
**Vendor:**Linux Foundation

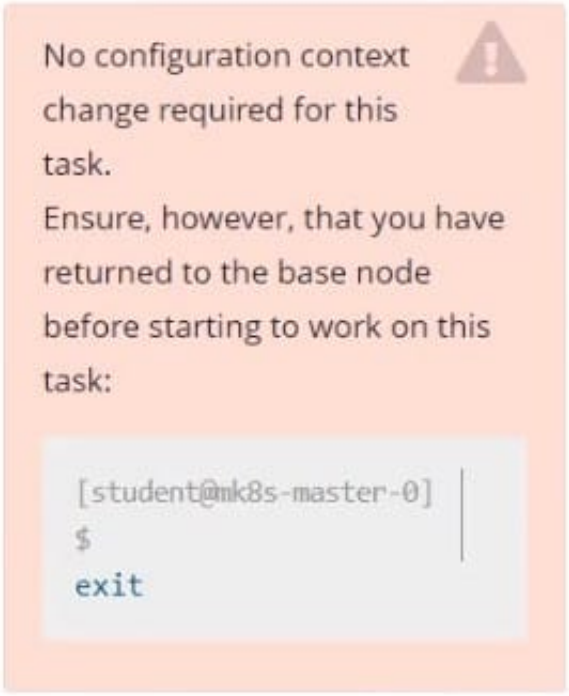
**Exam Code:**CKA

**Exam Name:**Certified Kubernetes Administrator (CKA) Program

**Version:**Demo

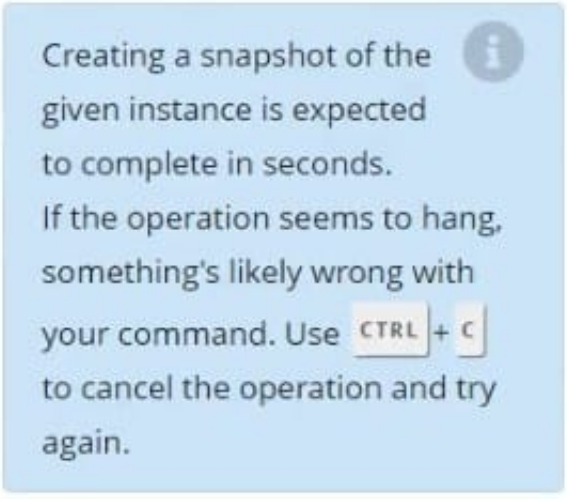
**QUESTION 1**

CORRECT TEXT




Task

First, create a snapshot of the existing etcd instance running at https://127.0.0.1:2379, saving the snapshot to /srv/data/etcd-snapshot.db.




Next, restore an existing, previous snapshot located at /var/lib/backup/etcd-snapshot-previo us.db


Correct Answer: Check the answer in explanation.

#backup ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" -- cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt -- key=/opt/KUIN000601/etcd-client.key snapshot save /etc/data/etcd-snapshot.db

#restore ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" -- cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt -- key=/opt/KUIN000601/etcd-client.key snapshot restore /var/lib/backup/etcd-snapshot- previoys.db

---

**QUESTION 2**

CORRECT TEXT


Task

Schedule a pod as follows:

1.

Name: nginx-kusc00401

2.

Image: nginx

3.

Node selector: disk=ssd

Correct Answer: Check the answer in explanation.

Solution:

#yaml apiVersion: v1 kind: Pod metadata: name: nginx-kusc00401 spec: containers:

-name: nginx image: nginx imagePullPolicy: IfNotPresent nodeSelector: disk: spinning # kubectl create -f node-select.yaml

