



# Linux-Foundation

## Exam Questions CKA

Certified Kubernetes Administrator (CKA) 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**

Create a pod that echo "hello world" and then exists. Have the pod deleted automatically when it's completed

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
kubectl run busybox --image=busybox -it --rm --restart=Never -
/bin/sh -c 'echo hello world'
kubectl get po # You shouldn't see pod with the name "busybox"
```

**NEW QUESTION 2**

Monitor the logs of pod foo and:

- > Extract log lines corresponding to error unable-to-access-website
- > Write them to /opt/KULM00201/foo



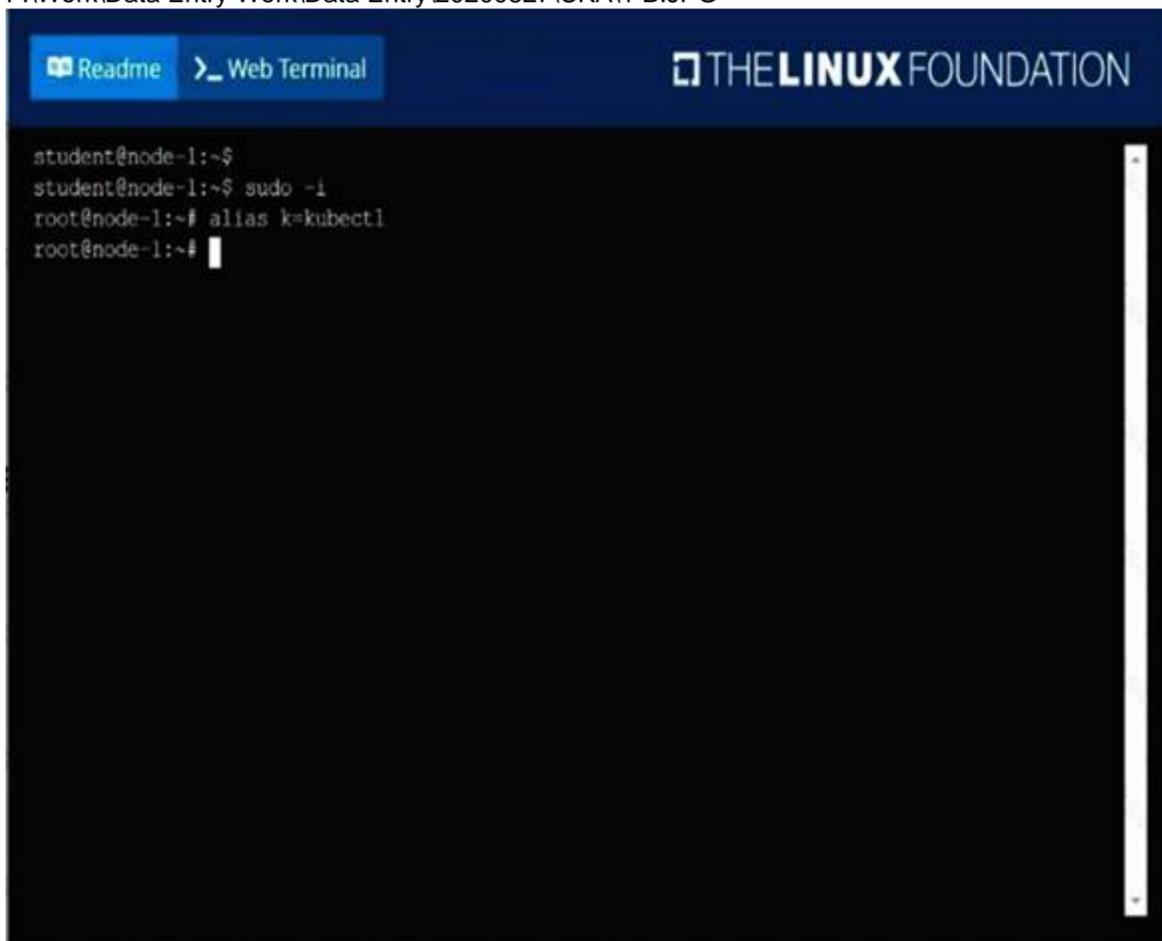
- A. Mastered
- B. Not Mastered

**Answer:** A

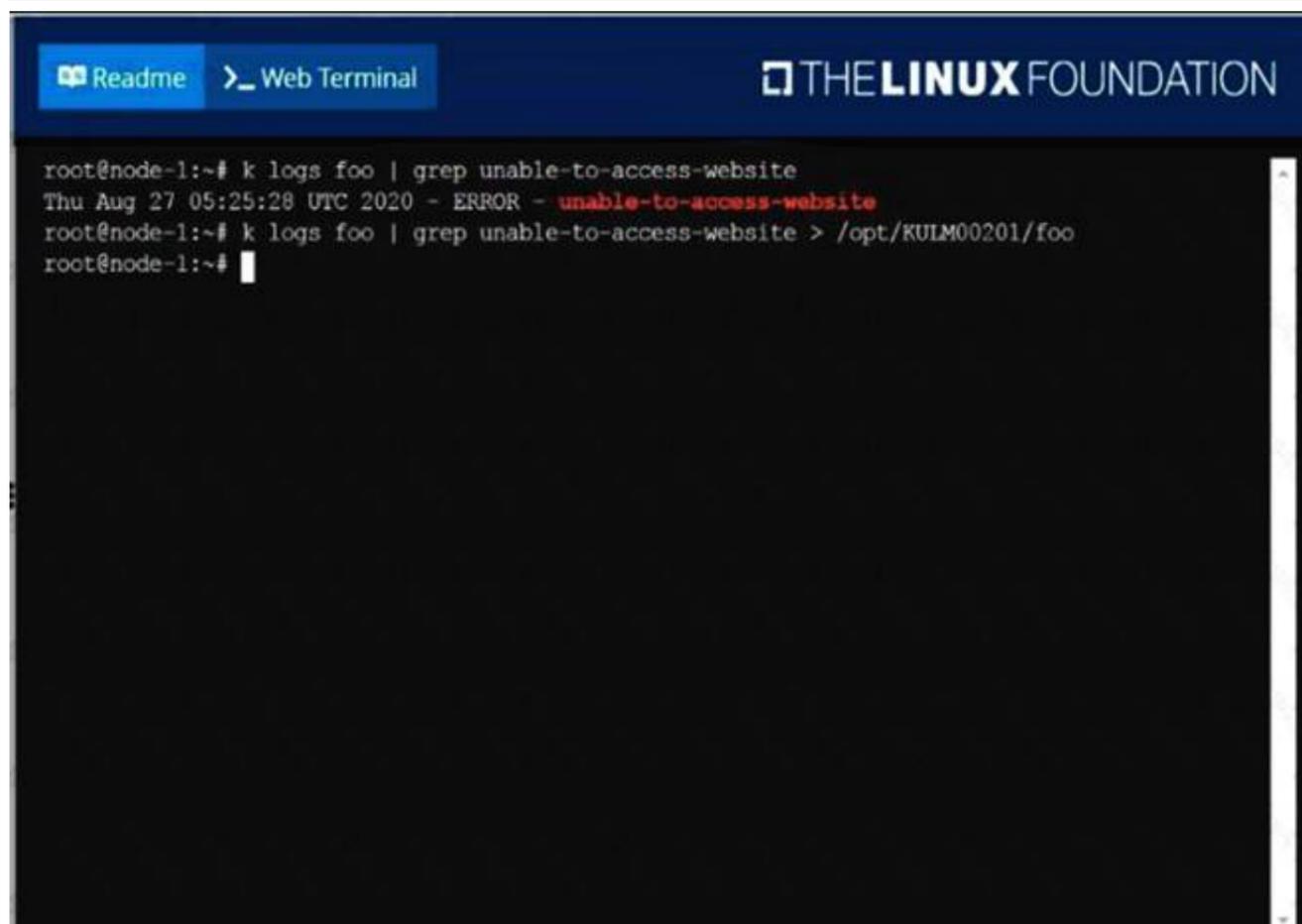
**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA1 B.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA1 C.JPG



```

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:~#

```

**NEW QUESTION 3**

Create a pod as follows:

- > Name:mongo
- > Using Image:mongo
- > In anew Kubernetes namespacenamed:my-website

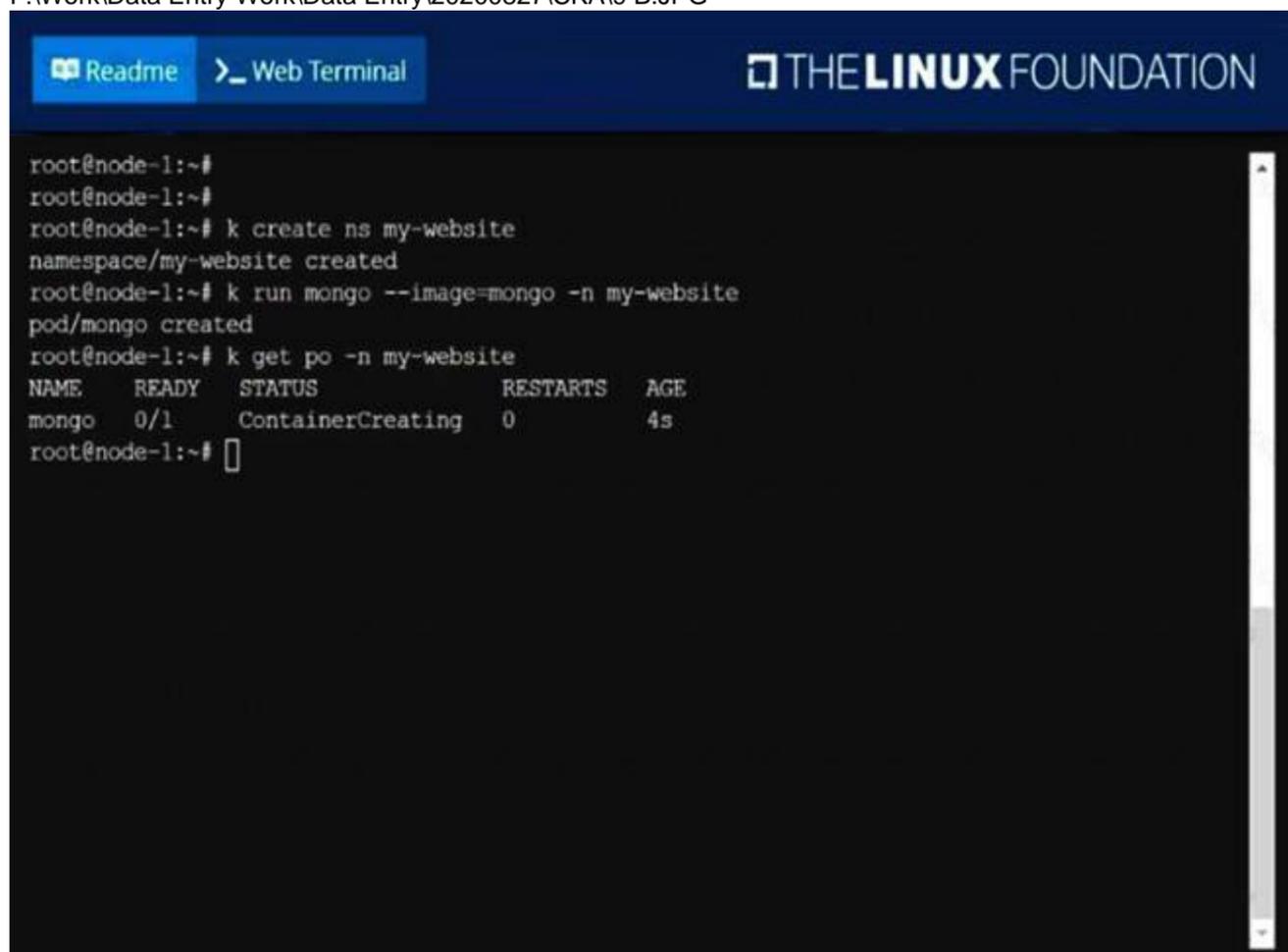
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\9 B.JPG



