec:
          containers:
            - image: busybox
              name: hello
              args: ["/bin/sh", "-c", "date"]
              restartPolicy: Never
  schedule: '*/* * * * *'
  startingDeadlineSeconds: 22
  concurrencyPolicy: Allow

```

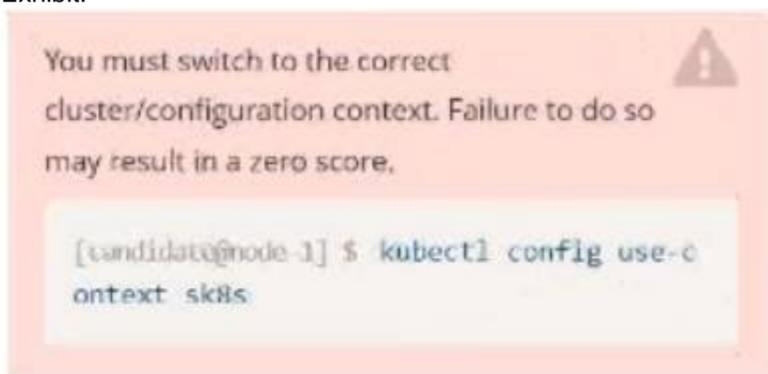
```

Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yaml", allowed formats are: go-template, go-template-file, json, jsonpath, jsonpath-as-json, jsonpath-file, name, template, templatefile, yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
student@node-1:~$ kubectl create -f /opt/KDPD00301/periodic.yaml
cronjob.batch/hello created
student@node-1:~$ kubectl get cronjob
NAME      SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
hello    */* * * * *   False    0        <none>           6s
student@node-1:~$

```

NEW QUESTION 7

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 8

Exhibit:



Context

A user has reported an application is unteachable due to a failing livenessProbe . Task

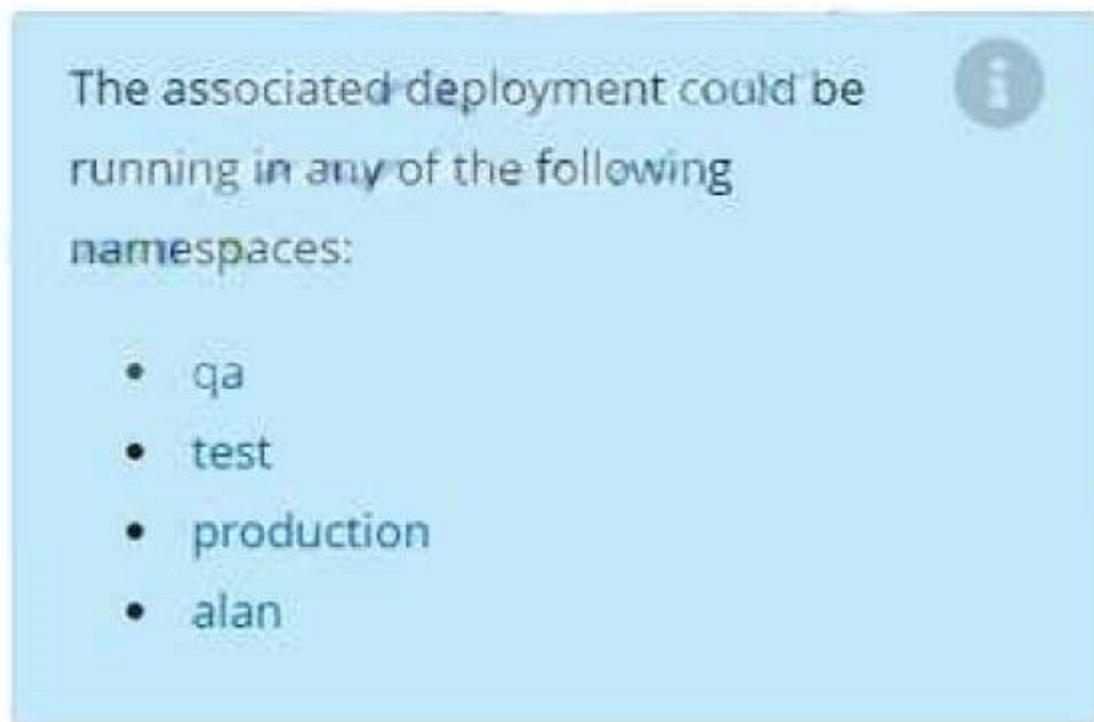
Perform the following tasks:

- 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 SubjectPath 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.

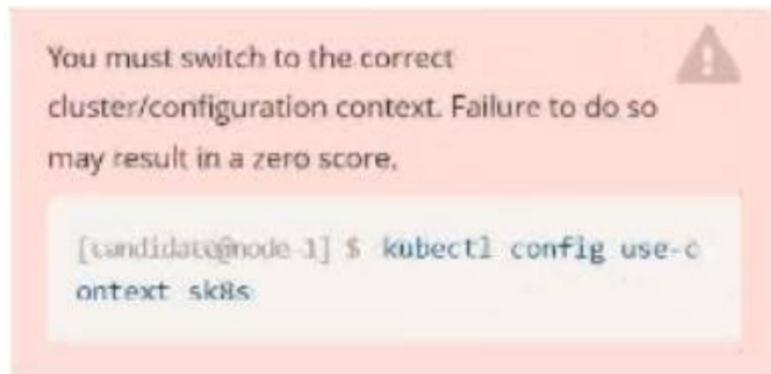
FirstSeen LastSeen Count From SubjectPath Type Reason Message