---

**QUESTION 3**

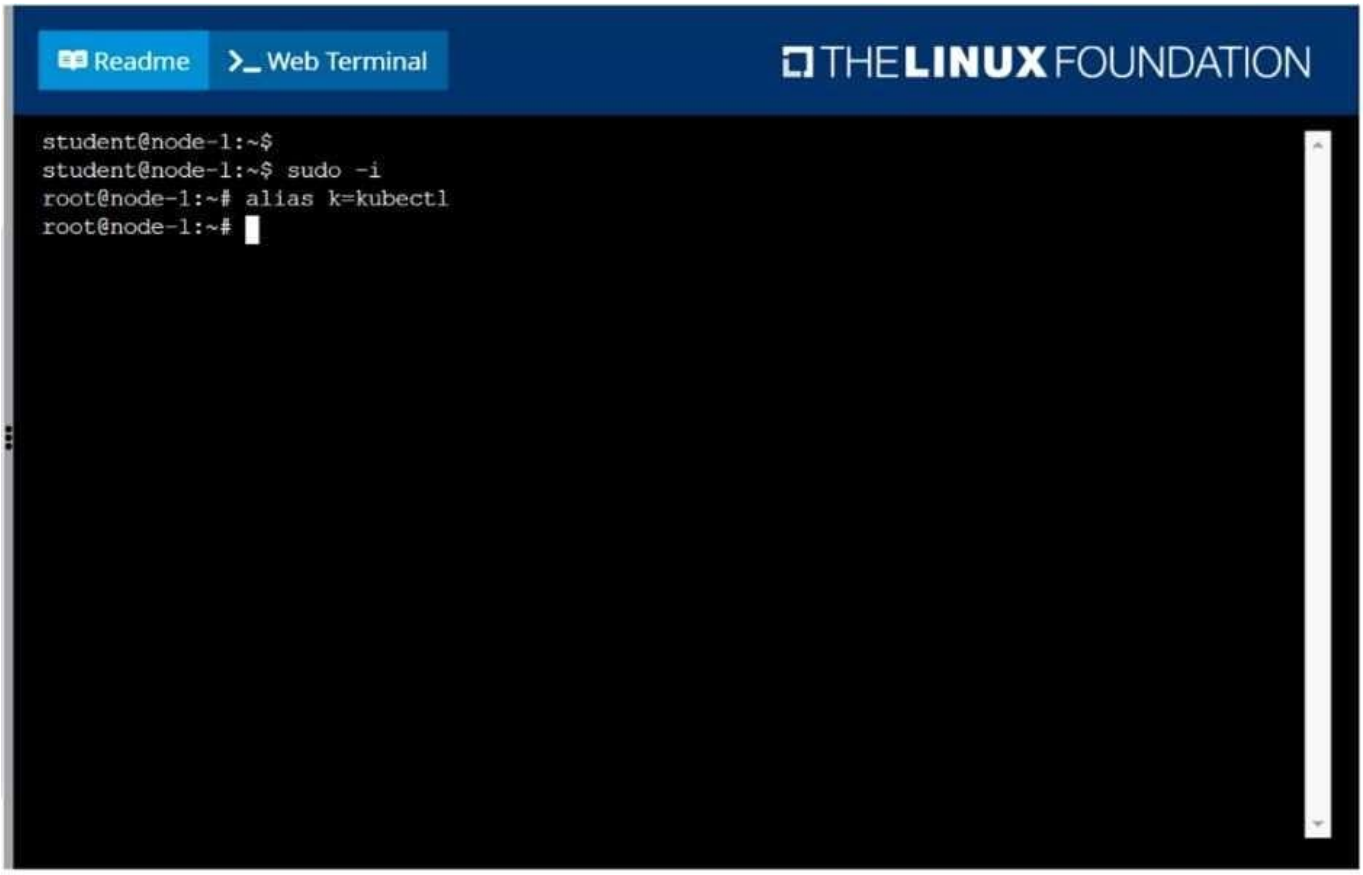
SIMULATION

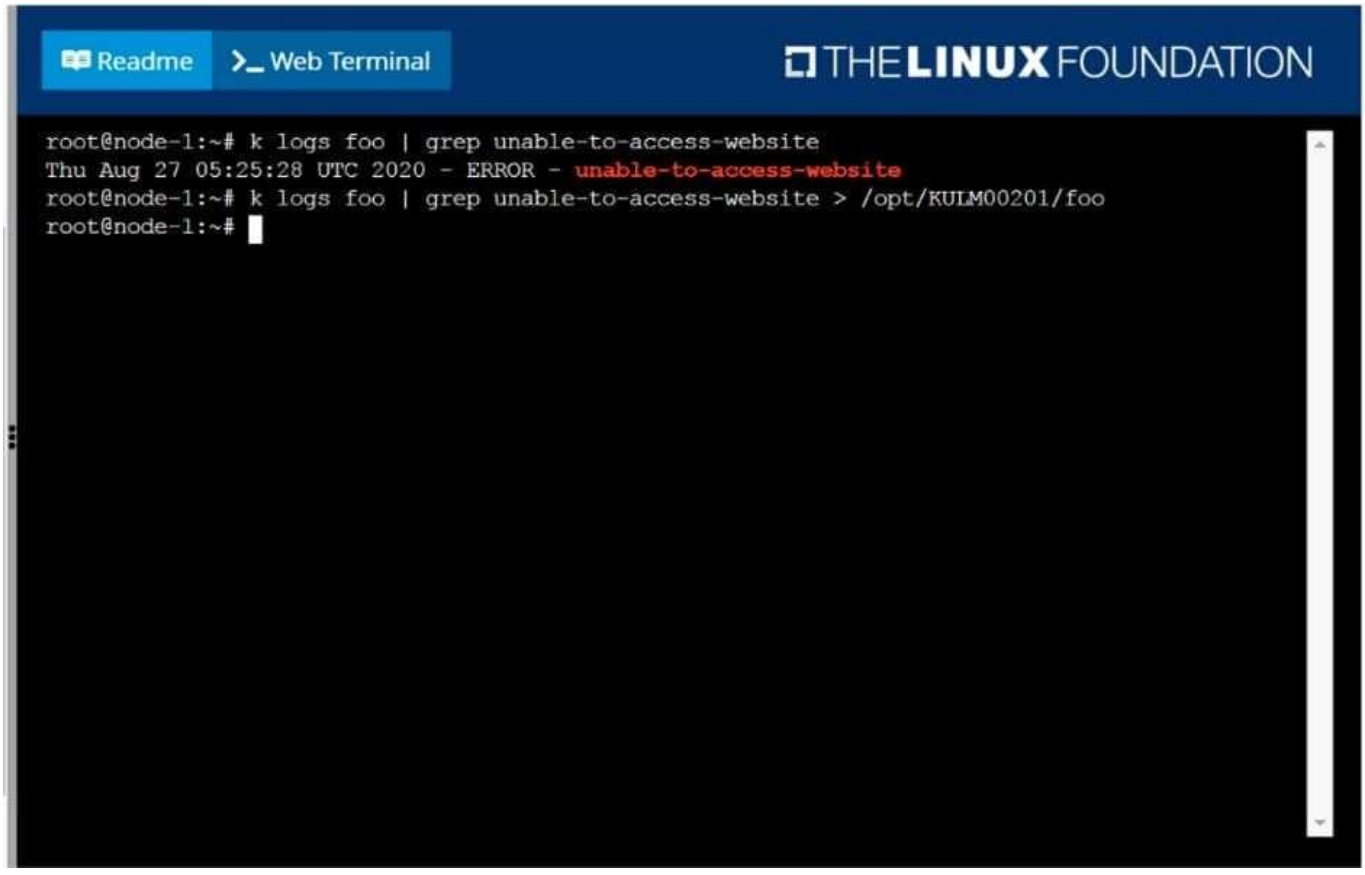
Monitor the logs of pod foo and:

1. Extract log lines corresponding to error

2. unable-to-access-website Write them to /opt/KULM00201/foo


Correct Answer: Check the answer in explanation.


Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$
student@node-1:~$ sudo -i
root@node-1:~# alias k=kubectl
root@node-1:~#


Readme
Web Terminal
THE LINUX FOUNDATION
root@node-1:~# k logs foo | grep unable-to-access-website
Thu Aug 27 05:25:28 UTC 2020 - ERROR - unable-to-access-website
root@node-1:~# k logs foo | grep unable-to-access-website > /opt/KULM00201/foo
root@node-1:~#

**QUESTION 4**

SIMULATION

Create a deployment as follows:

1.

Name: nginx-app

2.

