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.certbus.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

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.

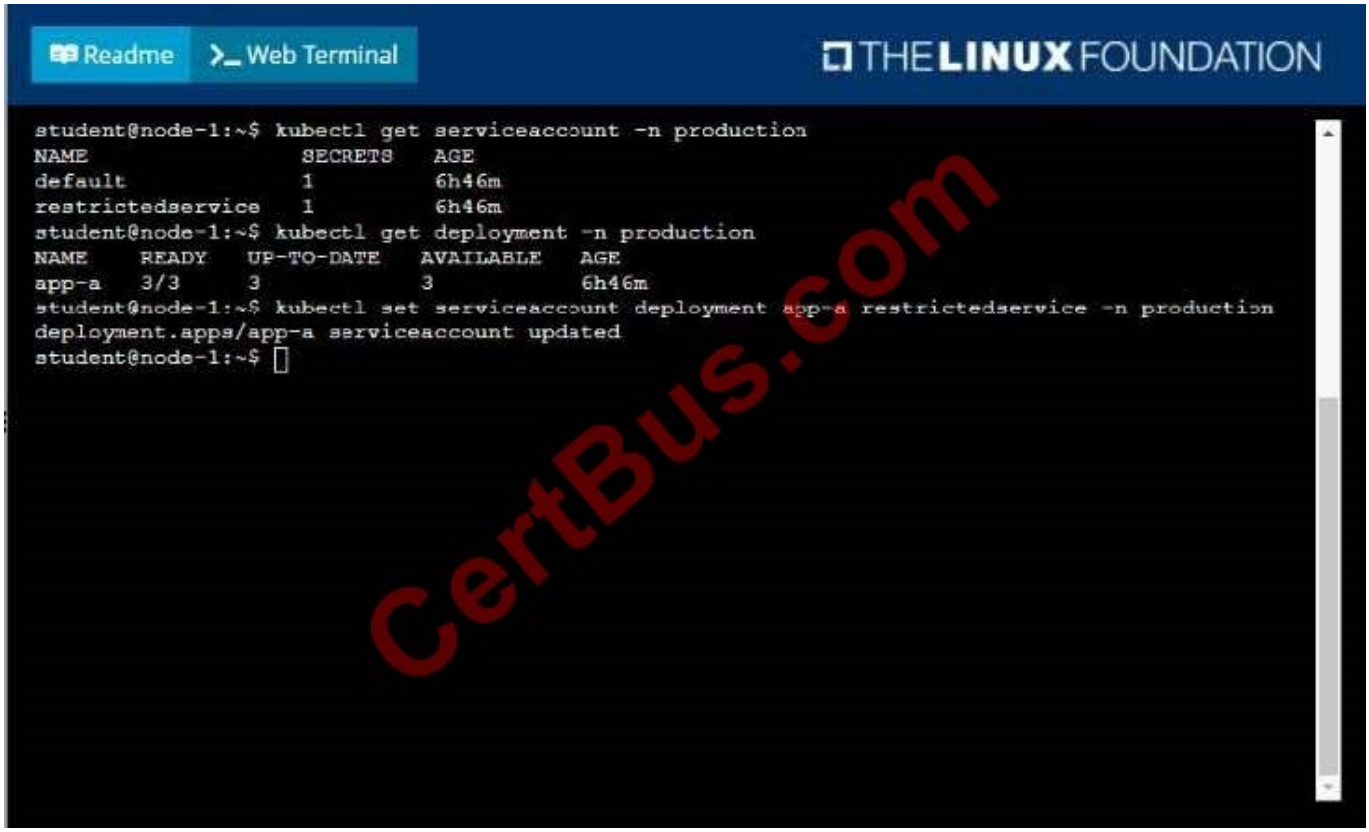
The service account has already been created.