```
-----
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 9

Exhibit:



Task:

Modify the existing Deployment named broker-deployment running in namespace quetzal so that its containers. The broker-deployment is manifest file can be found at:

```
~/daring-mocasin/broker-deployment.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
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
containers:
- name: broker
  image: redis:alpine
  ports:
  - containerPort: 6379
  securityContext:
    runAsUser: 30008
    privileged: false
```

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/daring-mocasin/broker-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/daring-mocasin/broker-deployment.yaml
deployment.apps/broker-deployment configured
candidate@node-1:~$ kubectl get pods -n quetzal
NAME                                READY   STATUS    RESTARTS   AGE
broker-deployment-65446d6d94-868p6  1/1     Running   0           30s
broker-deployment-65446d6d94-8dn7l  1/1     Running   0           32s
broker-deployment-65446d6d94-p4h4l  1/1     Running   0           31s
candidate@node-1:~$ kubectl get deploy -n quetzal
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
broker-deployment  3/3     3             3           7h3m
candidate@node-1:~$
```

NEW QUESTION 10

Exhibit:



Context

You have been tasked with scaling an existing deployment for availability, and creating a service to expose the deployment within your infrastructure.

Task

Start with the deployment named kdsn00101-deployment which has already been deployed to the namespace kdsn00101 . Edit it to:

- Add the func=webFrontEnd key/value label to the pod template metadata to identify the pod for the service definition
- Have 4 replicas

Next, create a deploy in namespace kdsn00101 a service that accomplishes the following:

- Exposes the service on TCP port 8080
- is mapped to me pods defined by the specification of kdsn00101-deployment
- Is of type NodePort
- Has a name of cherry

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
```

Readme
Web Terminal
THE LINUX FOUNDATION

```

Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2020-10-09T08:50:39Z"
  generation: 1
  labels:
    app: nginx
  name: kdsn00101-deployment
  namespace: kdsn00101
  resourceVersion: "4786"
  selfLink: /apis/apps/v1/namespaces/kdsn00101/deployments/kdsn00101-deployment
  uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
"/tmp/kubectl-edit-d4y5r.yaml" 70L, 1957C 1,1 Top
    
```

The screenshot shows a web terminal interface with a dark background. At the top, there are buttons for 'Readme' and 'Web Terminal', and the 'THE LINUX FOUNDATION' logo. The terminal content is as follows:

```
uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 4
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
        func: webFrontEnd
    spec:
      containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
        - containerPort: 80
```

```
student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
deployment.apps/kdsn00101-deployment edited
student@node-1:~$ kubectl get deployment kdsn00101-deployment -n kdsn00101
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
kdsn00101-deployment  4/4     4             4           7h17m
student@node-1:~$ kubectl expose deployment kdsn00101-deployment -n kdsn00101 --type NodePort --port 8080 --name cherry
service/cherry exposed
```