Using container nginx with version 1.11.10-alpine

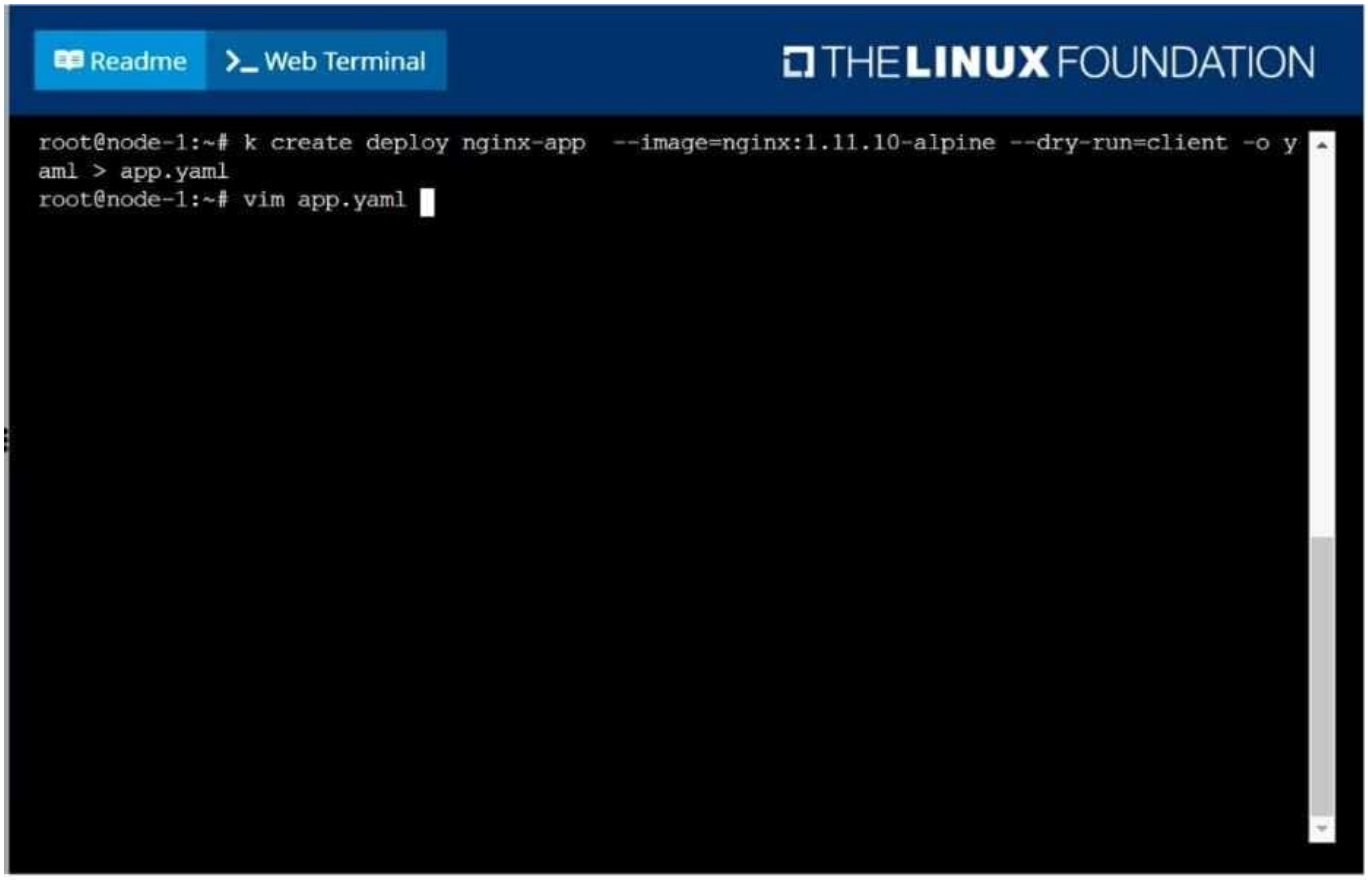
3.

The deployment should contain 3 replicas

Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update. Finally, rollback that update to the previous version 1.11.10-alpine.

Correct Answer: Check the answer in explanation.

Solution



Readme | Web Terminal

THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      labels:
        app: nginx-app
    spec:
      containers:
      - image: nginx:1.11.10-alpine
        name: nginx
~
~
~
~
~
~
~
'app.yaml
```

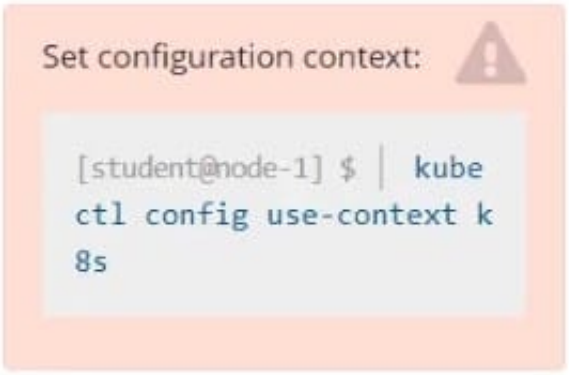
Readme | Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
root@node-1:~# k create -f app.yaml
deployment.apps/nginx-app created
root@node-1:~#
root@node-1:~#
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine  --record
deployment.apps/nginx-app image updated
root@node-1:~# k rollout undo deploy nginx-app
deployment.apps/nginx-app rolled back
root@node-1:~#
```

**QUESTION 5**

CORRECT TEXT




Task

From the pod label name=cpu-utilizer, find pods running high CPU workloads and write the name of the pod consuming most CPU to the file /opt/KUTR00401/KUTR00401.txt (which already exists).