A. Please check explations

B. Place Holder

Correct Answer: AB

Solution:



QUESTION 2

Exhibit: Given a container that writes a log file in format A and a container that converts log files from format A to format B, create a deployment that runs both containers such that the log files from the first container are converted by the second container, emitting logs in format B. Task: ?Create a deployment named deployment-xyz in the default namespace, that: ?Includes a primary lfccncf/busybox:1 container, named logger-dev ?includes a sidecar lfccncf/fluentd:v0.12 container, named adapter-zen ?Mounts a shared volume /tmp/log on both containers, which does not persist when the pod is deleted ?Instructs the logger-dev container to run the command

which should output logs to /tmp/log/input.log in plain text format, with example values:

The adapter-zen sidecar container should read /tmp/log/input.log and output the data to /tmp/log/output.* in Fluentd JSON format. Note that no knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at /opt/KDMC00102/fluentd-configma p.yaml , and mount that ConfigMap to /fluentd/etc in the adapter-zen sidecar container

A. Please check explations

B. Place Holder

Correct Answer: AB

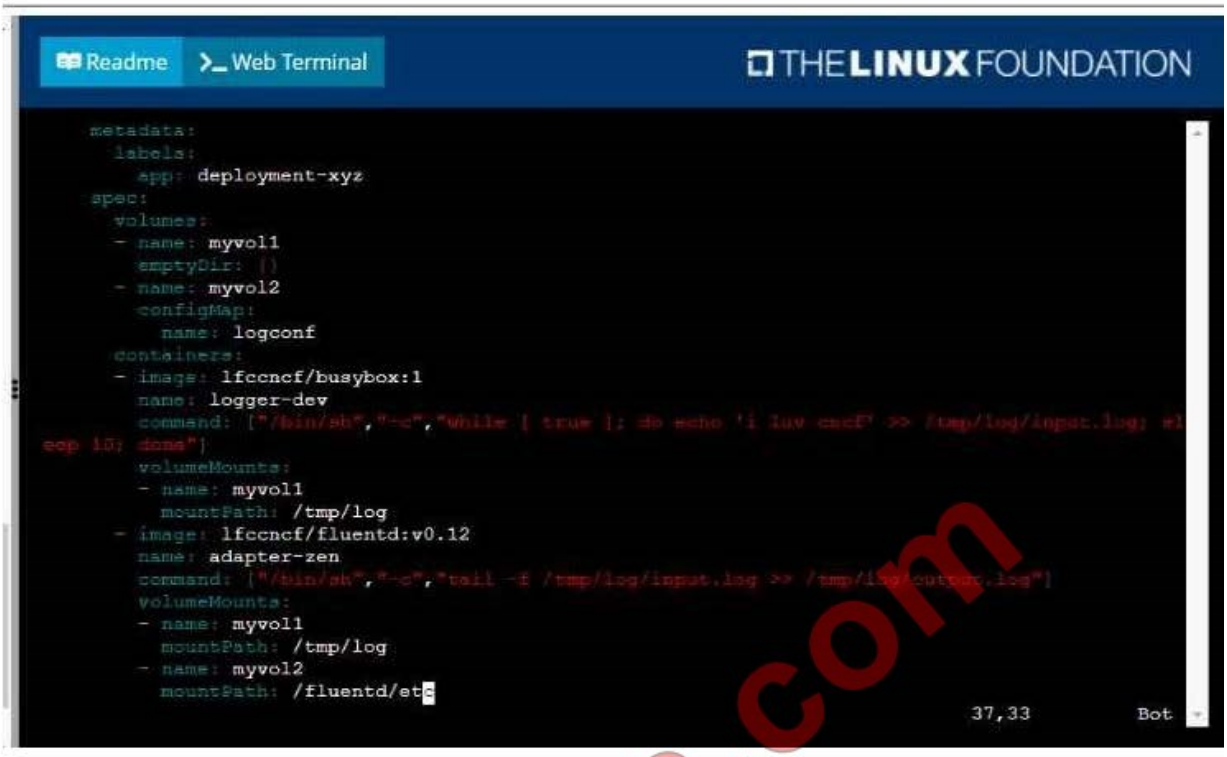
Solution:

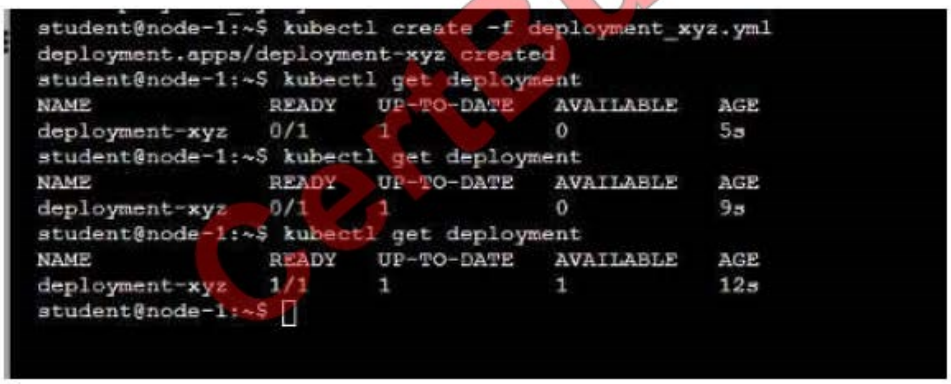

Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$ kubectl create deployment deployment-xyz --image=lfccncf/busybox:1 --dry-run=client -o yaml > deployment_xyz.yml
student@node-1:~$ vim deployment_xyz.yml


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: apps/v1
kind: Deployment
metadata:
creationTimestamp: null
labels:
app: deployment-xyz
name: deployment-xyz
spec:
replicas: 1
selector:
matchLabels:
app: deployment-xyz
strategy: {}
template:
metadata:
creationTimestamp: null
labels:
app: deployment-xyz
spec:
containers:
- image: lfccncf/busybox:1
name: busybox
resources: {}
status: {}
~
~
"deployment_xyz.yml" 24L, 434C
3,1
All


Readme
Web Terminal
THE LINUX FOUNDATION
kind: Deployment
metadata:
labels:
app: deployment-xyz
name: deployment-xyz
spec:
replicas: 1
selector:
matchLabels:
app: deployment-xyz
template:
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
containers:
- image: lfccncf/busybox:1
name: logger-dev
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
3 lines yanked
27,22
Bot


Readme
Web Terminal
THE LINUX FOUNDATION
replicas: 1
selector:
matchLabels:
app: deployment-xyz
template:
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
containers:
- image: lfccncf/busybox:1
name: logger-dev
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
volumeMounts:
- name: myvol1
mountPath: /tmp/log
29,83
Bot

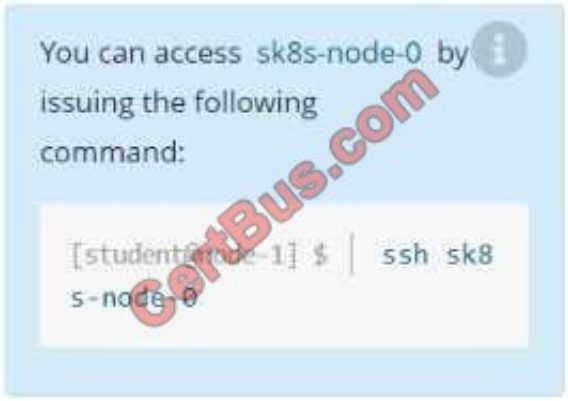

Readme
Web Terminal
THE LINUX FOUNDATION
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
- name: myvol2
configMap:
name: logconf
containers:
- image: lfccncf/busybox:1
name: logger-dev
command: ["/bin/sh","-c","while [true]; do echo 'i luv cncf' >> /tmp/log/input.log; sl
eep 10; done"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
command: ["/bin/sh","-c","tail -f /tmp/log/input.log >> /tmp/log/output.log"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- name: myvol2
mountPath: /fluentd/etc
37,33
Bot