NEW QUESTION 10

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:

The screenshot shows a web terminal interface similar to the first one. The terminal content is as follows:

```
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-7hnm  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 15

Exhibit:



Context

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

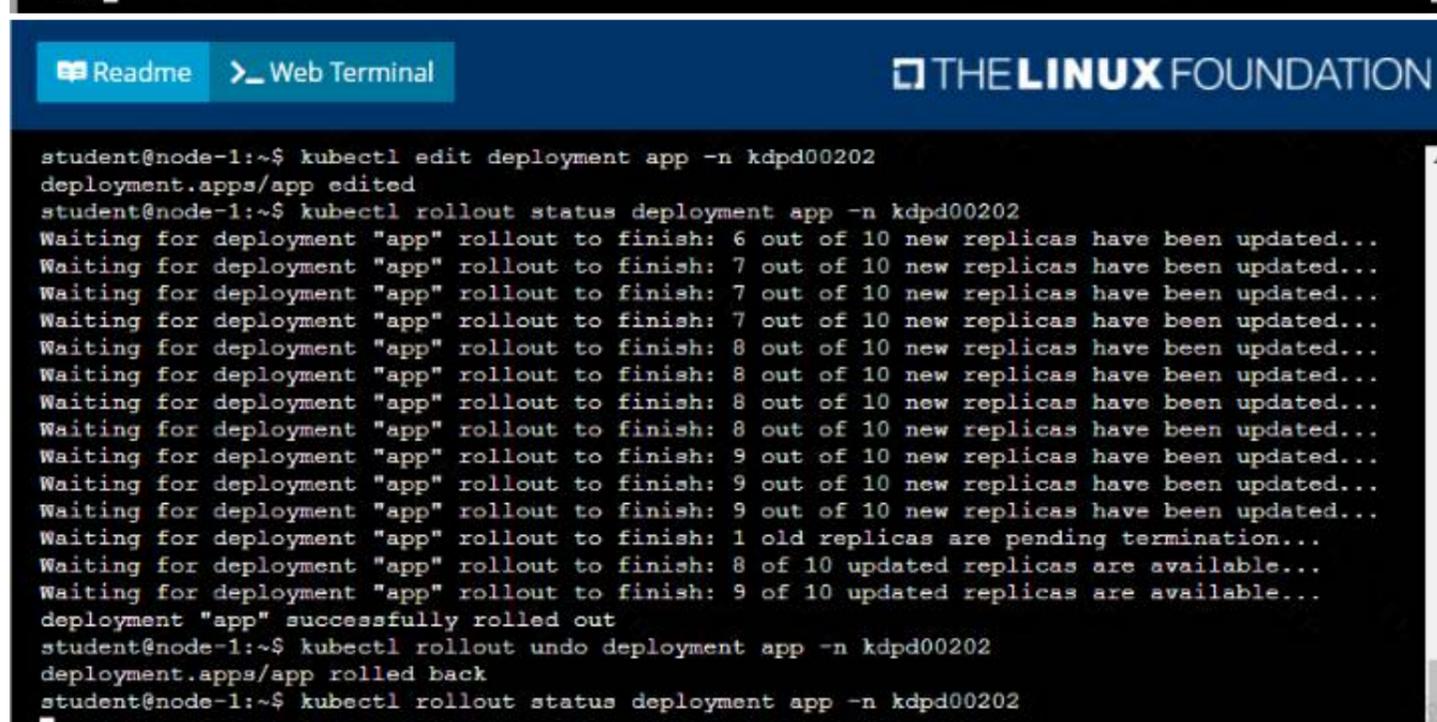
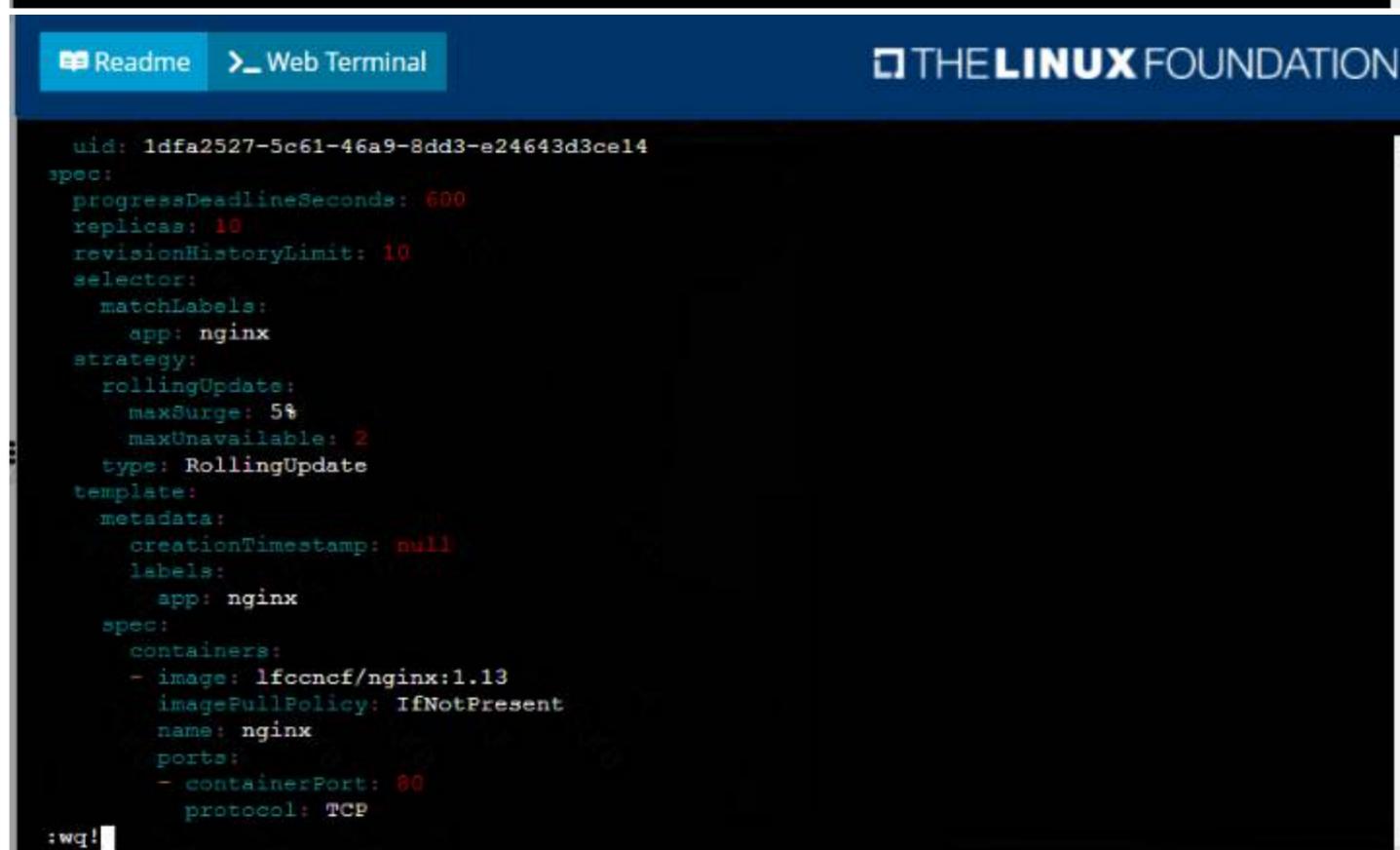
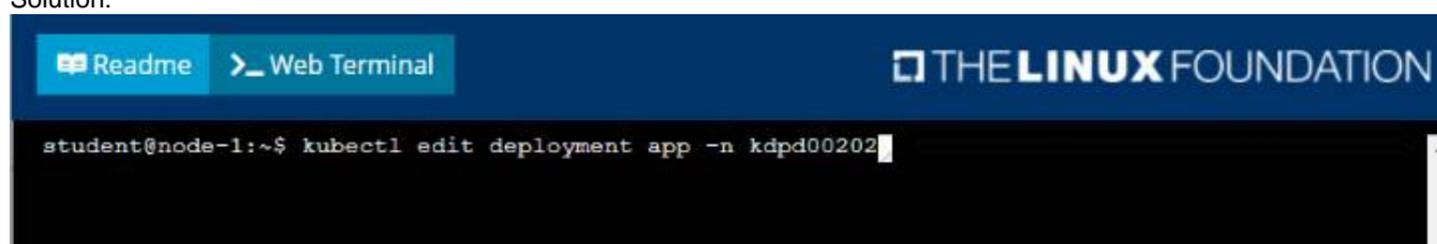
- Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%
- Perform a rolling update of the web1 deployment, changing the lfcncf/nginx image version to 1.13
- Roll back the app deployment to the previous version

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:



```
student@node-1:~$ kubectl 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: 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:~$
```

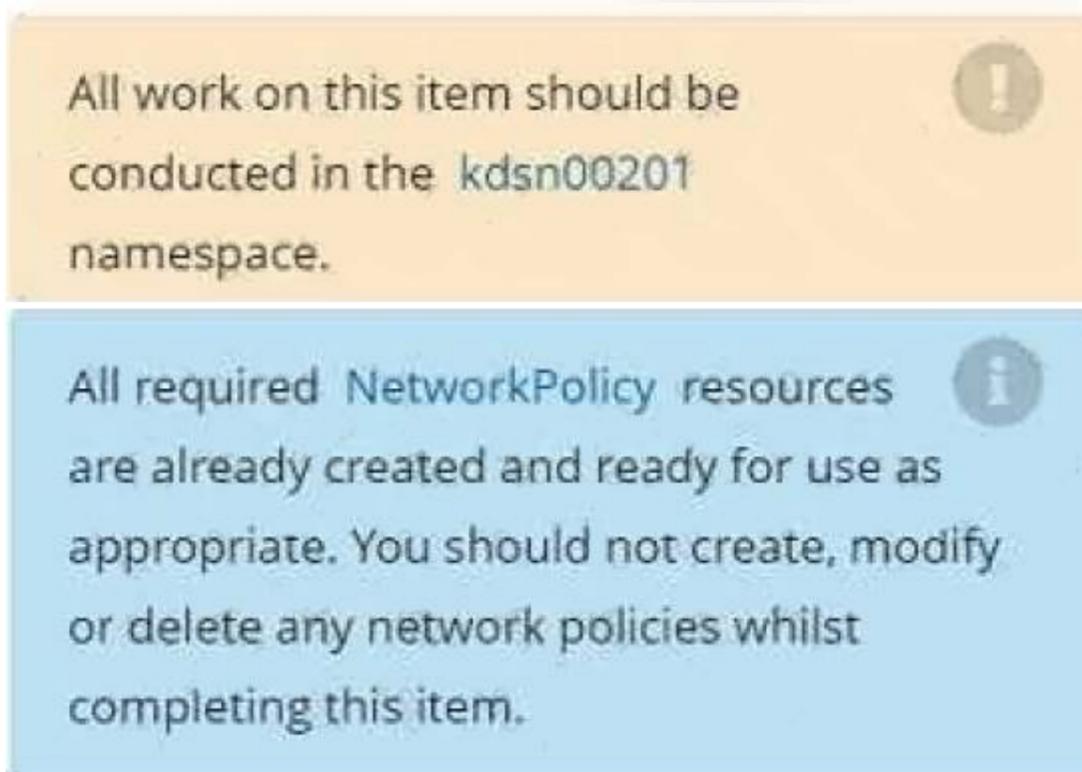
NEW QUESTION 17

Exhibit:



Task

You have rolled out a new pod to your infrastructure and now you need to allow it to communicate with the web and storage pods but nothing else. Given the running pod kdsn00201 -newpod edit it to use a network policy that will allow it to send and receive traffic only to and from the web and storage pods.



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
apiVersion: networking.k8s.io/v1 kind: NetworkPolicy
metadata:
name: internal-policy namespace: default spec:
podSelector: matchLabels: name: internal policyTypes:
- Egress
- Ingress ingress:
- {}
egress:
- to:
- podSelector: matchLabels: name: mysql ports:
- protocol: TCP port: 3306
- to:
- podSelector: matchLabels:
name: payroll ports:
- protocol: TCP port: 8080
- ports:
- port: 53 protocol: UDP
- port: 53 protocol: TCP
```

NEW QUESTION 19

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your CKAD Exam with Our Prep Materials Via below:

<https://www.certleader.com/CKAD-dumps.html>