Correct Answer: Check the answer in explanation.

kubectl top -l name=cpu-user -A echo \\'pod name\\' >> /opt/KUT00401/KUT00401.txt

---

**QUESTION 6**

Create a pod with environment variables as var1=value1.Check the environment variable in pod

Correct Answer: Check the answer in explanation.

Solution

kubectl run nginx --image=nginx --restart=Never --env=var1=value1 # then kubectl exec -it nginx -- env # or kubectl exec -it nginx -- sh -c \\'echo $var1\\' # or kubectl describe po nginx | grep value1

---

**QUESTION 7**

CORRECT TEXT

Set configuration context:

```
[student@node-1] $ kube
ctl config use-context m
k8s
```

Task

Given an existing Kubernetes cluster running version 1.20.0, upgrade all of the Kubernetes control plane and node components on the master node only to version 1.20.1.

Be sure to drain the master node before upgrading it and uncordon it after the upgrade.

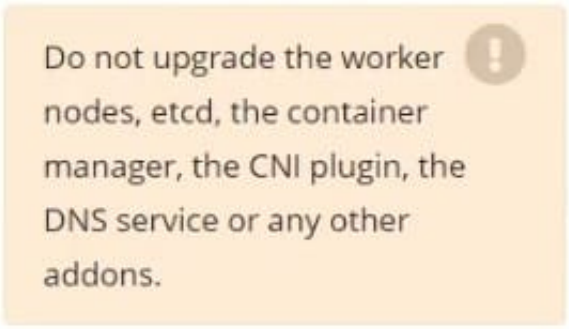
You can ssh to the master node using:

```
[student@node-1] $ ssh
mk8s-master-0
```

You can assume elevated privileges on the master node with the following command:

```
[student@mk8s-master-0]
$
sudo -i
```

You are also expected to upgrade kubelet and kubectl on the master node.




Correct Answer:

[student@node-1] > ssh ek8s kubectl cordon k8s-master kubectl drain k8s-master --delete-local-data --ignore-daemonsets --force apt-get install kubeadm=1.20.1-00 kubelet=1.20.1-00 kubectl=1.20.1-00 --disableexcludes=kubernetes kubeadm upgrade apply 1.20.1 --etcd-upgrade=false systemctl daemon-reload systemctl restart kubelet kubectl uncordon k8s-master

---

**QUESTION 8**

CORRECT TEXT


Task

Create a persistent volume with name app-data , of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-data .

Correct Answer:

#vi pv.yaml apiVersion: v1 kind: PersistentVolume metadata: name: app-config spec: capacity: storage: 1Gi accessModes:

-ReadOnlyMany hostPath: path: /srv/app-config # kubectl create -f pv.yaml

---

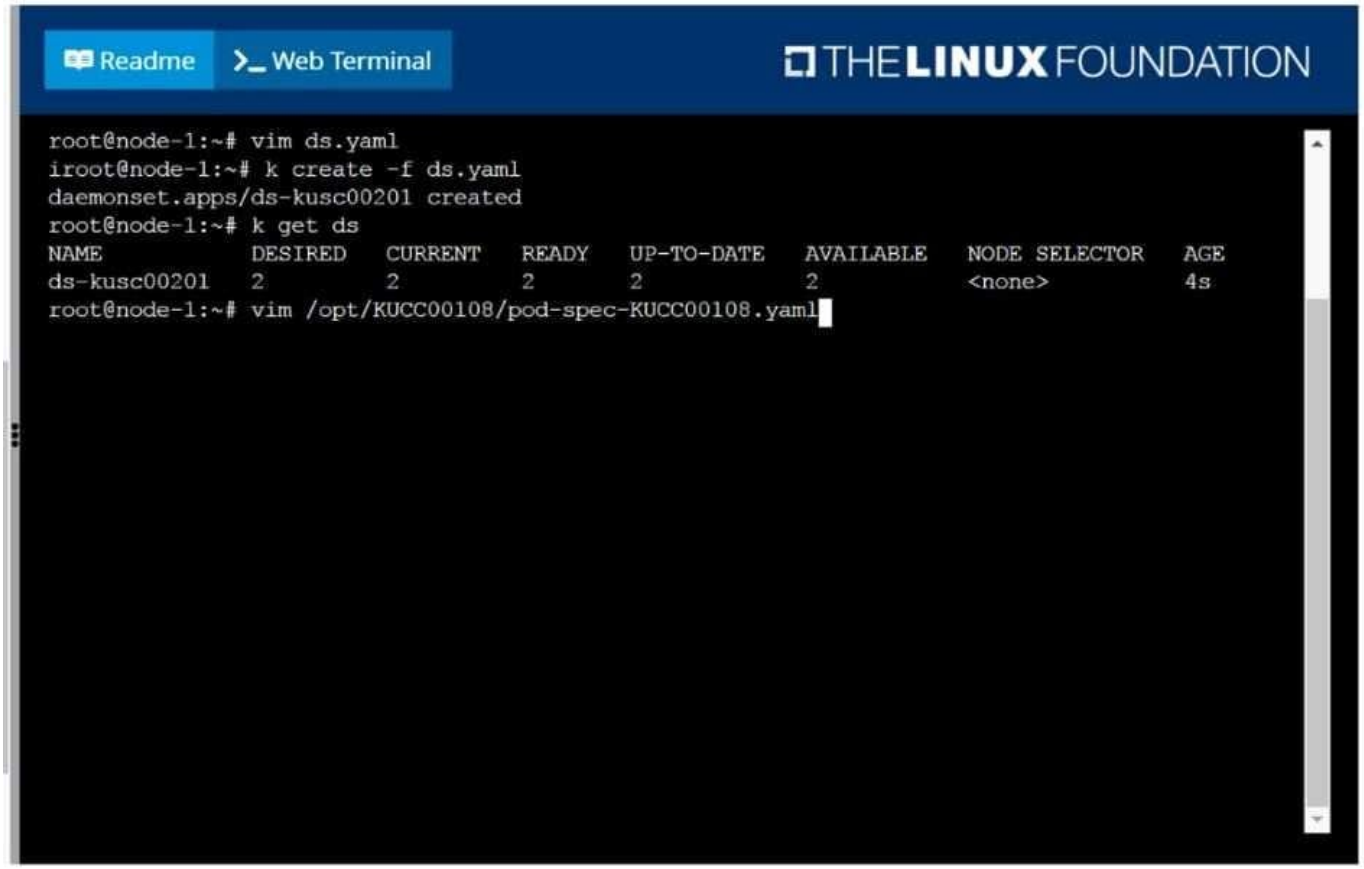
**QUESTION 9**

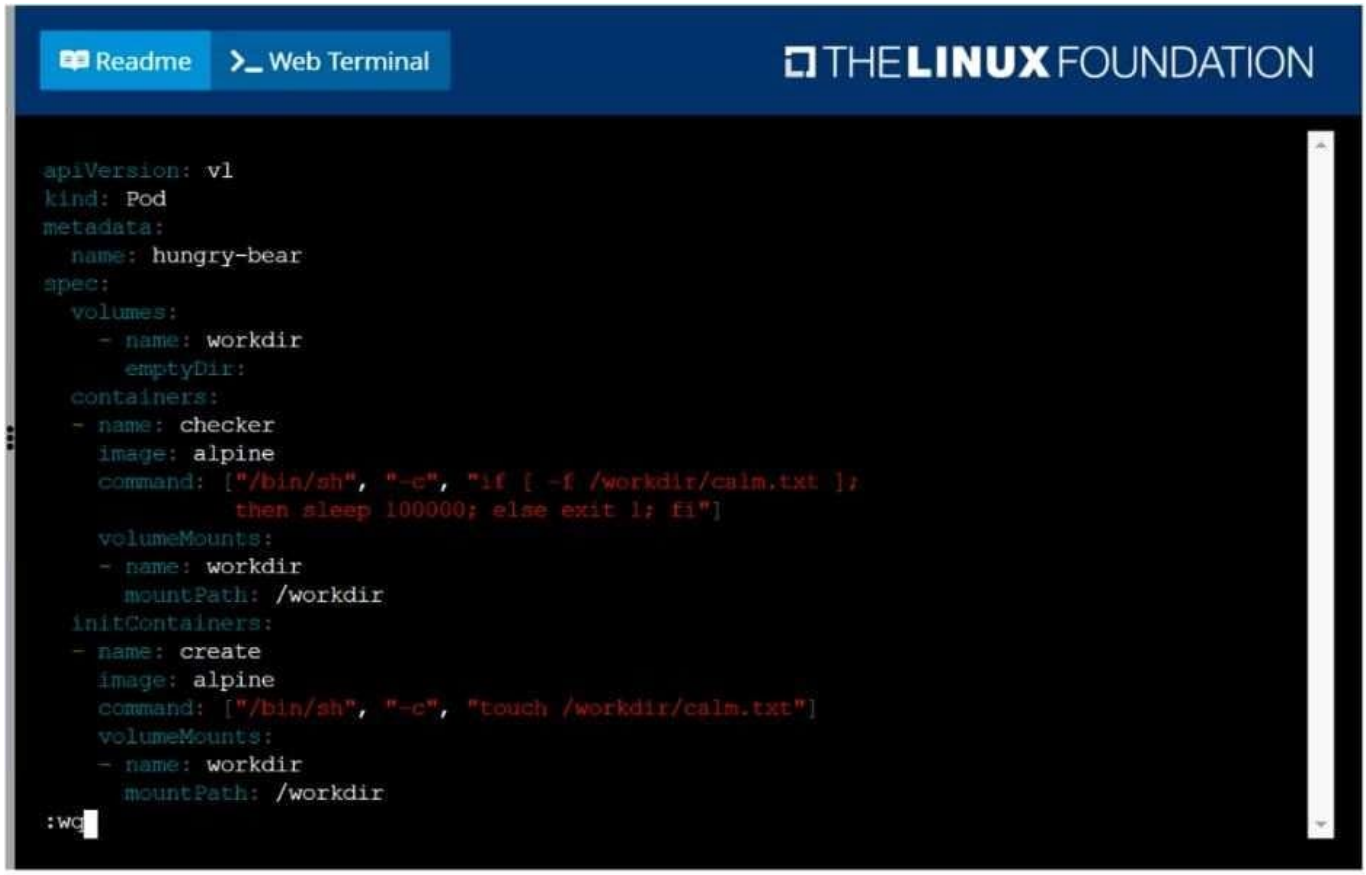
SIMULATION

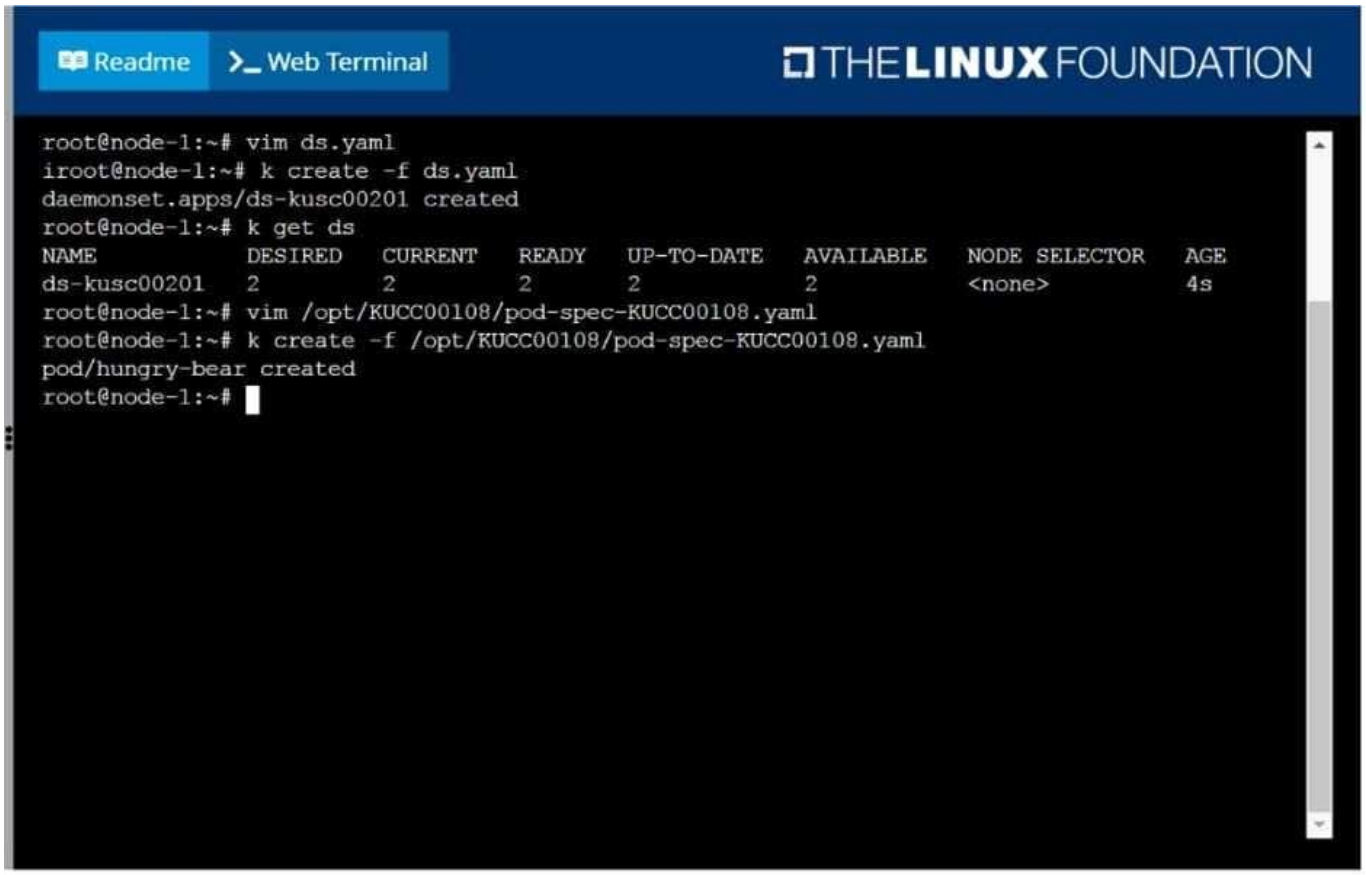
Perform the following tasks: Add an init container to hungry-bear (which has been defined in spec file

/opt/KUCC00108/pod- spec-KUCC00108.yaml) The init container should create an empty file named /workdir/calm.txt If /workdir/calm.txt is not detected, the pod should exit Once the spec file has been updated with the init container definition, the pod should be created

Correct Answer: Check the answer in explanation.




Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
name: hungry-bear
spec:
volumes:
- name: workdir
emptyDir:
containers:
- name: checker
image: alpine
command: ["/bin/sh", "-c", "if [ -f /workdir/calm.txt ];
then sleep 100000; else exit 1; fi"]
volumeMounts:
- name: workdir
mountPath: /workdir
initContainers:
- name: create
image: alpine
command: ["/bin/sh", "-c", "touch /workdir/calm.txt"]
volumeMounts:
- name: workdir
mountPath: /workdir
:wq


Readme
Web Terminal
THE LINUX FOUNDATION
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE
ds-kusc00201 2 2 2 2 2 <none> 4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~#

**QUESTION 10**

Create a busybox pod that runs the command "env" and save the output to "envpod" file

Correct Answer: Check the answer in explanation.

Solution

kubectl run busybox --image=busybox --restart=Never -rm -it -- env>; envpod.yaml

---

**QUESTION 11**

List the nginx pod with custom columns POD_NAME and POD_STATUS

Correct Answer: Check the answer in explanation.

Solution

kubectl get po -o=custom-columns="POD_NAME:.metadata.name, POD_STATUS:.status.containerStatuses[].state"

---

**QUESTION 12**

Create 2 nginx image pods in which one of them is labelled with env=prod and another one labelled with env=dev and verify the same.

Correct Answer: Check the answer in explanation.

Solution

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml > nginx-prodpod.yaml Now, edit nginx-prod-pod.yaml file and remove entries like "creationTimestamp:

null" "dnsPolicy: ClusterFirst"

vim nginx-prod-pod.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

env: prod

name: nginx-prod

spec:

containers:

-

image: nginx name: nginx-prod restartPolicy: Always # kubectl create -f nginx-prod-pod.yaml kubectl run --generator=run-pod/v1 --image=nginx -labels=env=dev nginx-dev --dry-run -o yaml > nginx-dev-pod.yaml apiVersion: v1 kind: Pod metadata: labels: env: dev name: nginx-dev spec: containers:

-

image: nginx name: nginx-dev restartPolicy: Always # kubectl create -f nginx-prod-dev.yaml Verify : kubectl get po --show-labels kubectl get po -l env=prod kubectl get po -l env=dev