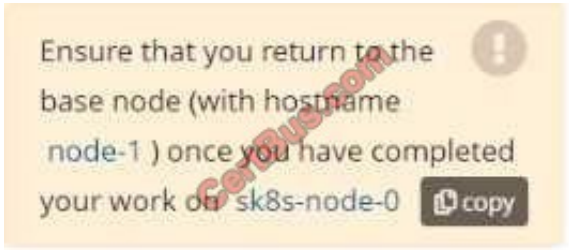

student@node-1:~$ kubectl create -f deployment_xyz.yml
deployment.apps/deployment-xyz created
student@node-1:~$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-xyz 0/1 1 0 5s
student@node-1:~$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-xyz 0/1 1 0 9s
student@node-1:~$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-xyz 1/1 1 1 12s
student@node-1:~$

QUESTION 3

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 PersistenetVolume. Create a PefsissentVolumeClaim 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 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



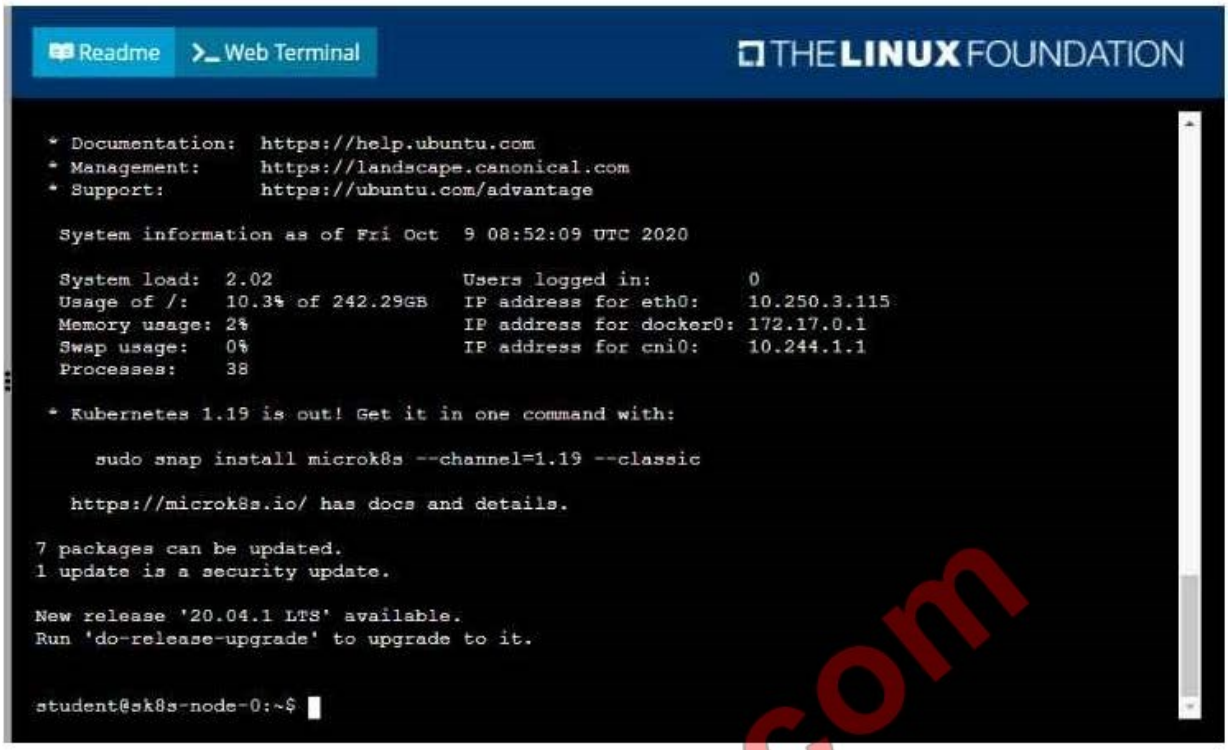


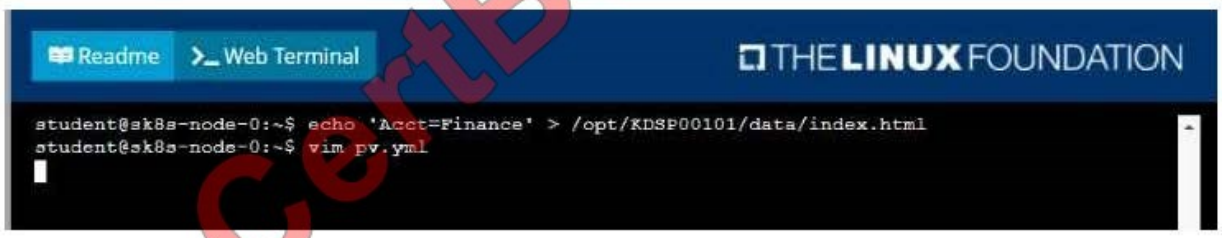
A. Please check explations

B. Place Holder

Correct Answer: AB

Solution:

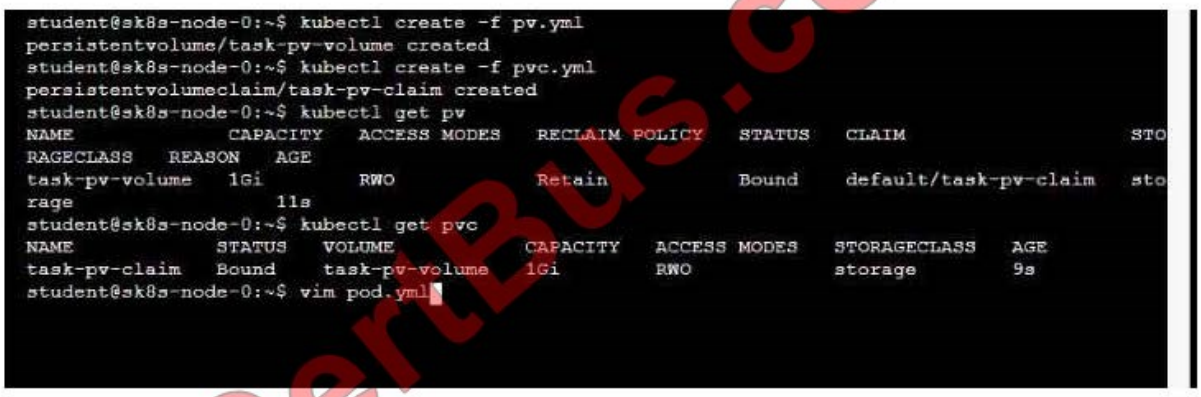

Readme
Web Terminal
THE LINUX FOUNDATION
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Fri Oct 9 08:52:09 UTC 2020
System load: 2.02
Usage of /: 10.3% of 242.29GB
Memory usage: 2%
Swap usage: 0%
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:~$

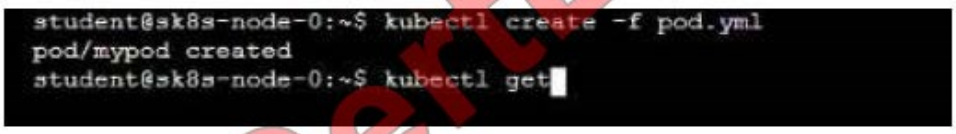

Readme
Web Terminal
THE LINUX FOUNDATION
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml

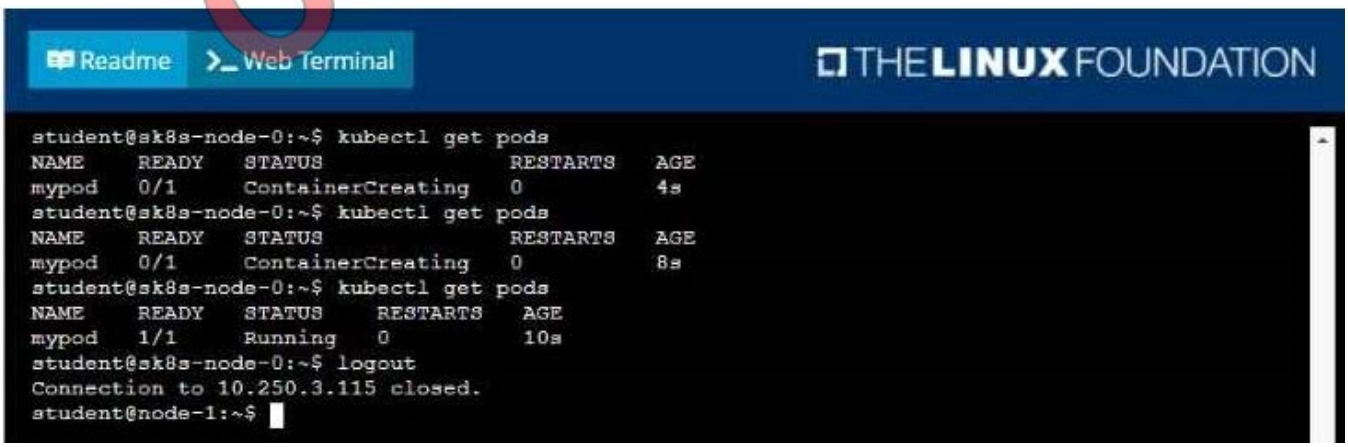

Readme
Web Terminal
THE LINUX FOUNDATION
-- INSERT --
0,1
All


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: Directory


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





QUESTION 4

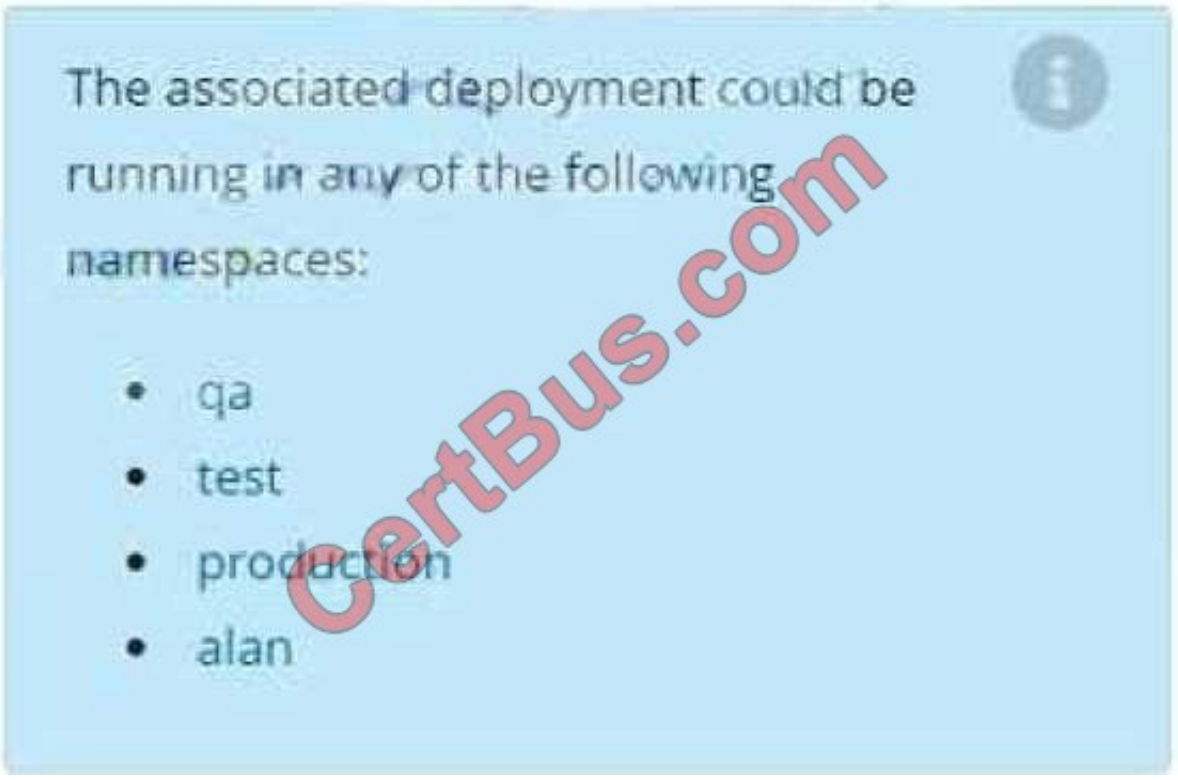
Exhibit:

Context A user has reported an aopticauon 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:

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. Please check explations

B. Place Holder

Correct Answer: AB

Solution:

Create the Pod:

kubectl create -f http://k8s.io/docs/tasks/configure-pod-container/exec-liveness.yaml Within 30 seconds,

view the Pod events:

kubectl describe pod liveness-exec

The output indicates that no liveness probes have failed yet:

FirstSeen LastSeen Count From SubobjectPath Type Reason Message --------- -------- ----- ---- -------------------- ------ ------- 24s 24s 1 {default-scheduler } Normal Scheduled Successfully assigned liveness- exec

to worker0

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/

google_containers/busybox"

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/

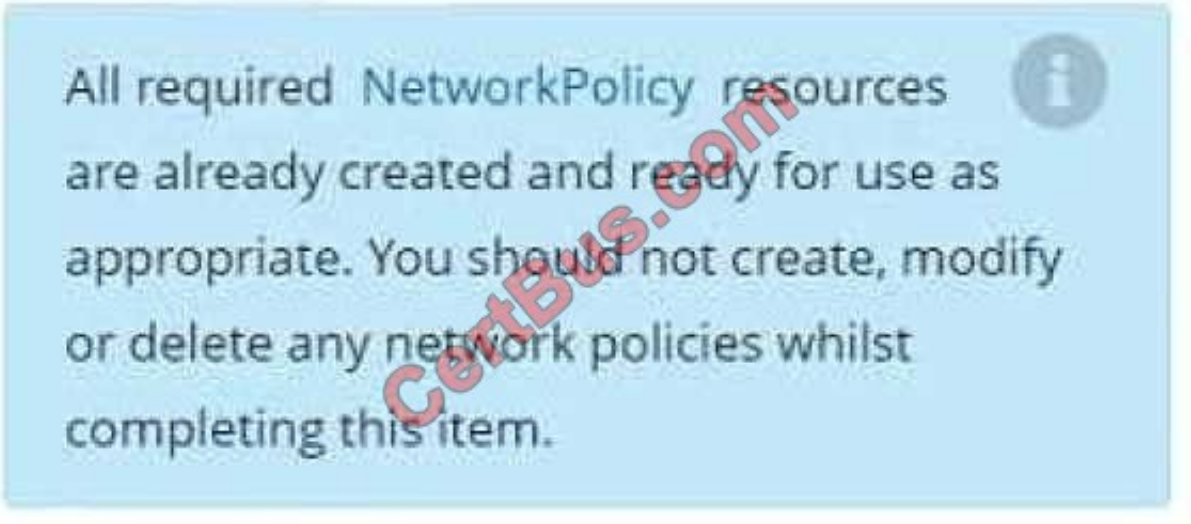
google_containers/busybox"

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined] 23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e After 35 seconds, view the Pod events again: kubectl describe pod liveness-exec At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated. FirstSeen LastSeen Count From SubobjectPath Type Reason Message --------- -------- ----- ---- ---------- ------ ------- 37s 37s 1 {default-scheduler } Normal Scheduled Successfully assigned liveness- exec to worker0 36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/ google_containers/busybox" 36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/ google_containers/busybox" 36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined] 36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e 2s 2s 1 {kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can\\'t open \\'/tmp/healthy\\': No such file or directory Wait another 30 seconds, and verify that the Container has been restarted: kubectl get pod liveness-exec The output shows that RESTARTS has been incremented: NAME READY STATUS RESTARTS AGE liveness-exec 1/1 Running 1 m

QUESTION 5

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. Please check explations

B. Place Holder

Correct Answer: AB

Pending

Latest CKAD Dumps CKAD PDF Dumps CKAD Exam Questions

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.certbus.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: