# CKAD Q&As

Certified Kubernetes Application Developer (CKAD) Program

## Pass Linux Foundation CKAD Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

**https://www.pass4itsure.com/ckad.html**

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

**QUESTION 1**

CORRECT TEXT


Context You are tasked to create a secret and consume the secret in a pod using environment variables as follow: Task

1.

Create a secret named another-secret with a key/value pair; key1/value4

2.

Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key1, using COOL_VARIABLE as the name for the environment variable inside the pod

A. Please check explanations

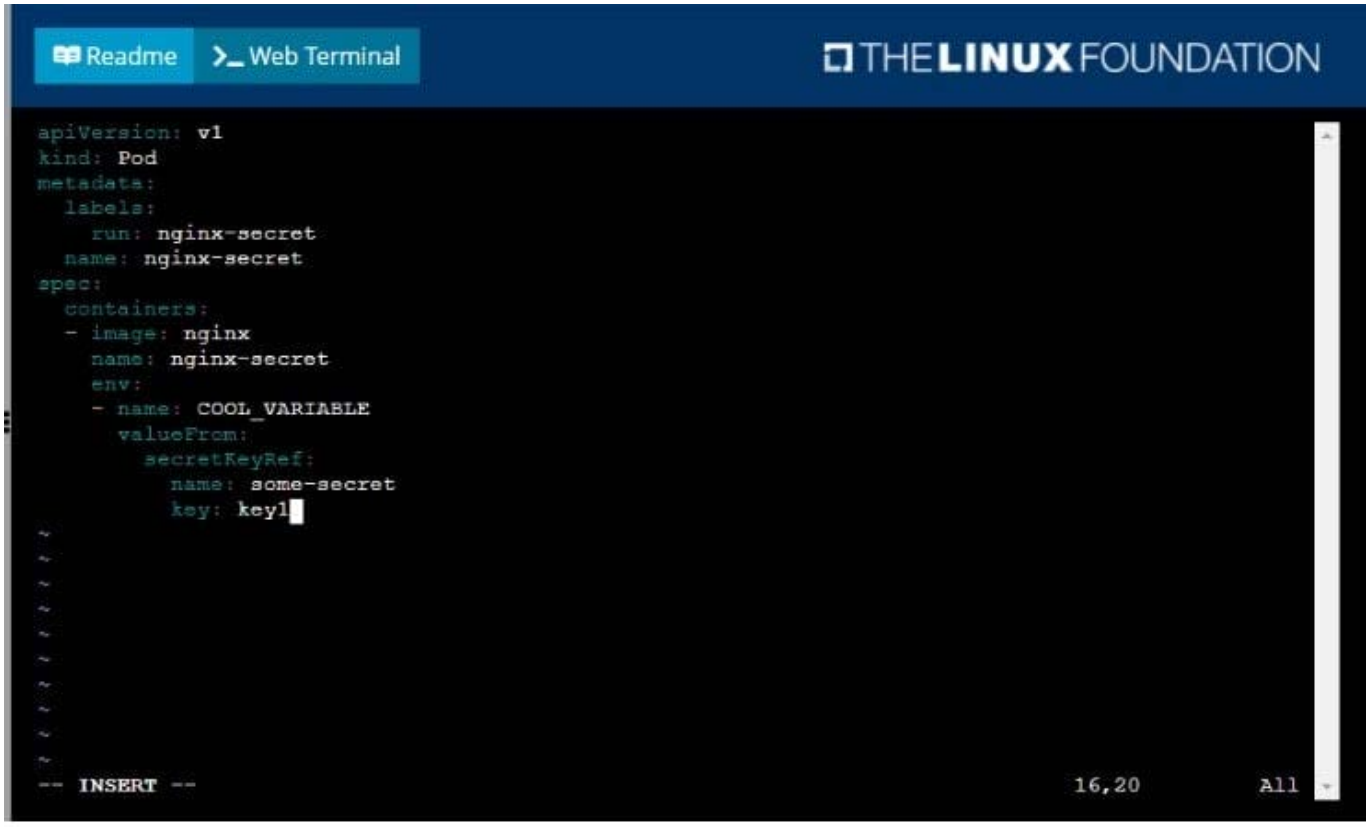
B. Place Holder

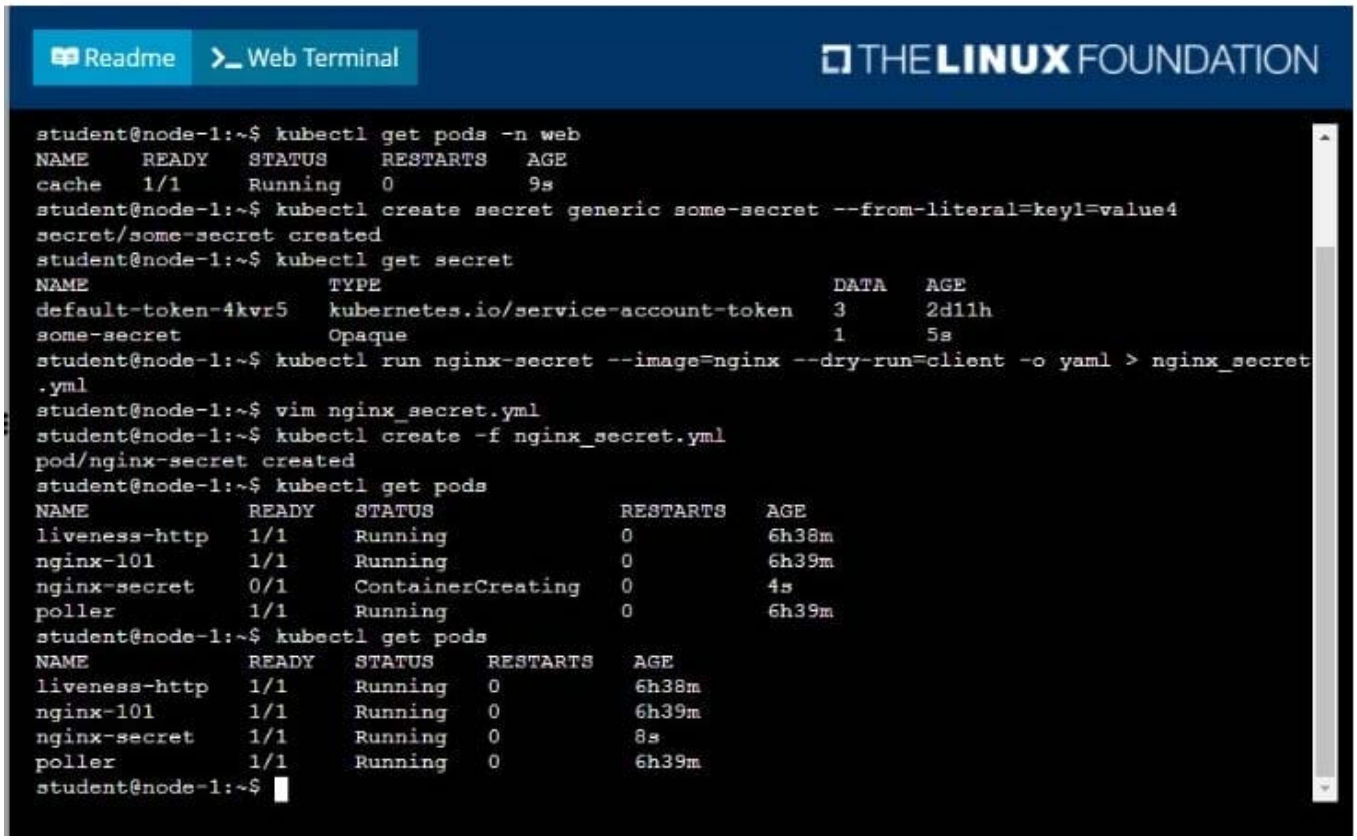
Correct Answer: A

```
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:~$

**QUESTION 2**

CORRECT TEXT


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:

1.

Create a file on node sk8s-node-0 at /opt/KDSP00101/data/index.html with the content Acct=Finance

2.

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.

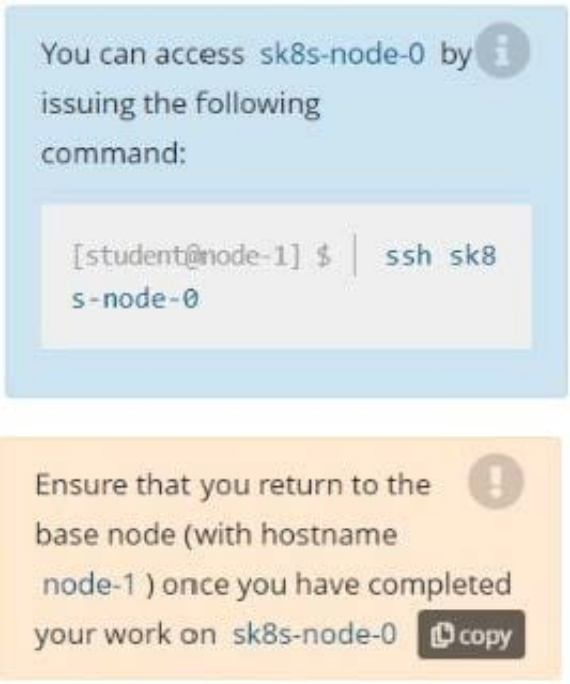