```

Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~#
root@node-1:~#
root@node-1:~# k create ns my-website
namespace/my-website created
root@node-1:~# k run mongo --image=mongo -n my-website
pod/mongo created
root@node-1:~# k get po -n my-website
NAME     READY   STATUS             RESTARTS   AGE
mongo    0/1     ContainerCreating  0           4s
root@node-1:~#

```

**NEW QUESTION 4**

Create a deployment as follows:

- > Name:nginx-app
- > Using containernginxwithversion 1.11.10-alpine

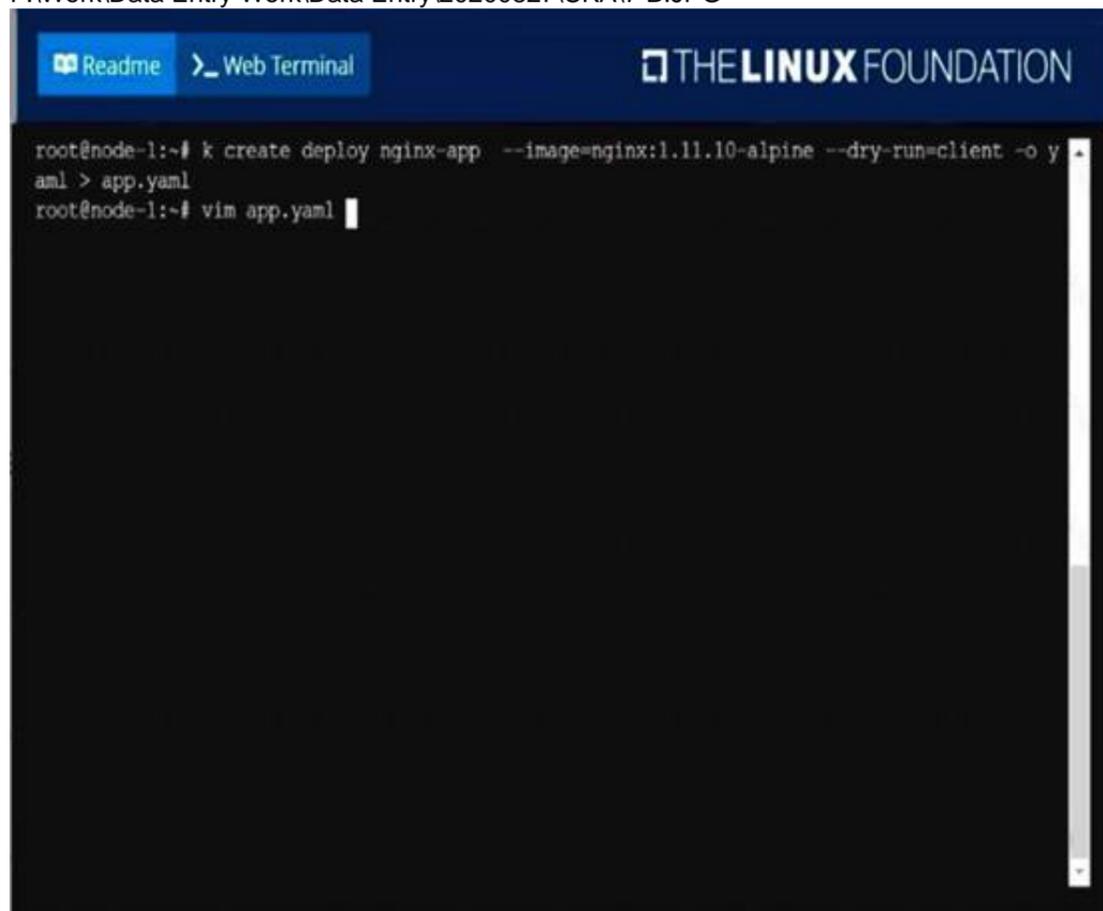
> The deployment should contain 3 replicas  
 Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update.  
 Finally, rollback that update to the previous version 1.11.10-alpine.

- A. Mastered
- B. Not Mastered

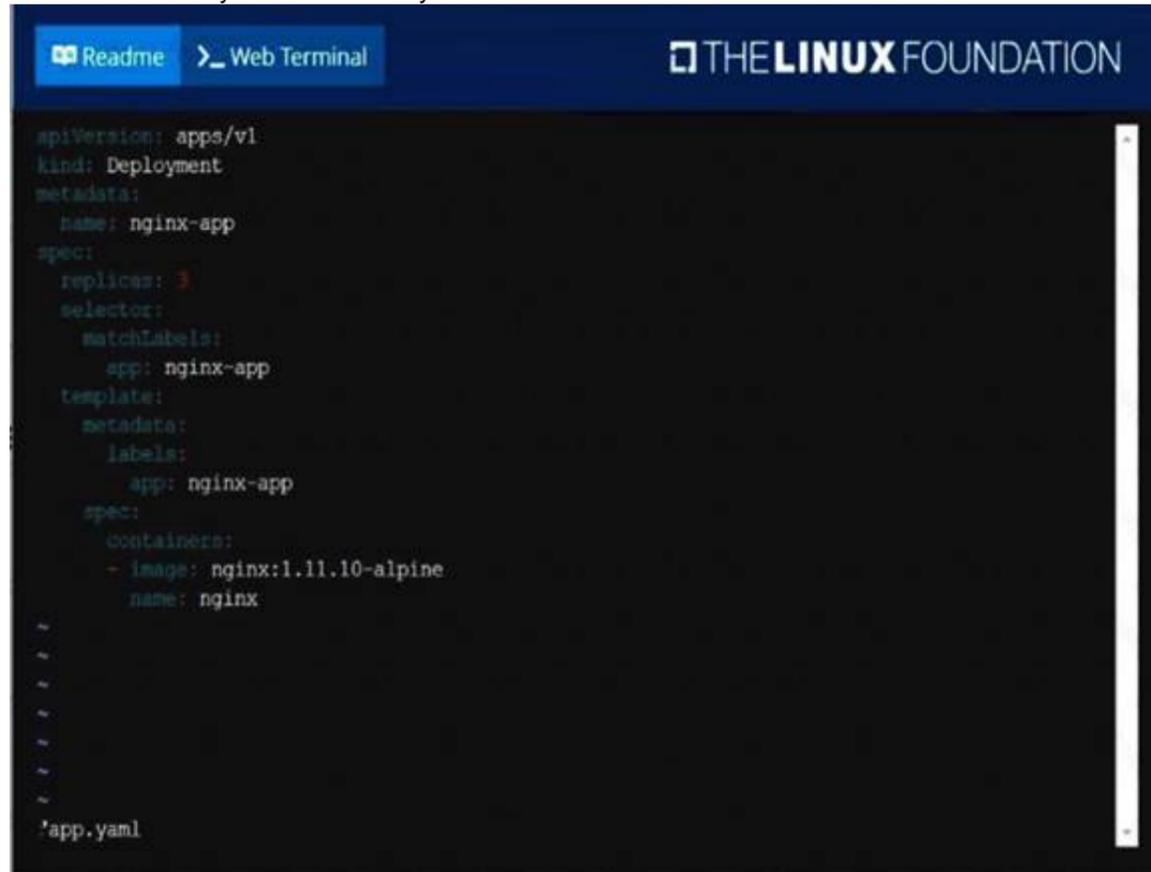
**Answer:** A

**Explanation:**

solution  
 F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 B.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 D.JPG

```
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:~#
```

#### NEW QUESTION 5

List `nginx-dev` and `nginx-prod` pod and delete those pods

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

```
kubectl get pods -o wide
kubectl delete po nginx-dev kubectl delete po nginx-prod
```

#### NEW QUESTION 6

Set the node named `ek8s-node-1` as unavailable and reschedule all the pods running on it.

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

```
solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\19 B.JPG
```

[Readme](#)
>
[Web Terminal](#)
THE LINUX FOUNDATION

```

root@node-1:~# kubectl config use-context ek8s
Switched to context "ek8s".
root@node-1:~# k drain ek8s-node-1 --ignore-daemonsets --delete-local-data --force
node/ek8s-node-1 cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-amd64-qj7w8, kube-system/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zkj
evicting pod kube-system/metrics-server-64b57fd654-cktk5

```

#### NEW QUESTION 7

Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadWriteMany. The type of volume is hostPath and its location is /srv/app-data.

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

solution  
 Persistent Volume

A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a cluster-level resource like nodes, which don't belong to any namespace. It is provisioned by the administrator and has a particular file size. This way, a developer deploying their app on Kubernetes need not know the underlying infrastructure. When the developer needs a certain amount of persistent storage for their application, the system administrator configures the cluster so that they consume the PersistentVolume provisioned in an easy way.

Creating PersistentVolume

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: app-data
spec:
  capacity: # defines the capacity of PV we are creating
  storage: 2Gi # the amount of storage we are trying to claim
  accessModes: # defines the rights of the volume we are creating
  - ReadWriteMany
  hostPath:
    path: "/srv/app-data" # path to which we are creating the volume
```

Challenge

> Create a Persistent Volume named app-data, with access mode ReadWriteMany, storage class name shared, 2Gi of storage capacity and the host path /srv/app-data.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /srv/app-data
  storageClassName: share
```

"app-data.yaml" 12L, 194C

\* 2. Save the file and create the persistent volume. Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl create -f pv.yaml
persistentvolume/pv created
```

\* 3. View the persistent volume.

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
app-data	2Gi	RWX	Retain	Available		shared		31s

> Our persistent volume status is available meaning it is available and it has not been mounted yet. This status will change when we mount the persistentVolume to a persistentVolumeClaim.

PersistentVolumeClaim

In a real ecosystem, a system admin will create the PersistentVolume then a developer will create a PersistentVolumeClaim which will be referenced in a pod. A PersistentVolumeClaim is created by specifying the minimum size and the access mode they require from the persistentVolume.

Challenge

> Create a Persistent Volume Claim that requests the Persistent Volume we had created above. The claim should request 2Gi. Ensure that the Persistent Volume Claim has the same storageClassName as the persistentVolume you had previously created.

kind: PersistentVolumeClaim

apiVersion: v1

metadata: name: app-data

spec: accessModes: - ReadWriteMany

resources: requests: storage: 2Gi storageClassName: shared

\* 2. Save and create the pvc

```
njerry191@cloudshell:~(extreme-clone-2654111)$ kubectl create -f app-data.yaml persistentvolumeclaim/app-data created
```

\* 3. View the pvc Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
pv	Bound	pv	512m	RWX	shared

\* 4. Let's see what has changed in the pv we had initially created.

Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
pv	512m	RWX	Retain	Bound	default/pv	shared		16m

Our status has now changed from available to bound.

\* 5. Create a new pod named myapp with image nginx that will be used to Mount the Persistent Volume Claim with the path /var/app/config.

Mounting a Claim

apiVersion: v1

kind: Pod

metadata: creationTimestamp: null name: app-data

spec: volumes: - name: config persistentVolumeClaim: claimName: app-data

containers: - image: nginx name: app volumeMounts: - mountPath: "/srv/app-data" name: config

### NEW QUESTION 8

List all persistent volumes sorted by capacity, saving the full kubectl output to

/opt/KUCC00102/volume\_list. Use kubectl's own functionality for sorting the output, and do not manipulate it any further.

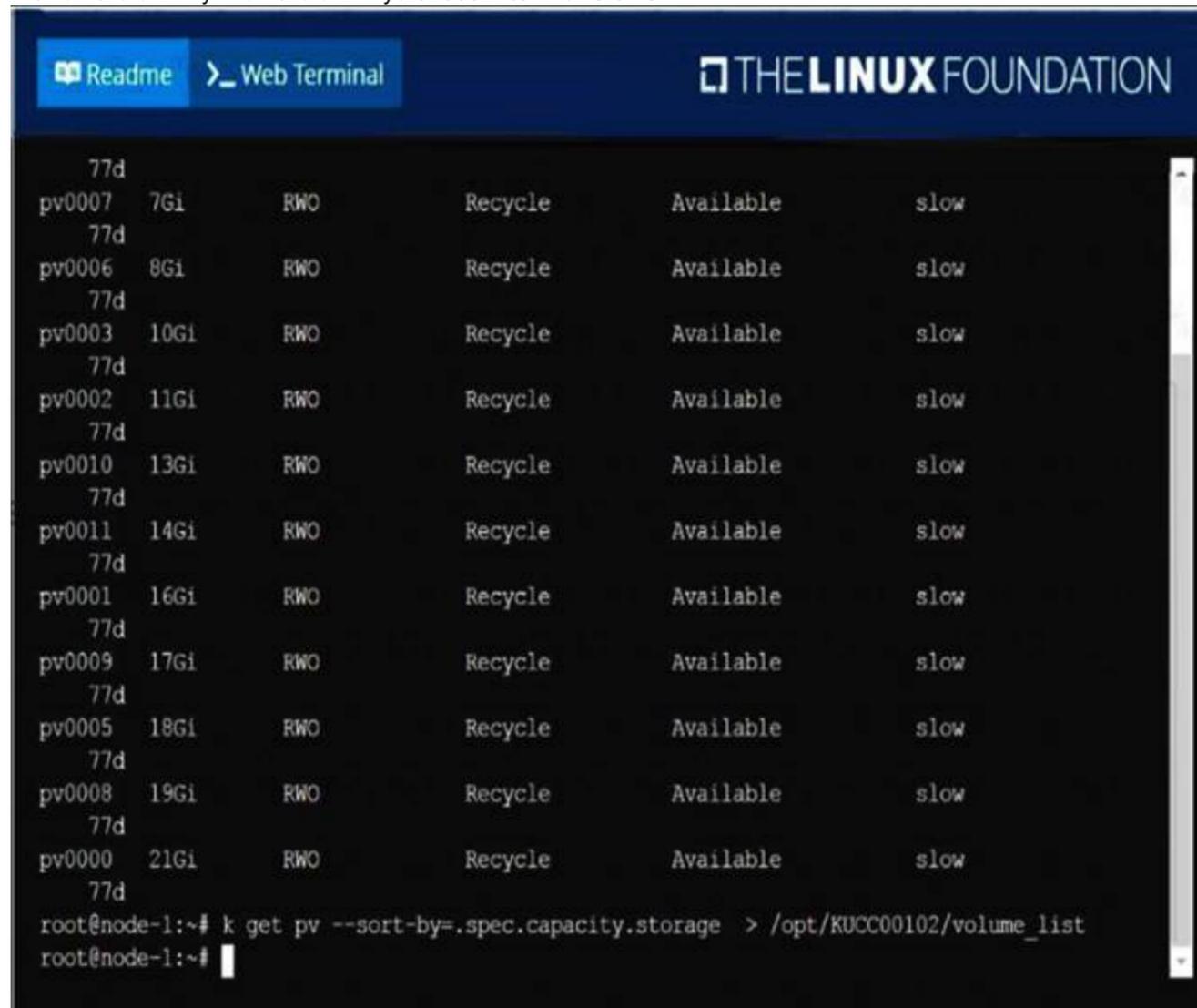
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\2 C.JPG



```

77d
pv0007 7Gi      RWO      Recycle   Available slow
77d
pv0006 8Gi      RWO      Recycle   Available slow
77d
pv0003 10Gi     RWO      Recycle   Available slow
77d
pv0002 11Gi     RWO      Recycle   Available slow
77d
pv0010 13Gi     RWO      Recycle   Available slow
77d
pv0011 14Gi     RWO      Recycle   Available slow
77d
pv0001 16Gi     RWO      Recycle   Available slow
77d
pv0009 17Gi     RWO      Recycle   Available slow
77d
pv0005 18Gi     RWO      Recycle   Available slow
77d
pv0008 19Gi     RWO      Recycle   Available slow
77d
pv0000 21Gi     RWO      Recycle   Available slow
77d
root@node-1:~# k get pv --sort-by=.spec.capacity.storage > /opt/KUCC00102/volume_list
root@node-1:~#

```

**NEW QUESTION 9**

Create a pod as follows:

- > Name:non-persistent-redis
- > container Image:redis
- > Volume with name:cache-control
- > Mount path:/data/redis

The pod should launch in the staging namespace and the volume must not be persistent.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

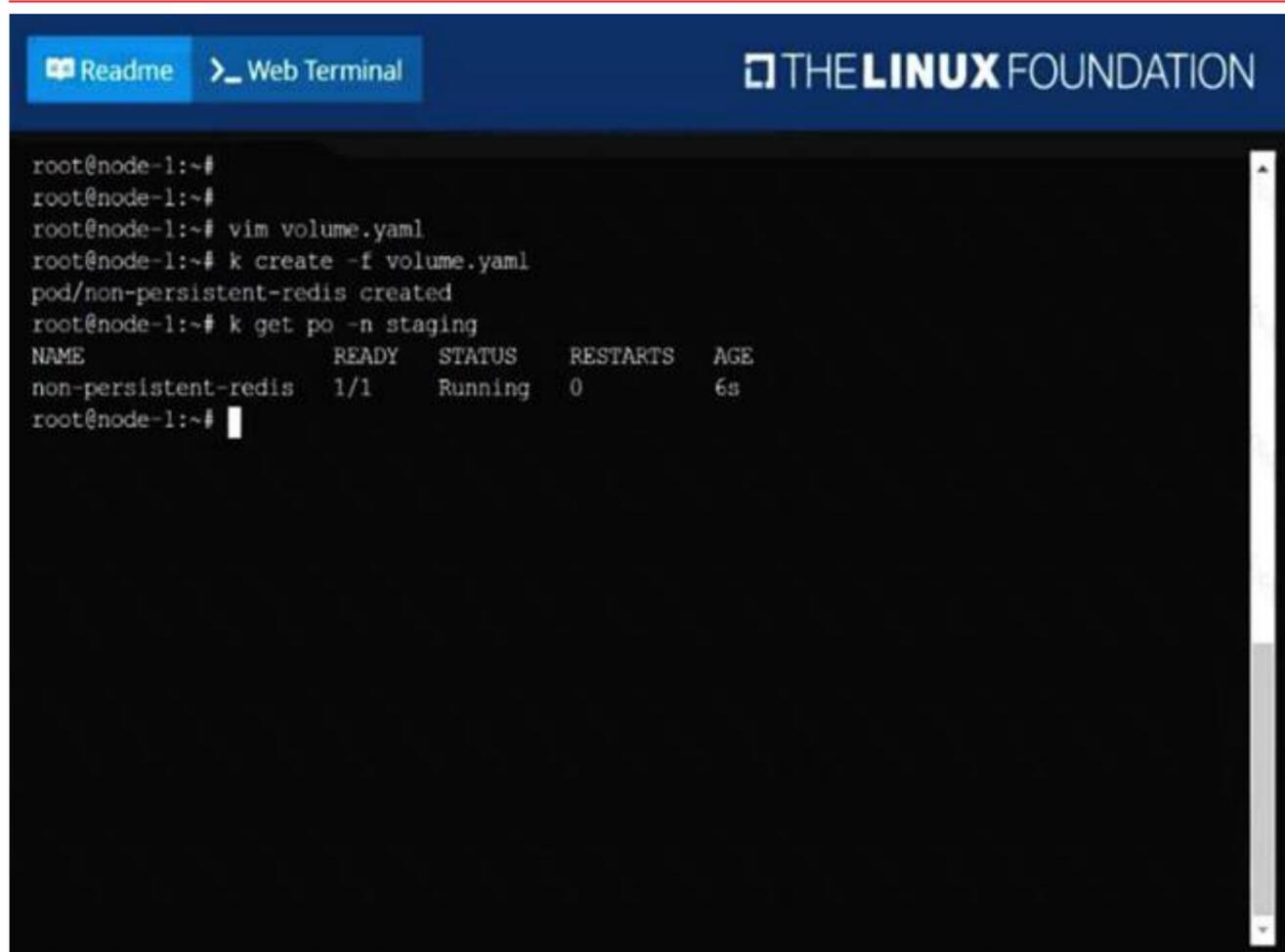
F:\Work\Data Entry Work\Data Entry\20200827\CKA\13 B.JPG

```
Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\13 C.JPG

```
Readme Web Terminal THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
  name: non-persistent-redis
  namespace: staging
spec:
  containers:
  - name: redis
    image: redis
    volumeMounts:
    - name: cache-control
      mountPath: /data/redis
  volumes:
  - name: cache-control
    emptyDir: {}
:~
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\13 D.JPG



```

root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml
root@node-1:~# k create -f volume.yaml
pod/non-persistent-redis created
root@node-1:~# k get po -n staging
NAME                READY   STATUS    RESTARTS   AGE
non-persistent-redis 1/1     Running   0           6s
root@node-1:~#

```

**NEW QUESTION 10**

Create a Kubernetes secret as follows:

- > Name: super-secret
- > password: bob

Create a pod named pod-secrets-via-file, using the redis image, which mounts a secret named super-secret at /secrets.

Create a second pod named pod-secrets-via-env, using the redis image, which exports password as CONFIDENTIAL

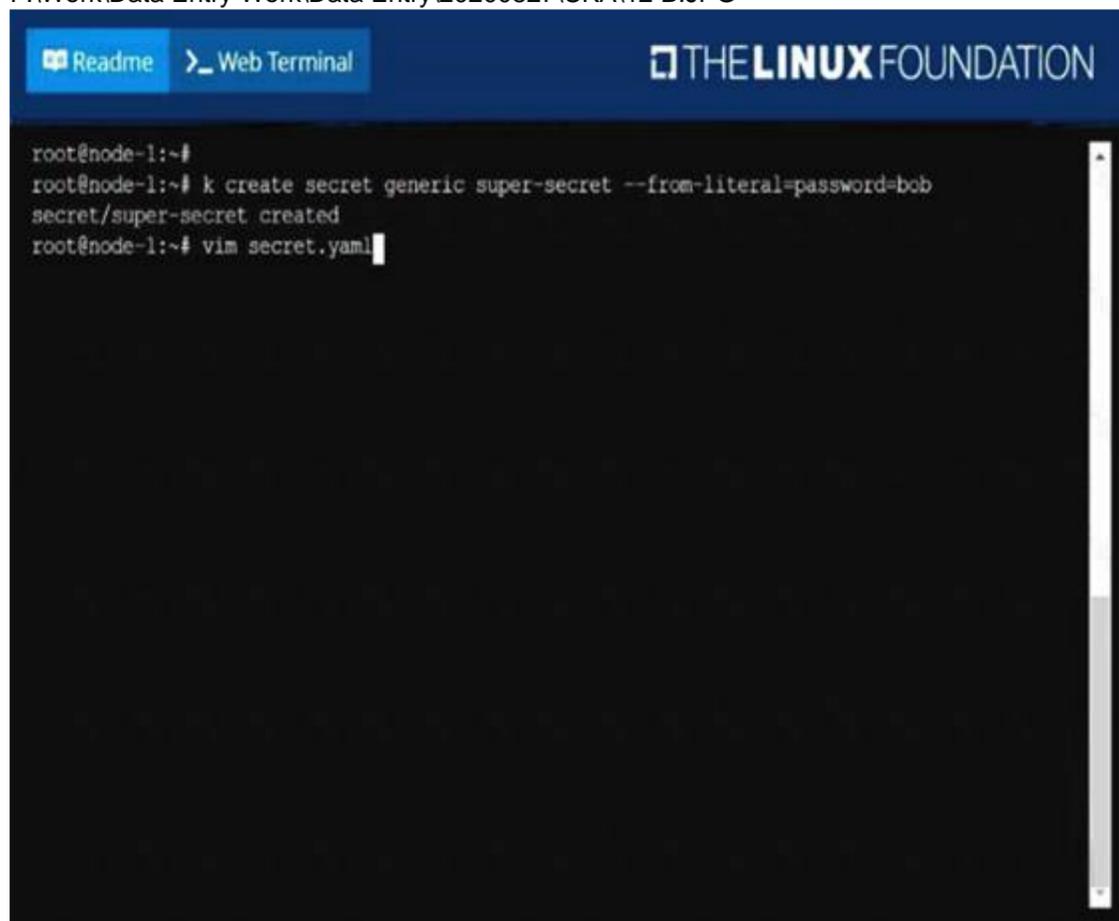
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\12 B.JPG



```

root@node-1:~#
root@node-1:~# k create secret generic super-secret --from-literal=password=bob
secret/super-secret created
root@node-1:~# vim secret.yaml

```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\12 C.JPG

```

Readme Web Terminal THE LINUX FOUNDATION

apiVersion: v1
kind: Pod
metadata:
  name: pod-secrets-via-file
spec:
  containers:
  - name: redis
    image: redis
  volumeMounts:
  - name: foo
    mountPath: "/secrets"
volumes:
- name: foo
  secret:
    secretName: super-secret

```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\12 D.JPG

```

Readme Web Terminal THE LINUX FOUNDATION

root@node-1:~# k create -f secret.yaml
pod/pod-secrets-via-file created
root@node-1:~# vim secret1.yaml
root@node-1:~# k create -f secret1.yaml
pod/pod-secrets-via-env created
root@node-1:~# k get po
NAME                                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se                 1/1     Running   0           6h25m
cpu-utilizer-ab2d3s                 1/1     Running   0           6h25m
cpu-utilizer-kipb9a                 1/1     Running   0           6h25m
ds-kusc00201-2r2k9                  1/1     Running   0           40m
ds-kusc00201-hzm9q                  1/1     Running   0           40m
foo                                  1/1     Running   0           6h28m
front-end                            1/1     Running   0           6h27m
hungry-bear                          1/1     Running   0           36m
kucc8                                 3/3     Running   0           34m
nginx-app-848cfcf495-9prjh          1/1     Running   0           19m
nginx-app-848cfcf495-gl2kh          1/1     Running   0           19m
nginx-app-848cfcf495-pg2c8          1/1     Running   0           19m
nginx-kusc00101                      1/1     Running   0           26m
pod-secrets-via-env                  1/1     Running   0           4s
pod-secrets-via-file                 1/1     Running   0           106s
webserver-84c55967f4-qzjcv          1/1     Running   0           6h43m
webserver-84c55967f4-t4791          1/1     Running   0           6h43m
root@node-1:~#

```

**NEW QUESTION 10**

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.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml > nginx-prod-pod.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
```

#### NEW QUESTION 11

Configure the kubelet systemd-managed service, on the node labelled with name=wk8s-node-1, to launch a pod containing a single container of Image http://dname:webtool automatically. Any spec files required should be placed in the /etc/kubernetes/manifests directory on the node.

You can ssh to the appropriate node using:

```
[student@node-1] $ ssh wk8s-node-1
```

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

```
[student@wk8s-node-1] $ |sudo ?Ci
```

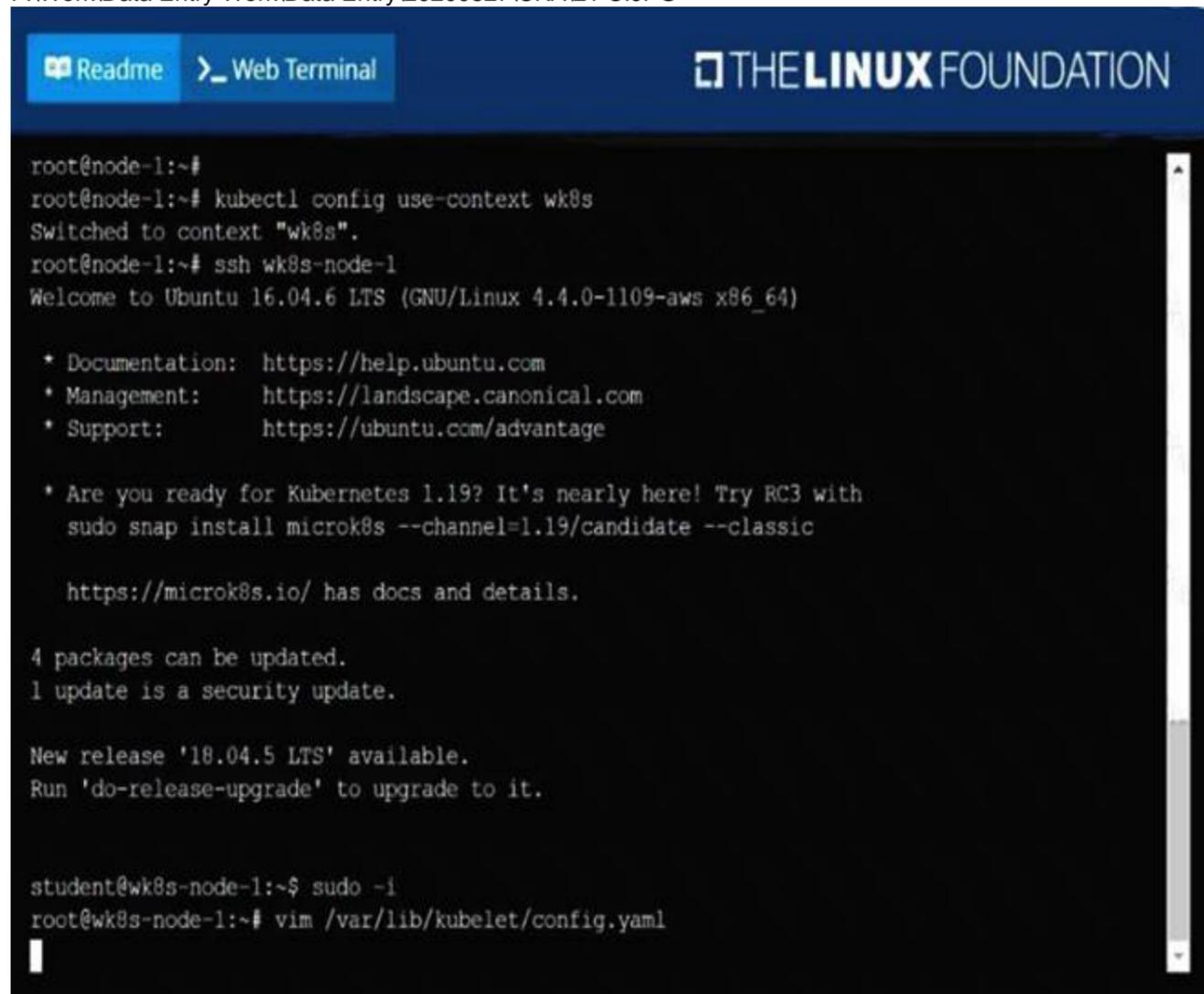
- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 C.JPG



```
Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~#
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# ssh wk8s-node-1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

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

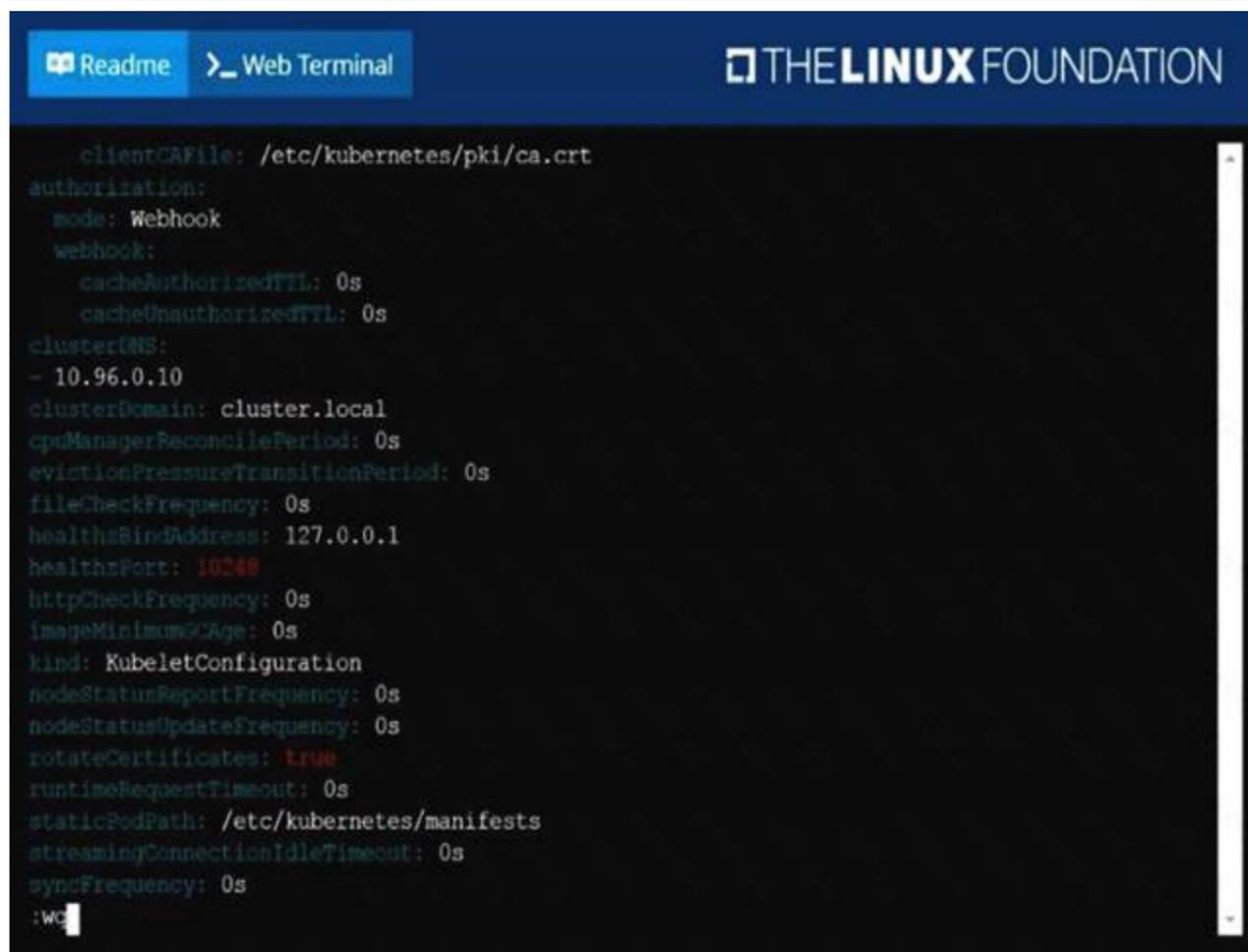
 * Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic
   https://microk8s.io/ has docs and details.

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 D.JPG

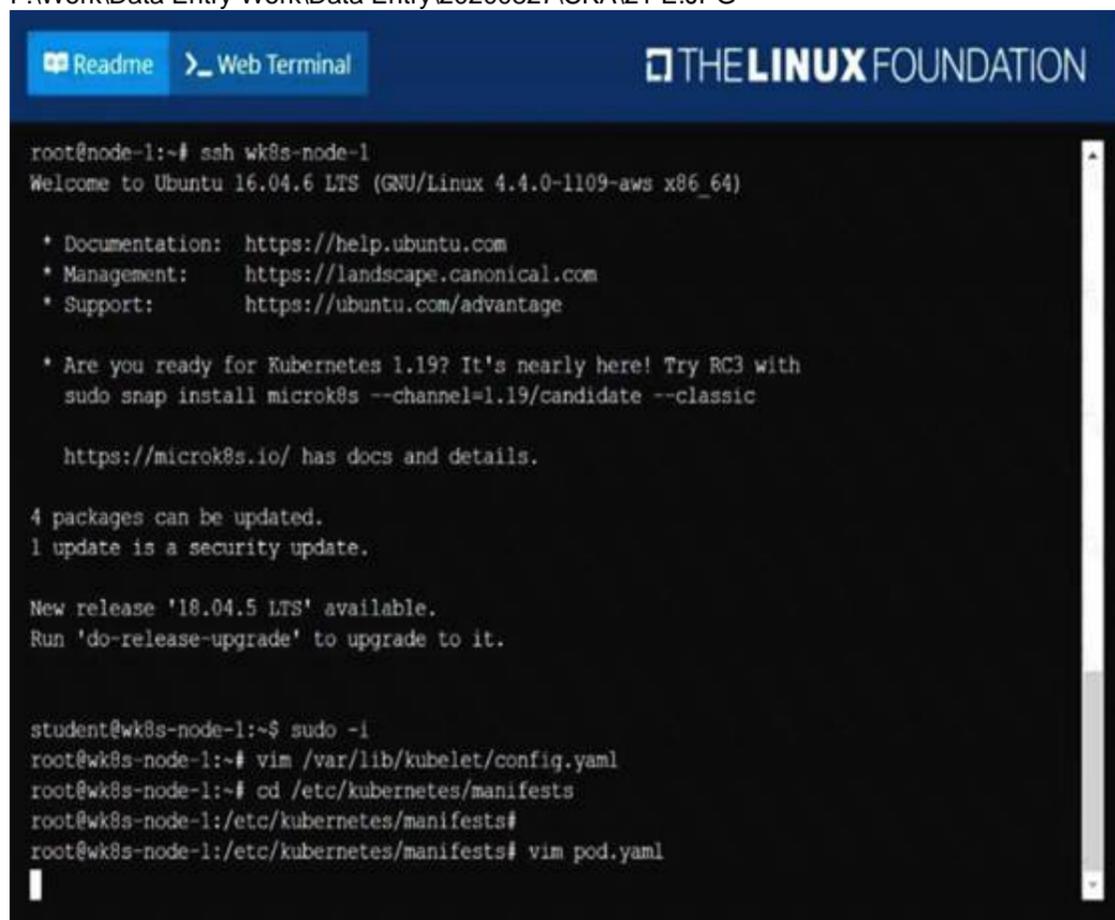


```

clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
syncFrequency: 0s
:wc

```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 E.JPG



```

root@node-1:~# ssh wk8s-node-1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

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

 * Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic

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

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

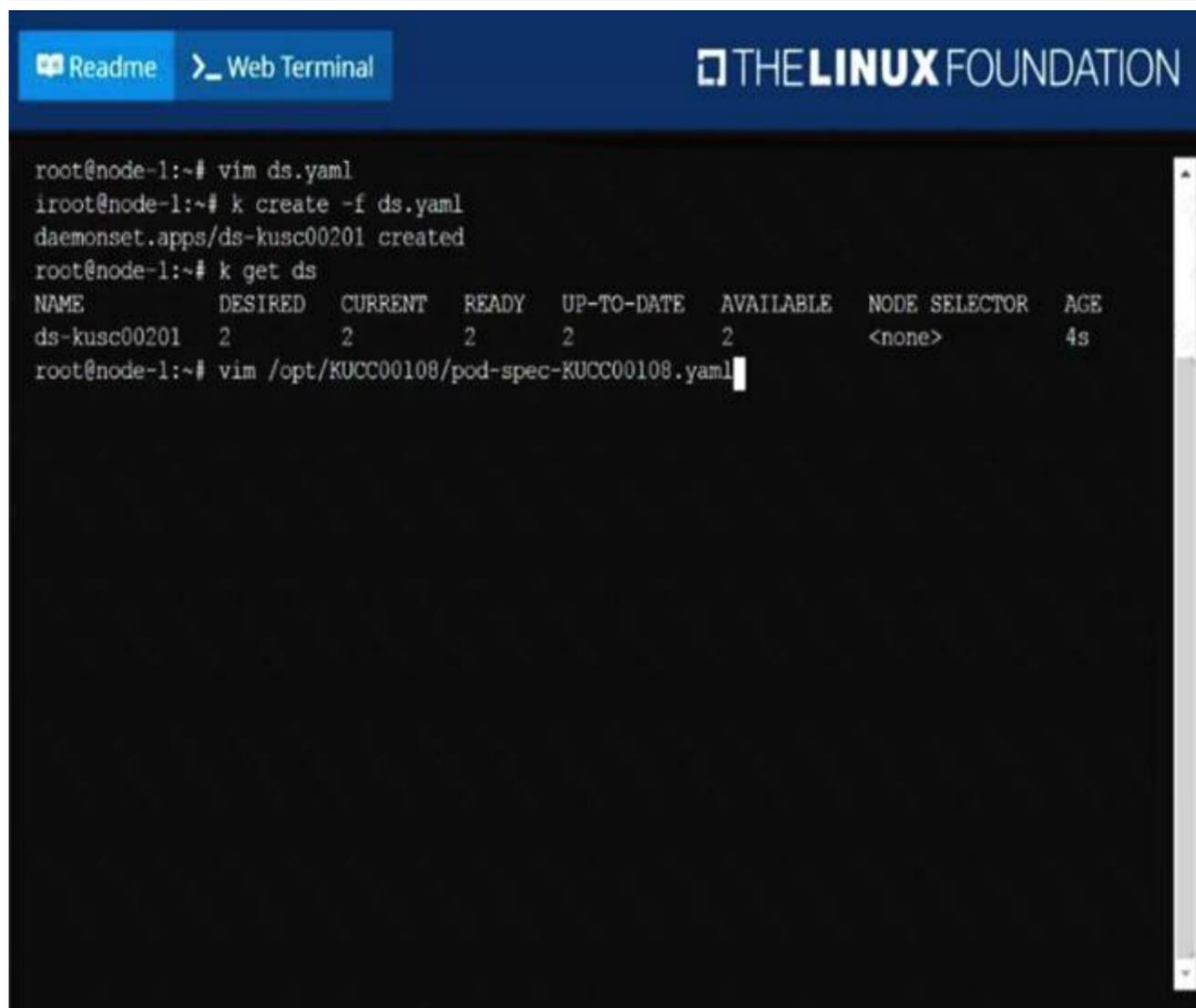
student@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests#
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml

```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 F.JPG

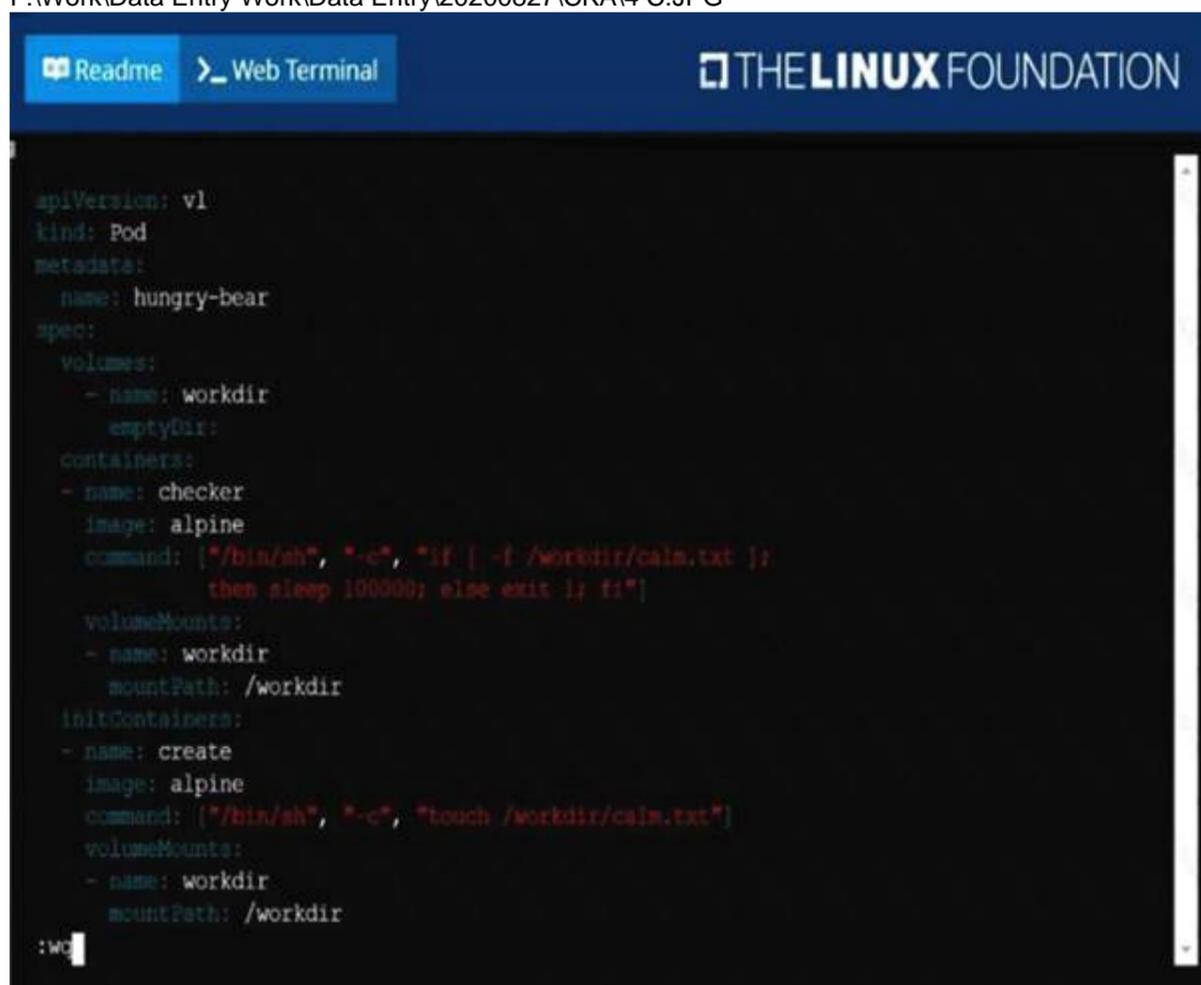






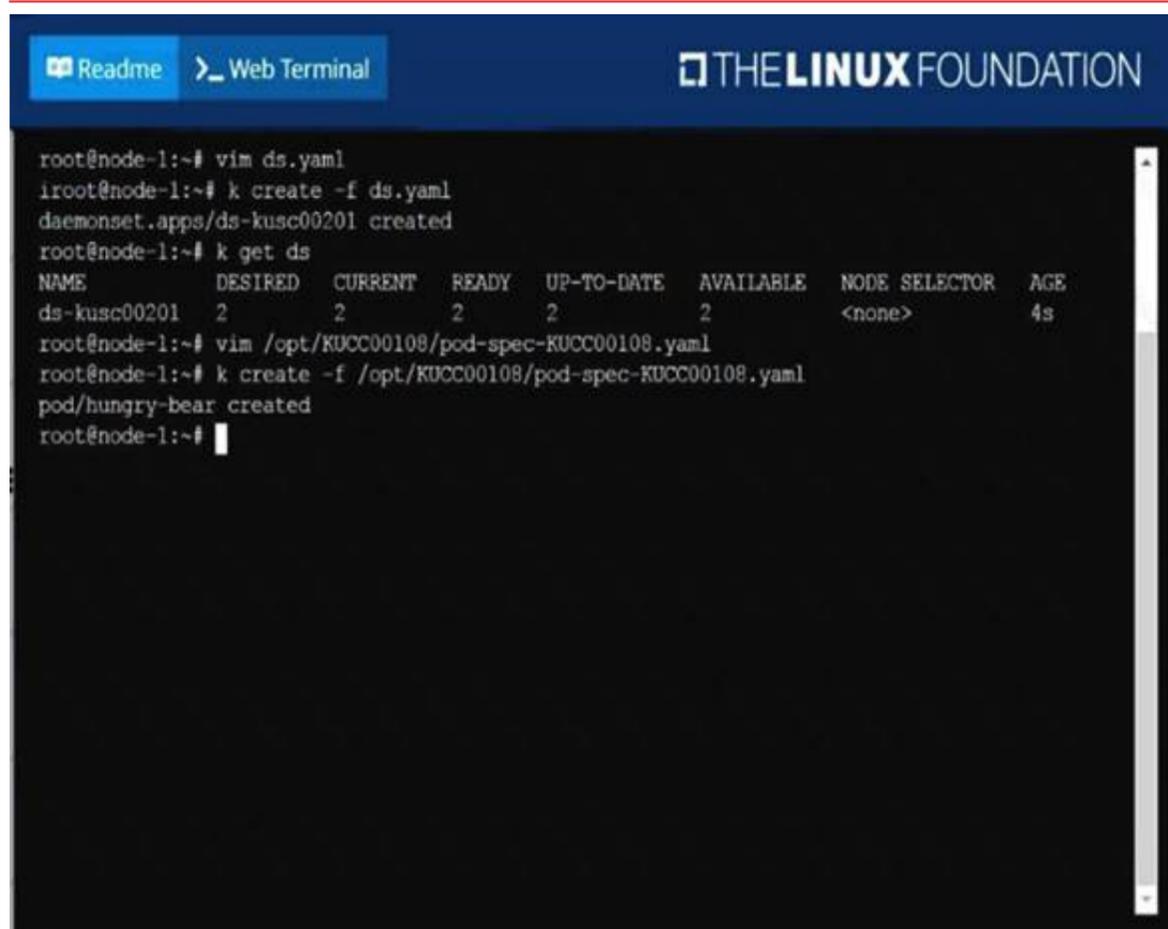
```
Readme Web Terminal THE LINUX FOUNDATION  
root@node-1:~# vim ds.yaml  
iroot@node-1:~# k create -f ds.yaml  
daemonset.apps/ds-kusc00201 created  
root@node-1:~# k get ds  
NAME           DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE  
ds-kusc00201   2        2        2    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

F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 C.JPG



```
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/caln.txt ];  
  then sleep 100000; else exit 1; fi"]  
  volumeMounts:  
  - name: workdir  
    mountPath: /workdir  
initContainers:  
- name: create  
  image: alpine  
  command: ["/bin/sh", "-c", "touch /workdir/caln.txt"]  
  volumeMounts:  
  - name: workdir  
    mountPath: /workdir  
:wq
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 D.JPG



```

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:~#

```

#### NEW QUESTION 19

Check the Image version of nginx-dev pod using jsonpath

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

kubect1 get po nginx-dev -o jsonpath='{.spec.containers[].image}'

#### NEW QUESTION 22

Create a busybox pod and add `sleep 3600` command

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

kubectl run busybox --image=busybox --restart=Never -- /bin/sh -c "sleep 3600"

#### NEW QUESTION 26

Get IP address of the pod `nginx-dev`

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

Kubect1 get po -o wide Using JsonPath  
kubect1 get pods -o=jsonpath='{range items[\*]}{.metadata.name}{\t}{.status.podIP}{\n}{end}'

#### NEW QUESTION 29

List all the pods showing name and namespace with a json path expression

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

kubectl get pods -o=jsonpath='{.items[\*]}[\'metadata.name\', \'metadata.namespace\']'

#### NEW QUESTION 34

Create a snapshot of the etcd instance running at `https://127.0.0.1:2379`, saving the snapshot to the file path `/srv/data/etcd-snapshot.db`.

The following TLS certificates/key are supplied for connecting to the server with etcdctl:

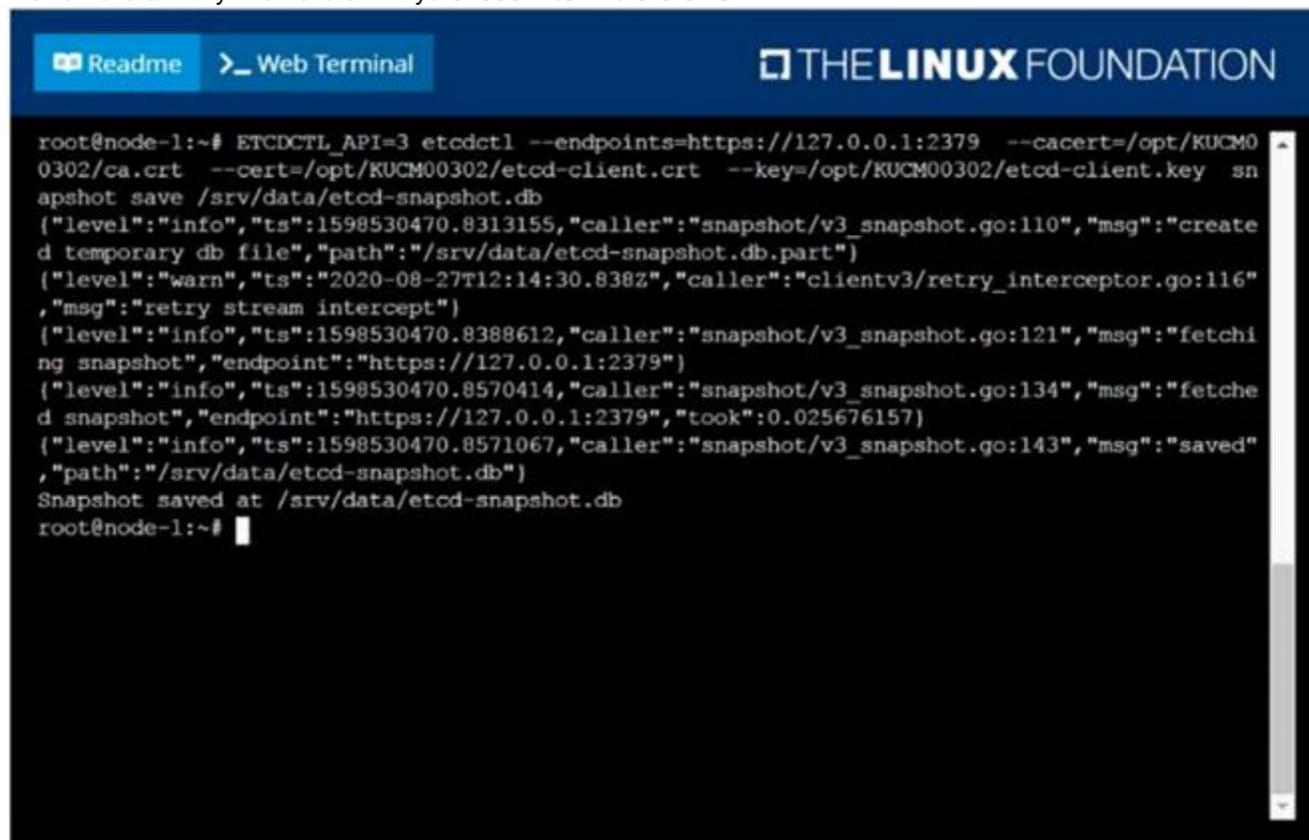
- > CA certificate: /opt/KUCM00302/ca.crt
- > Client certificate: /opt/KUCM00302/etcd-client.crt
- > Client key: /opt/KUCM00302/etcd-client.key

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution  
 F:\Work\Data Entry Work\Data Entry\20200827\CKA\18 C.JPG



**NEW QUESTION 37**

Schedule a pod as follows:

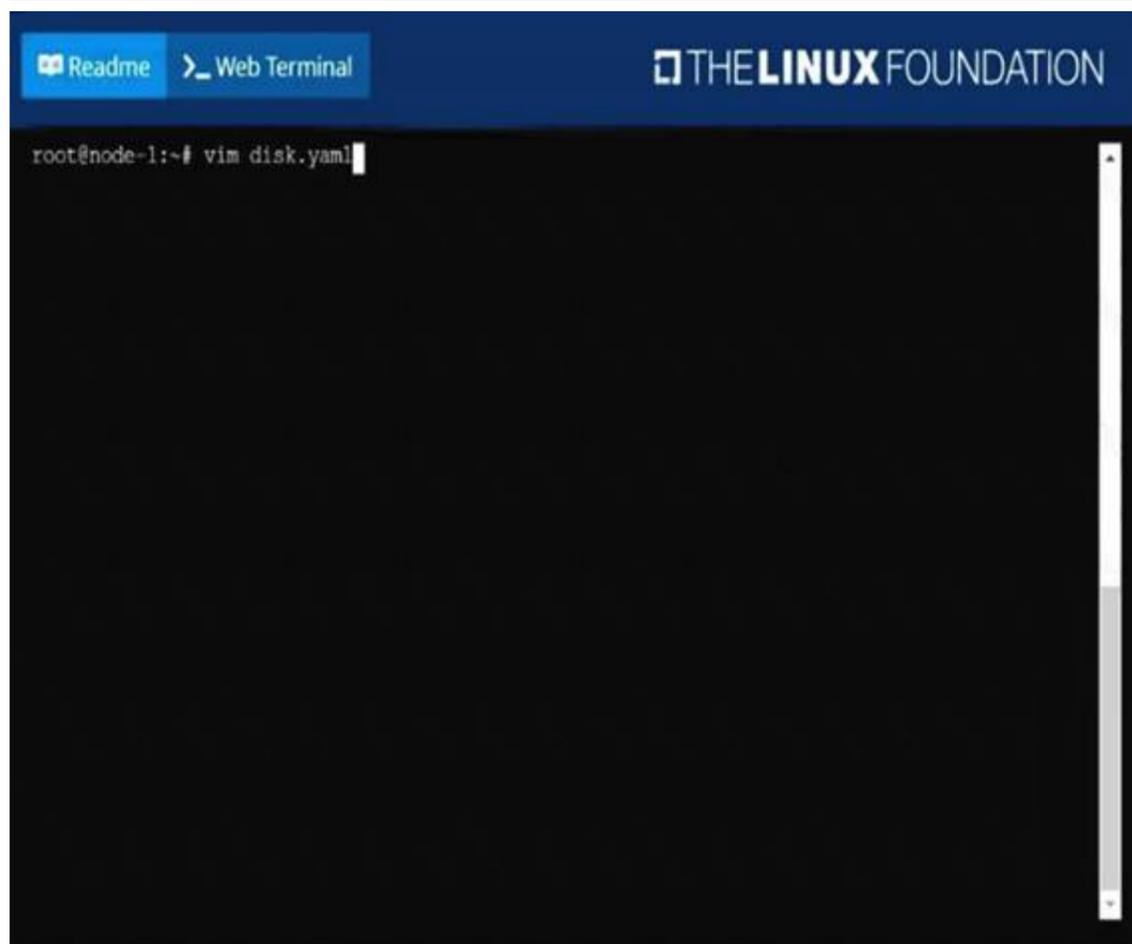
- > Name: nginx-kusc00101
- > Image: nginx
- > Node selector: disk=ssd

- A. Mastered
- B. Not Mastered

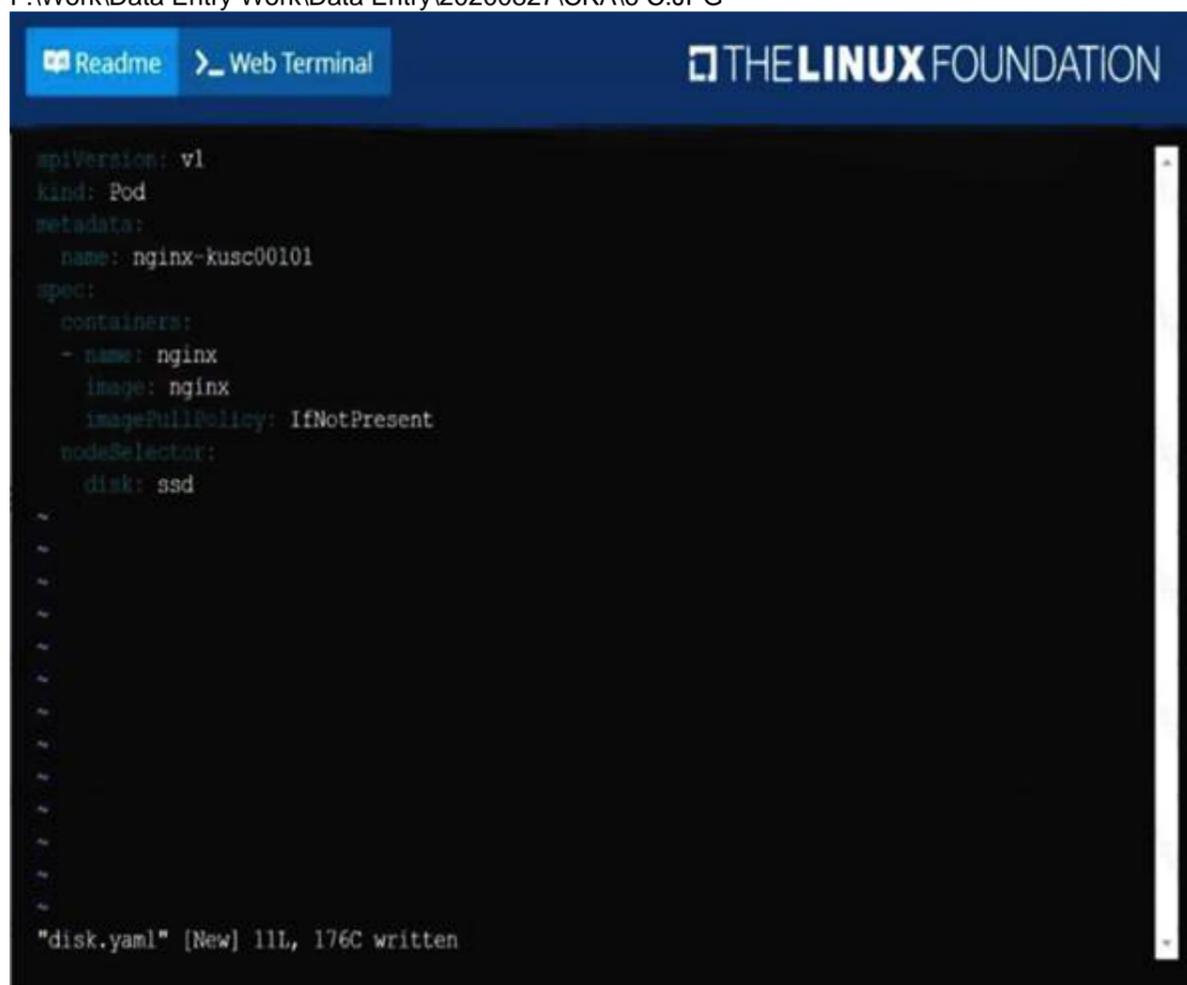
**Answer:** A

**Explanation:**

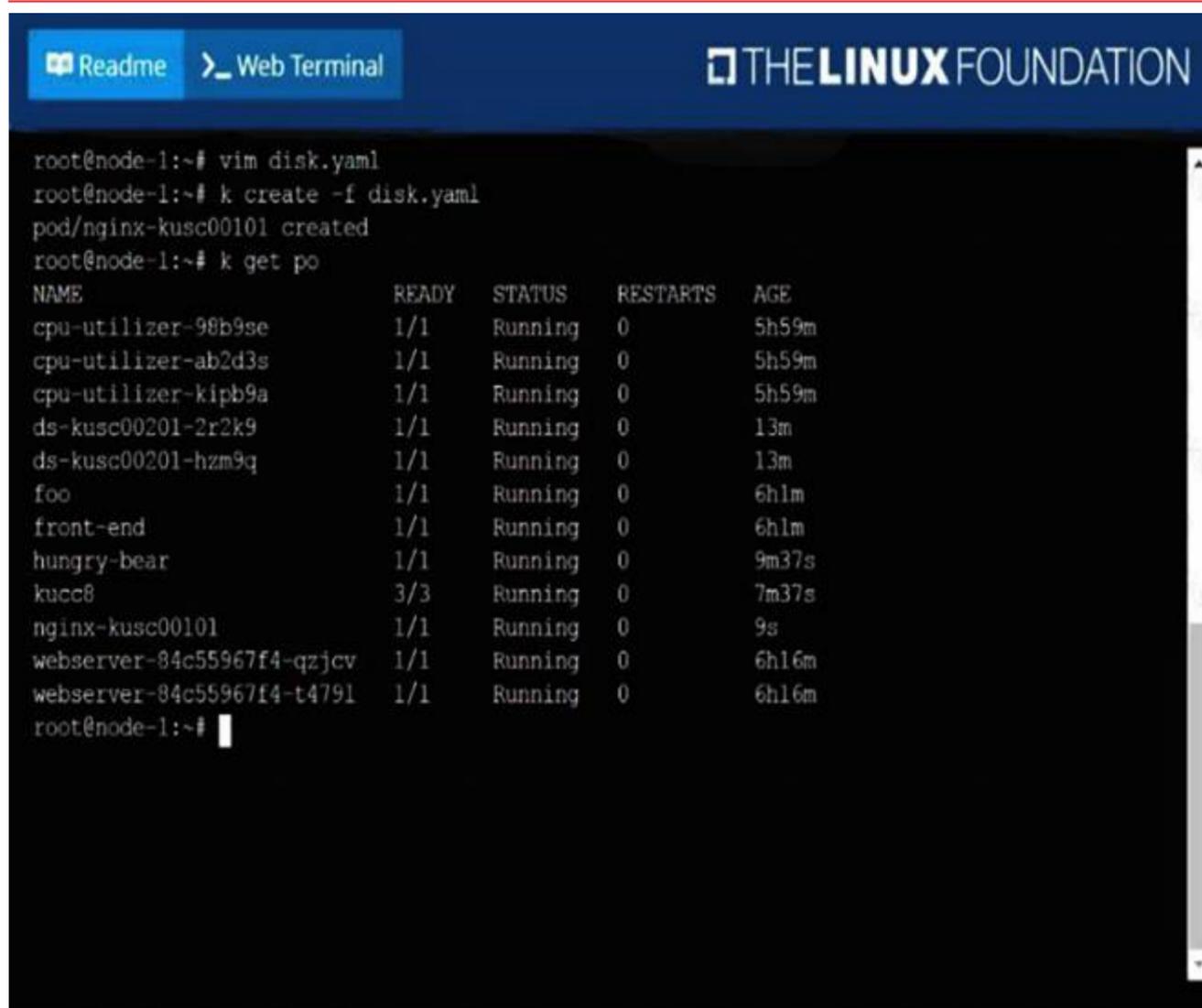
solution  
 F:\Work\Data Entry Work\Data Entry\20200827\CKA\16 B.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 D.JPG



```

root@node-1:~# vim disk.yaml
root@node-1:~# k create -f disk.yaml
pod/nginx-kusc00101 created
root@node-1:~# k get po
NAME                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se  1/1     Running   0           5h59m
cpu-utilizer-ab2d3s  1/1     Running   0           5h59m
cpu-utilizer-kipb9a  1/1     Running   0           5h59m
ds-kusc00201-2r2k9   1/1     Running   0           13m
ds-kusc00201-hzm9q   1/1     Running   0           13m
foo                  1/1     Running   0           6h1m
front-end            1/1     Running   0           6h1m
hungry-bear          1/1     Running   0           9m37s
kucc8                 3/3     Running   0           7m37s
nginx-kusc00101      1/1     Running   0           9s
webserver-84c55967f4-qzjcv  1/1     Running   0           6h16m
webserver-84c55967f4-t479l  1/1     Running   0           6h16m
root@node-1:~#

```

**NEW QUESTION 40**

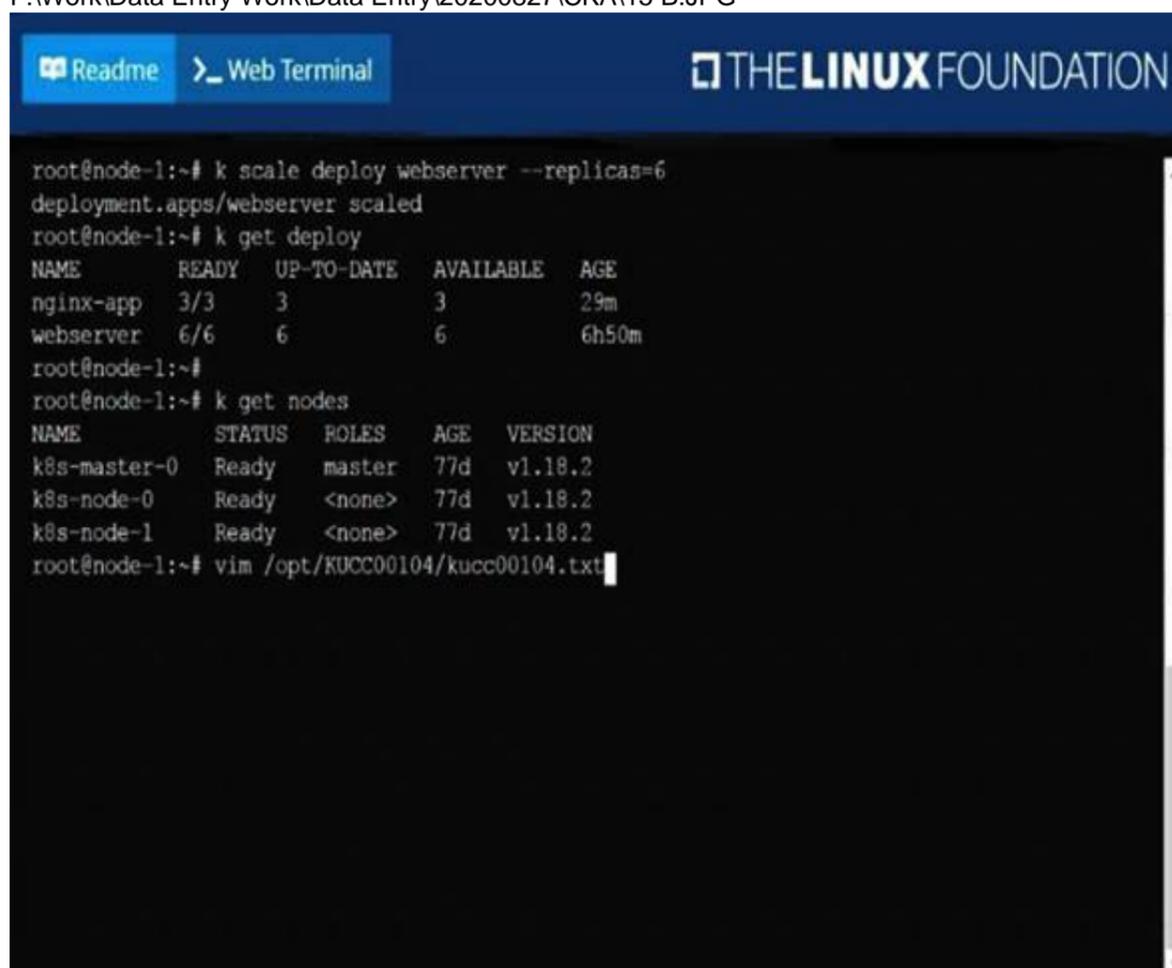
Check to see how many worker nodes are ready (not including nodes taintedNoSchedule) and write the number to/opt/KUCC00104/kucc00104.txt.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution  
F:\Work\Data Entry Work\Data Entry\20200827\CKA\15 B.JPG

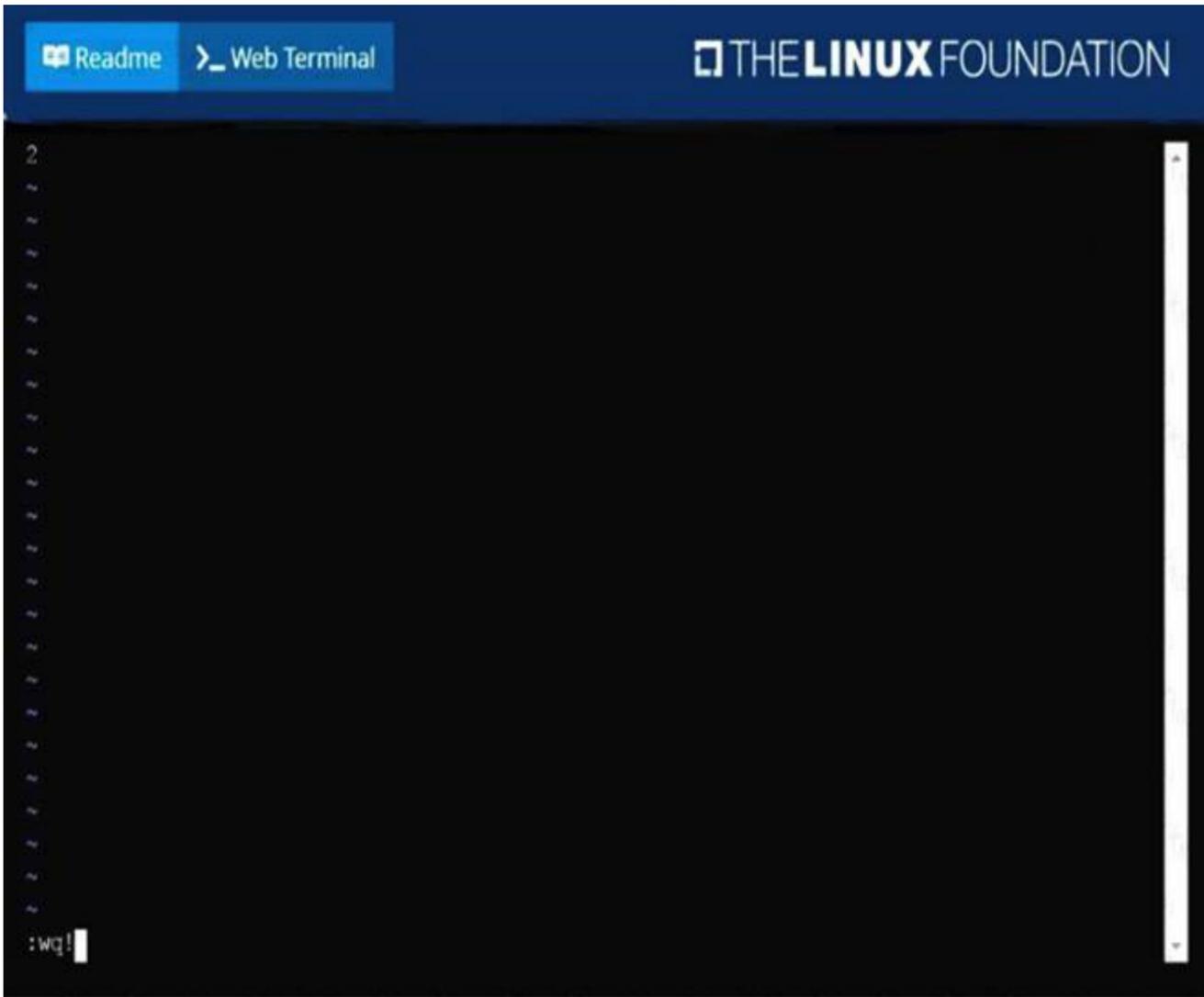


```

root@node-1:~# k scale deploy webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app   3/3     3             3           29m
webserver   6/6     6             6           6h50m
root@node-1:~#
root@node-1:~# k get nodes
NAME           STATUS   ROLES    AGE   VERSION
k8s-master-0  Ready   master   77d   v1.18.2
k8s-node-0    Ready   <none>   77d   v1.18.2
k8s-node-1    Ready   <none>   77d   v1.18.2
root@node-1:~# vim /opt/KUCC00104/kucc00104.txt

```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\15 C.JPG



**NEW QUESTION 45**

.....

## Relate Links

**100% Pass Your CKA Exam with Examible Prep Materials**

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