The configuration should specify the access mode of ReadWriteOnce. It should define the StorageClass name exam for the PersistentVolume, which will be used to bind PersistentVolumeClaim requests to this PersistenetVolume.

1.

Create a PefsissentVolumeClaim named task-pv-claim that requests a volume of at least 100Mi and specifies an access mode of ReadWriteOnce

2.

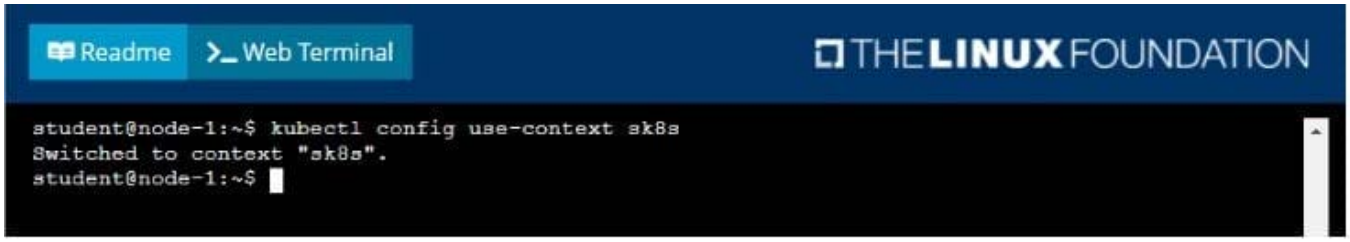
Create a pod that uses the PersistentVolmeClaim as a volume with a label app: my- storage-app mounting the resulting volume to a mountPath /usr/share/nginx/html inside the pod

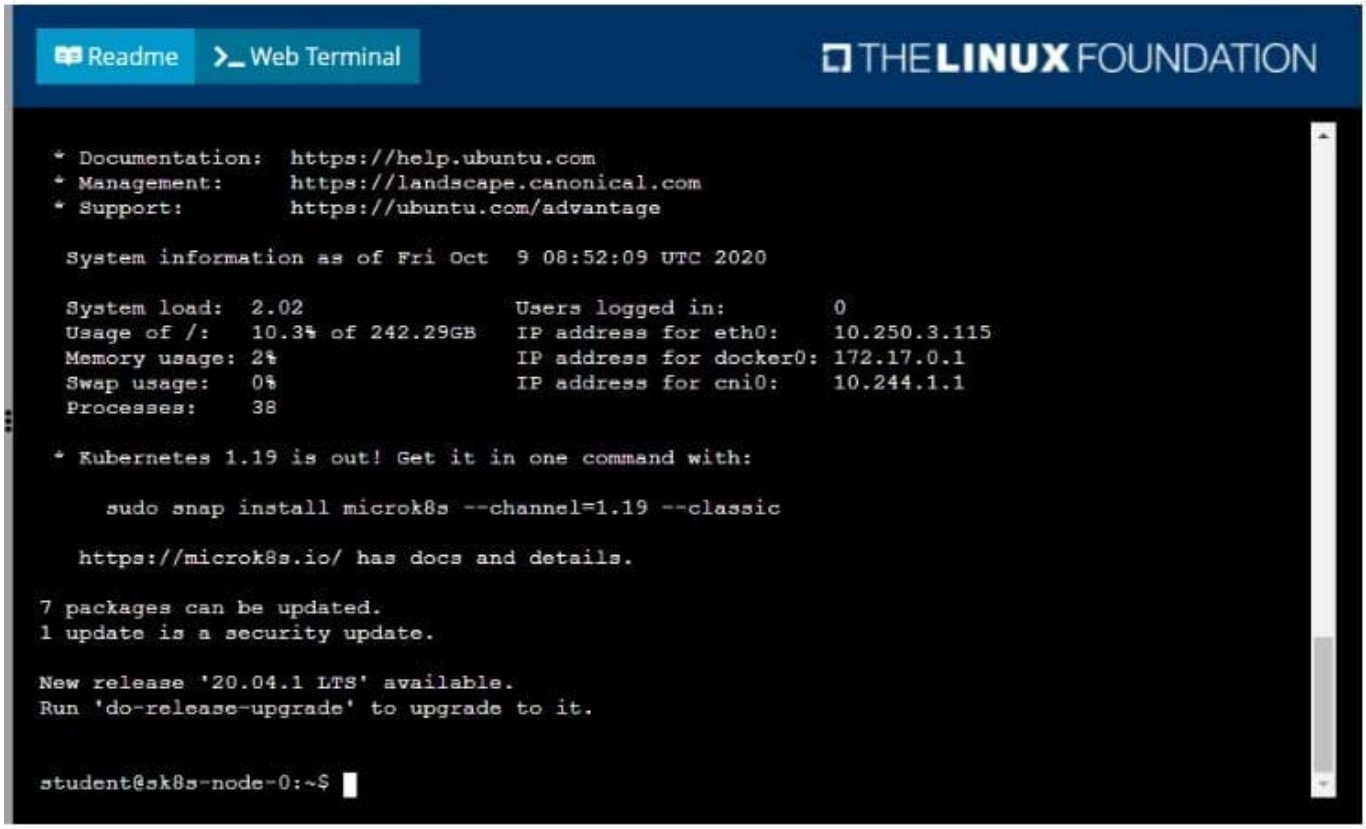


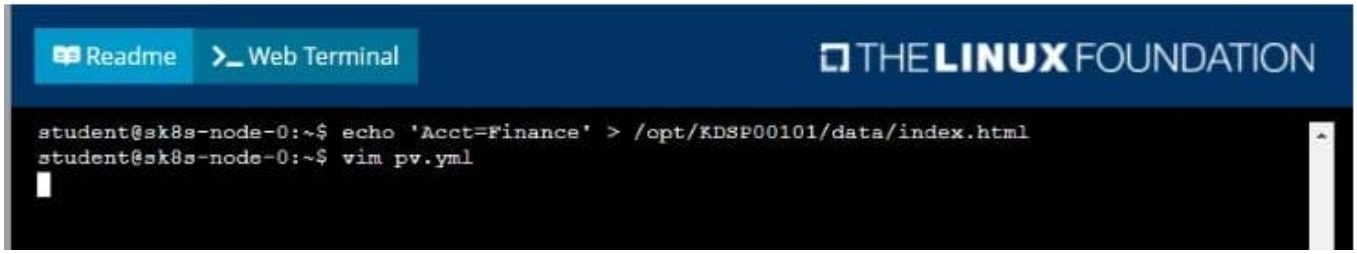

A. Please check explanations

B. Place Holder

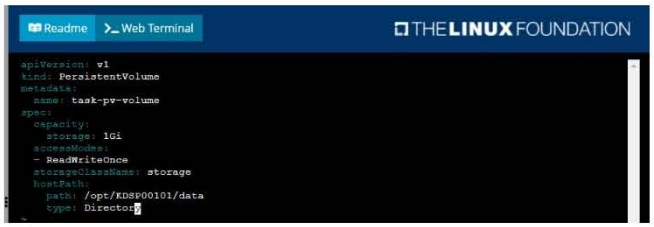
Correct Answer: A


Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$

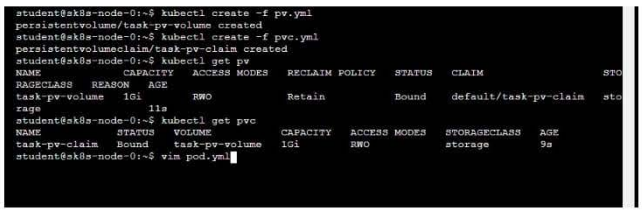

Readme
Web Terminal
THE LINUX FOUNDATION
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Fri Oct 9 08:52:09 UTC 2020
System load: 2.02
Usage of /: 10.3% of 242.29GB
Memory usage: 2%
Swap usage: 0%
Processes: 38
Users logged in: 0
IP address for eth0: 10.250.3.115
IP address for docker0: 172.17.0.1
IP address for cni0: 10.244.1.1
* 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:~$

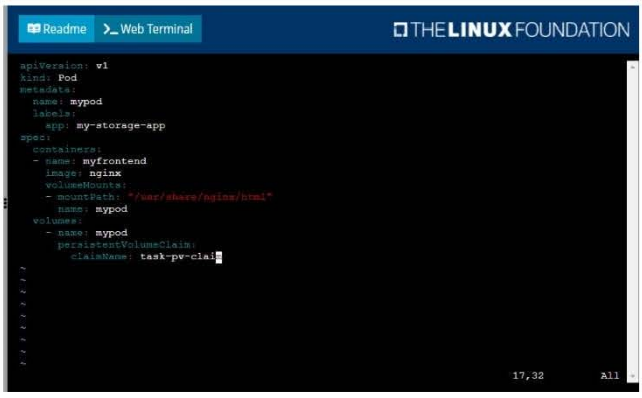

Readme
Web Terminal
THE LINUX FOUNDATION
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml

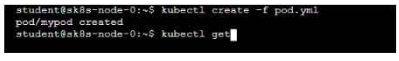

Readme
Web Terminal
THE LINUX FOUNDATION
-- INSERT --
0,1
All

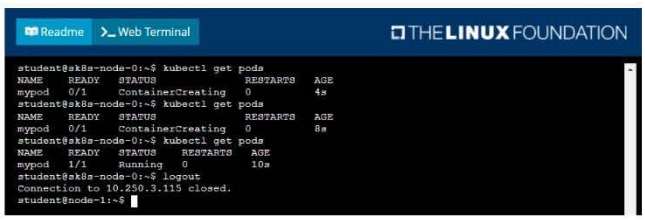

Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: PersistentVolume
metadata:
name: task-pv-volume
spec:
capacity:
storage: 1Gi
accessModes:
- ReadWriteOnce
storageClassName: storage
hostPath:
path: /opt/KDSP00101/data
type: Director


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: task-pv-claim
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 100Mi
storageClassName: storage


student@sk8s-node-0:~$ kubectl create -f pv.yml
persistentvolume/task-pv-volume created
student@sk8s-node-0:~$ kubectl create -f pvc.yml
persistentvolumeclaim/task-pv-claim created
student@sk8s-node-0:~$ kubectl get pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STO
RAGECLASS REASON AGE
task-pv-volume 1Gi RWO Retain Bound default/task-pv-claim sto
rage 11s
student@sk8s-node-0:~$ kubectl get pvc
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
task-pv-claim Bound task-pv-volume 1Gi RWO storage 9s
student@sk8s-node-0:~$ vim pod.yml

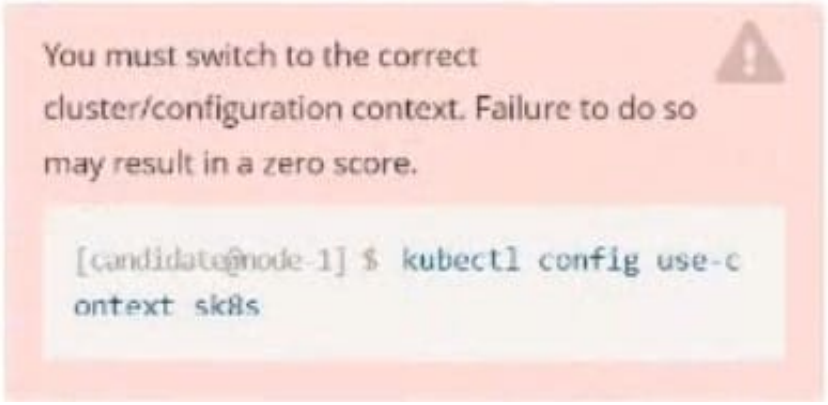

Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
name: mypod
labels:
app: my-storage-app
spec:
containers:
- name: myfrontend
image: nginx
volumeMounts:
- mountPath: "/usr/share/nginx/html"
name: mypod
volumes:
- name: mypod
persistentVolumeClaim:
claimName: task-pv-clai
17,32
All


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

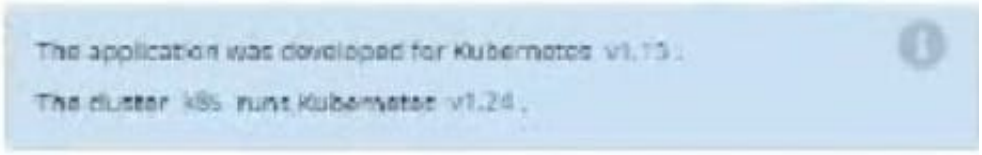

Readme
Web Terminal
THE LINUX FOUNDATION
student@sk8s-node-0:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
mypod 0/1 ContainerCreating 0 4s
student@sk8s-node-0:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
mypod 0/1 ContainerCreating 0 8s
student@sk8s-node-0:~$ kubectl get pods
NAME READY STATUS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

**QUESTION 3**

CORRECT TEXT




Task:



A. Please check explanations

B. Place Holder

Correct Answer: A

```
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: {}
~
~
:wq
```

```
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-fld7j   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:~$
```

**QUESTION 4**

CORRECT TEXT


Context

Your application\\'s namespace requires a specific service account to be used.

Task

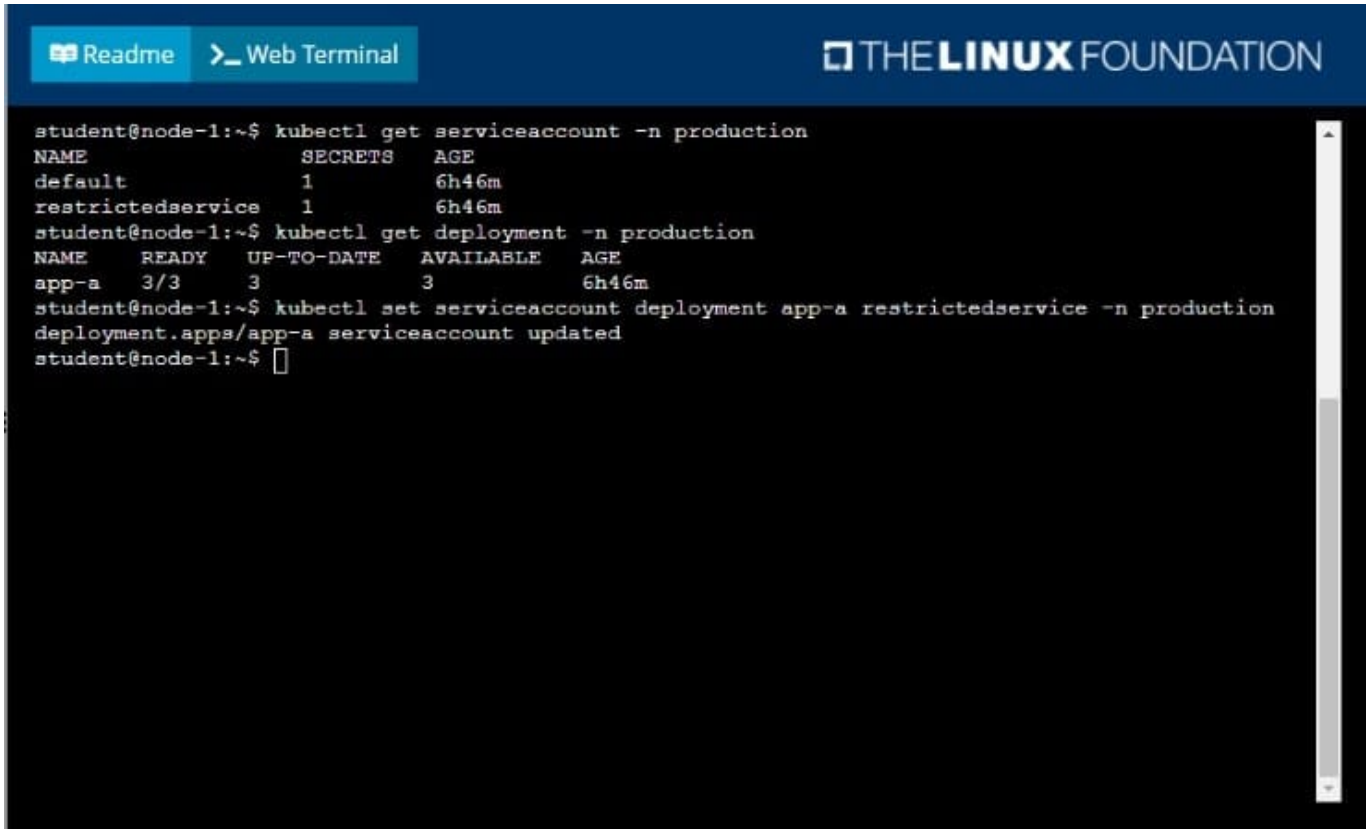
Update the app-a deployment in the production namespace to run as the restrictedservice service account.

The service account has already been created.

A. Please check explanations

B. Place Holder

Correct Answer: A




**QUESTION 5**

CORRECT TEXT


Context

You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount.

Task

Please complete the following:

1.

Create a ConfigMap named another-config containing the key/value pair: key4/value3

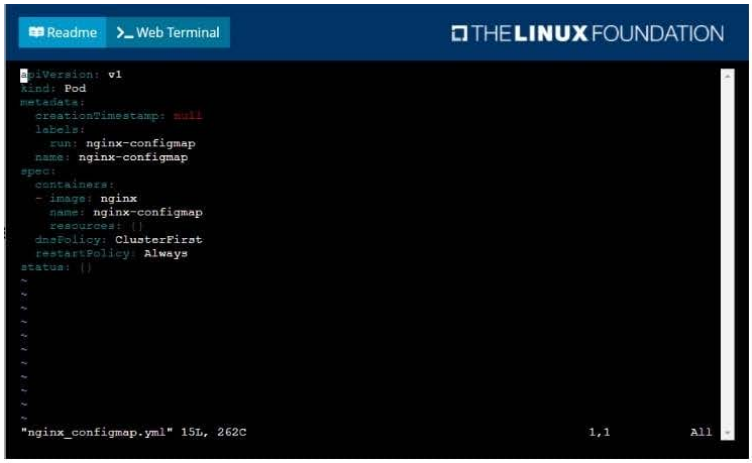
2.

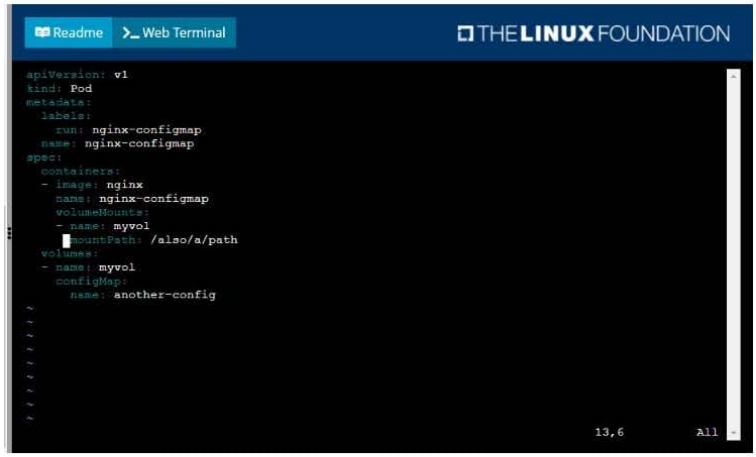
start a pod named nginx-configmap containing a single container using the nginx image, and mount the key you just created into the pod under directory /also/a/path

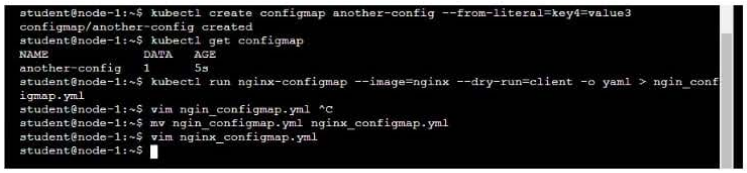
A. Please check explanations

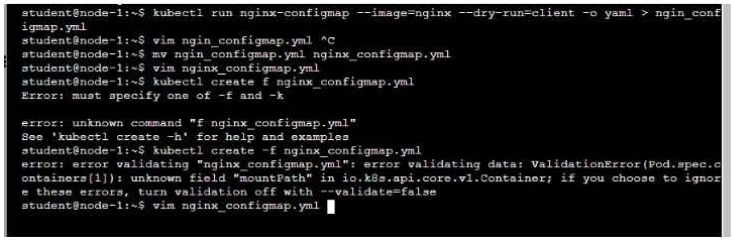
B. Place Holder

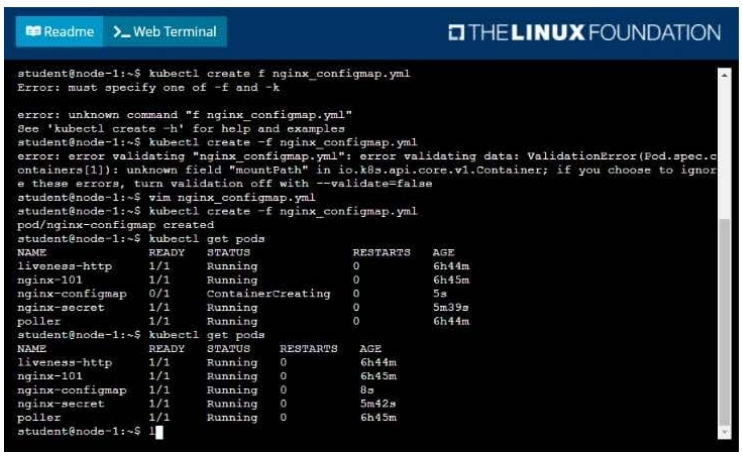
Correct Answer: A


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 > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_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 > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_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
THE LINUX FOUNDATION
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
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

**QUESTION 6**

CORRECT TEXT


Context

You sometimes need to observe a pod\\'s logs, and write those logs to a file for further analysis.

Task

Please complete the following;

1.

Deploy the counter pod to the cluster using the provided YAMLspec file at /opt/KDOB00201/counter.yaml

2.

Retrieve all currently available application logs from the running pod and store them in the file /opt/KDOB0020l/log_Output.txt, which has already been created

A. Please check explanations

B. Place Holder

Correct Answer: A

```
student@node-1:~$ kubectl create -f /opt/KDOB00201/counter.yaml
pod/counter created
student@node-1:~$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
counter           1/1     Running   0          10s
liveness-http     1/1     Running   0          6h45m
nginx-101         1/1     Running   0          6h46m
nginx-configmap   1/1     Running   0          107s
nginx-secret      1/1     Running   0          7m21s
poller            1/1     Running   0          6h46m
student@node-1:~$ kubectl logs counter
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828fbe5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
student@node-1:~$ kubectl logs counter  > /opt/KDOB00201/log_output.txt
student@node-1:~$
```

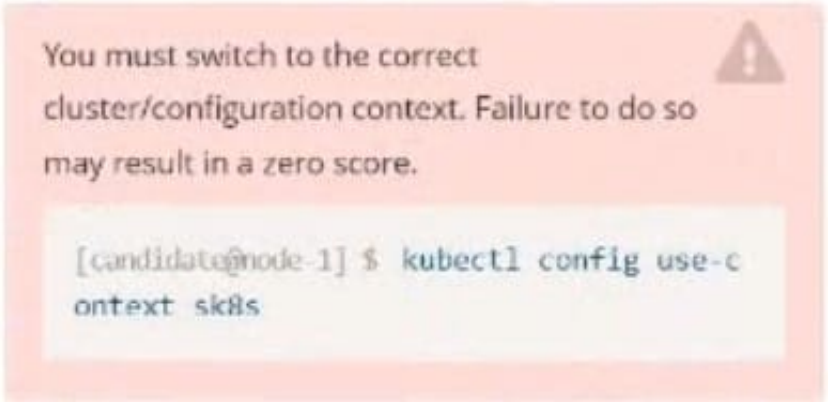
```
student@node-1:~$ kubectl logs counter  > /opt/KDOB00201/log_output.txt
student@node-1:~$ kubectl logs counter  > /opt/KDOB00201/log_output.txt
student@node-1:~$ cat /opt/KDOB00201/log_output.txt
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
student@node-1:~$ kubectl logs counter  > /opt/KDOB00201/log_output.txt
student@node-1:~$ cat /opt/KDOB00201/log_output.txt
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828fbe5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
11: 5493cd16a1790a5fb9512b0c9d4c5dd1
12: 03f169e93e6143438e6dfe4ecb3cc9ed
13: 764b37fe611373c42d0b47154041f6eb
14: 1a56fbe1896b0ee63941361662818 39e
15: ecc492eb17715de090c47345a98d98d3
16: 7974a6bec0fb44b6b8bbfc71aa3fbe74
17: 9ae01bef01740b12cc9f97a5f9f72cd6
18: 23fb22ee34d4272e4c9e005f1774515f
19: ec7e1a5d314da9a0ad45d53be5a7acae
20: 0bccdd8ee02cd42029e8162cd1c1197c
21: d6851ea43546216b95bcb81ced997102
22: 7ed9a38ea8bf0d86206569481442af44
23: 29b8416ddc63dbfcb987ab3c8198e9fe
24: 1f2062001df51a108ab25010f506716f
student@node-1:~$
```

**QUESTION 7**

CORRECT TEXT




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 Ifconf/nginx container image.

3.

Perform a rollback of the web1 Deployment to its previous version

A. Please check explanations

B. Place Holder

Correct Answer: A

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

```
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:~$
```

**QUESTION 8**

CORRECT TEXT


Context

A user has reported an application is unreachable 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:

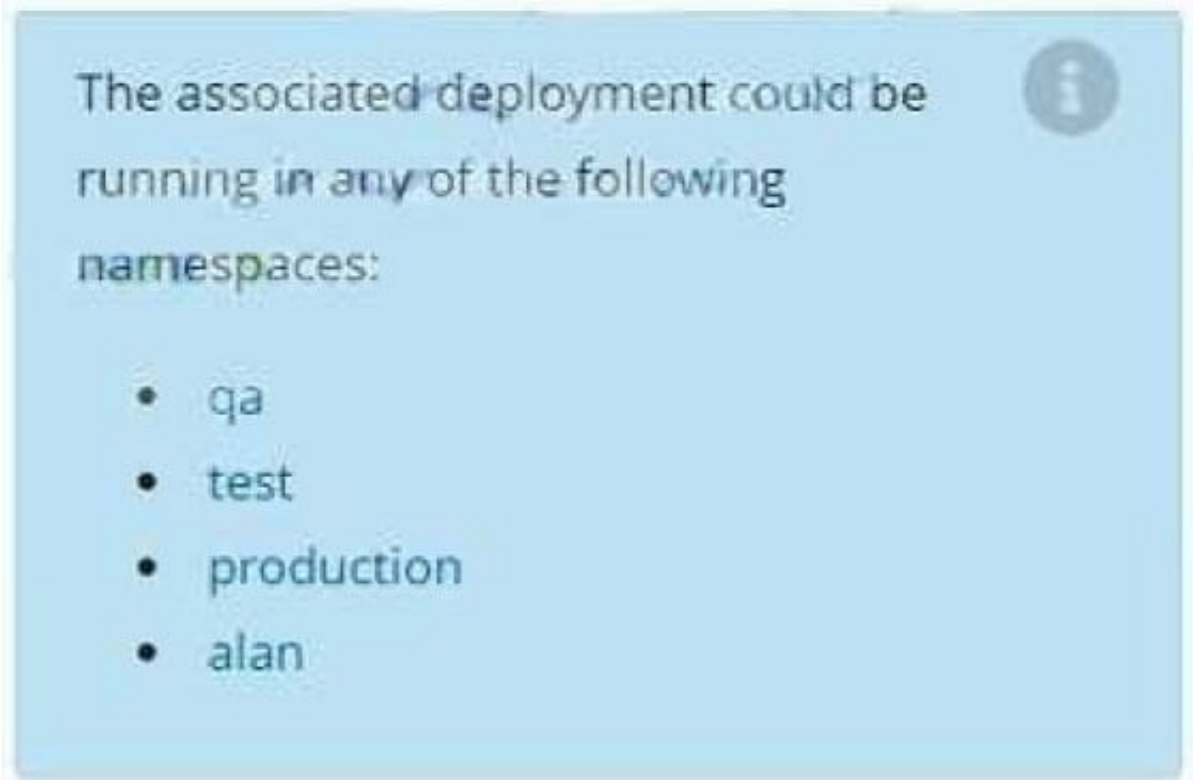

The output file has already been created

1.

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

2.

Fix the issue.




A. Please check explanations

B. Place Holder

Correct Answer: A

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

**QUESTION 9**

CORRECT TEXT


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:

1.

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

2.

Have 4 replicas

Next, create and deploy in namespace kdsn00l01 a service that accomplishes the following:

1.

Exposes the service on TCP port 8080

2.

is mapped to me pods defined by the specification of kdsn00l01-deployment

3.

Is of type NodePort

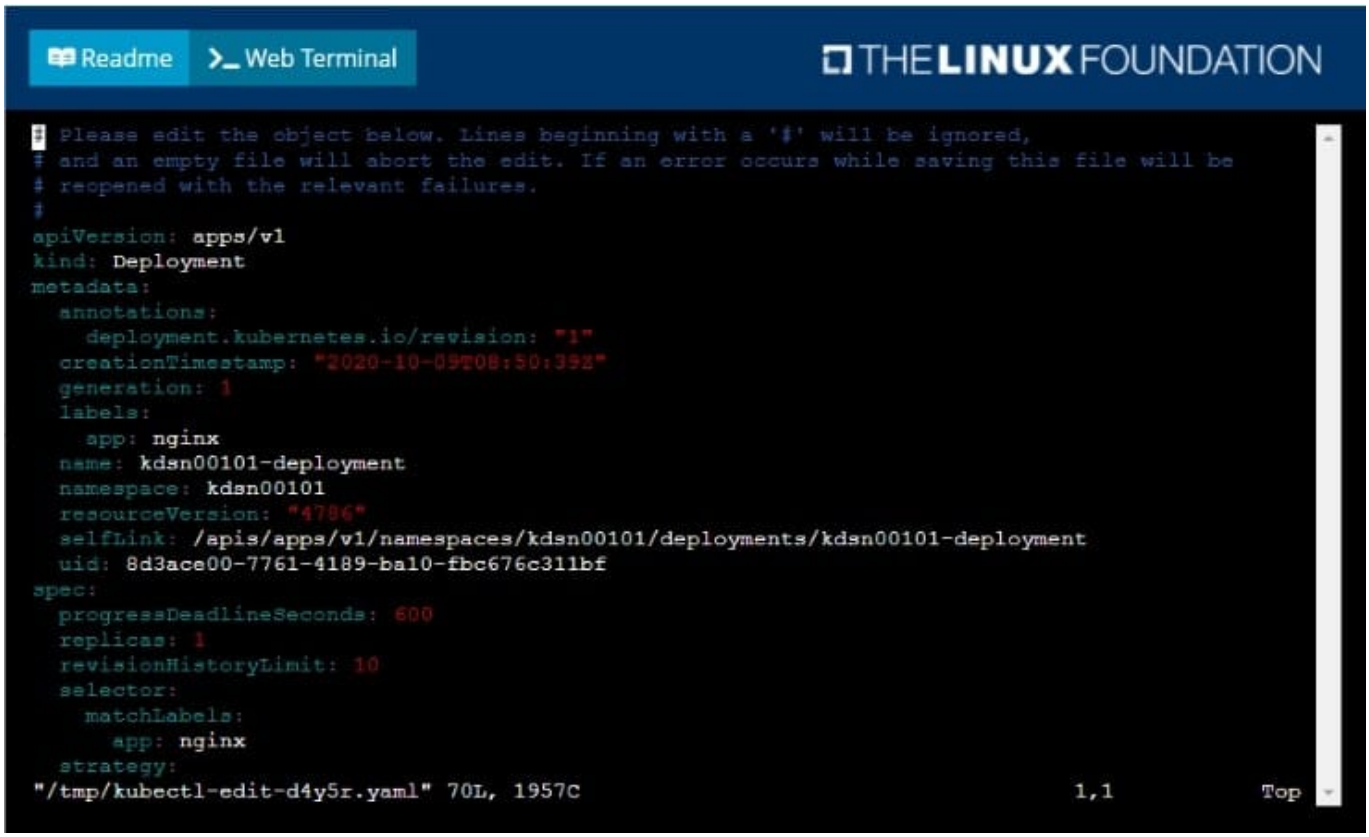
4.

Has a name of cherry

A. Please check explanations

B. Place Holder

Correct Answer: A

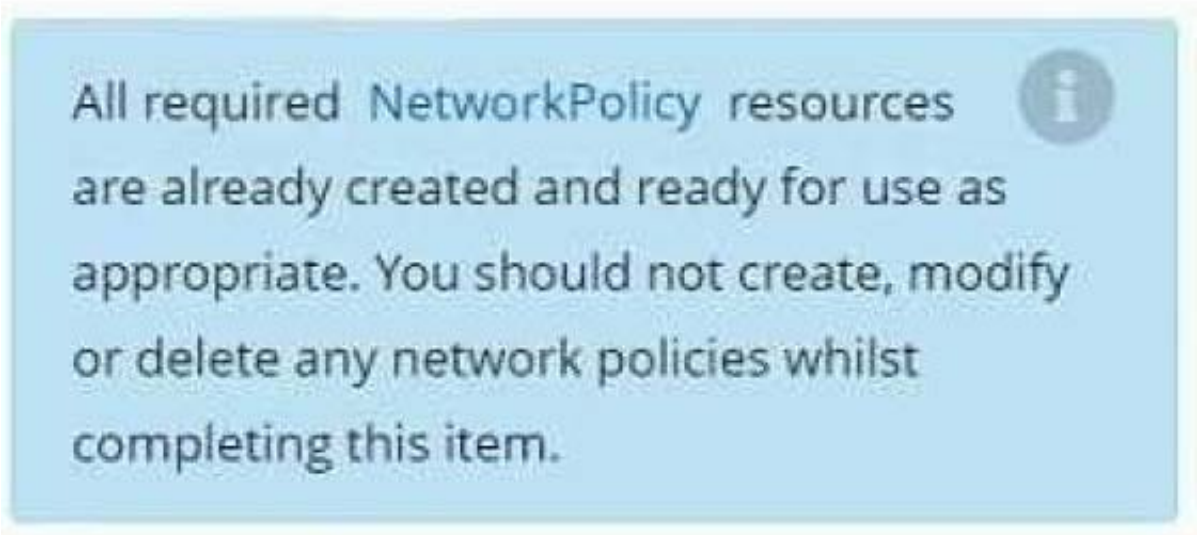


**QUESTION 10**

CORRECT TEXT


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. Please check explanations

B. Place Holder

Correct Answer: A

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