

## Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

<https://www.2passeasy.com/dumps/CKA/>



### NEW QUESTION 1

Create a deployment as follows:

- > Name:nginx-app
- > Using containernginxwithversion 1.11.10-alpine
- > The deployment should contain3replicas

Next, deploy the application with newversion1.11.13-alpine, byperforming a rolling update.

Finally, rollback that update to theprevious version1.11.10-alpine.

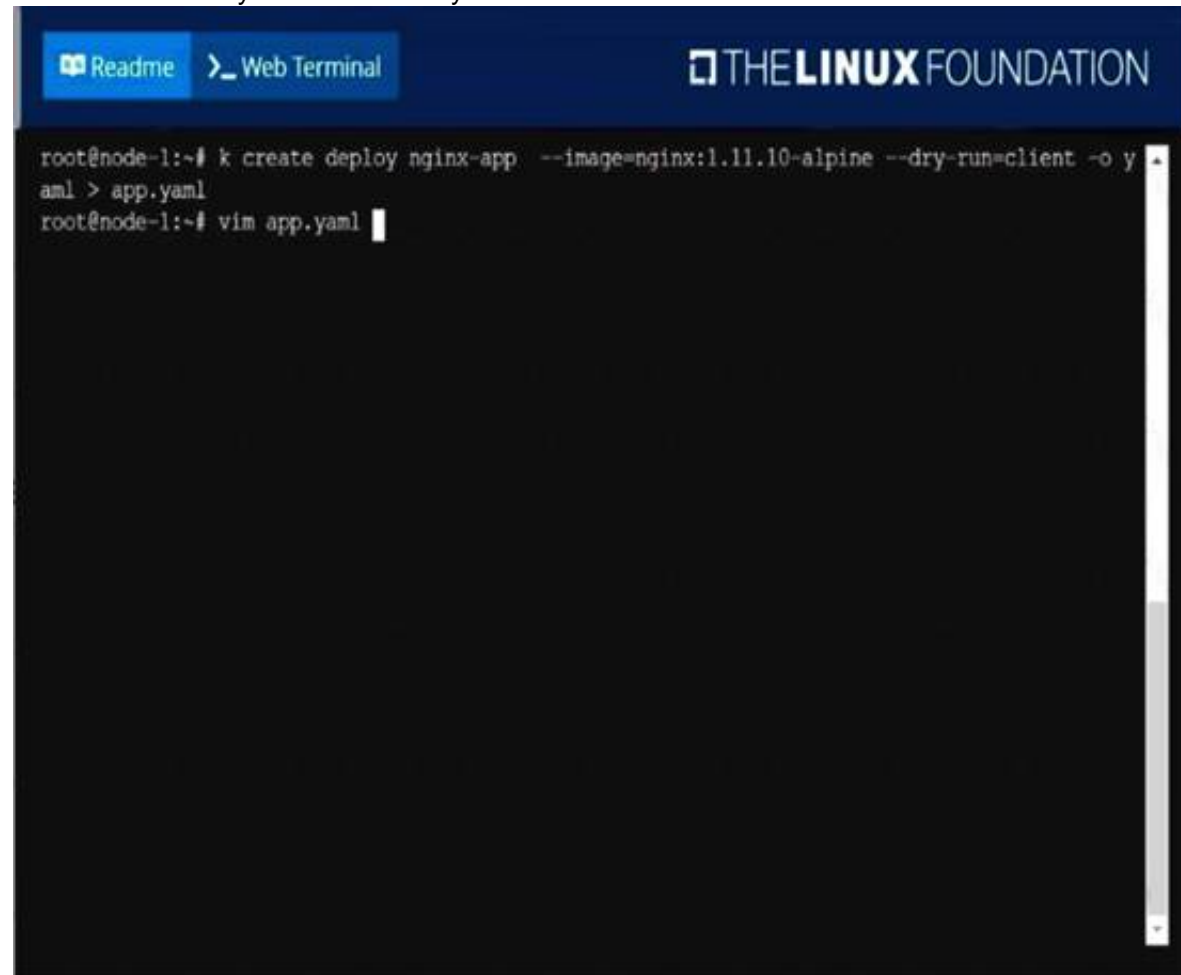
- A. Mastered
- B. Not Mastered

**Answer:** A

### Explanation:

solution

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

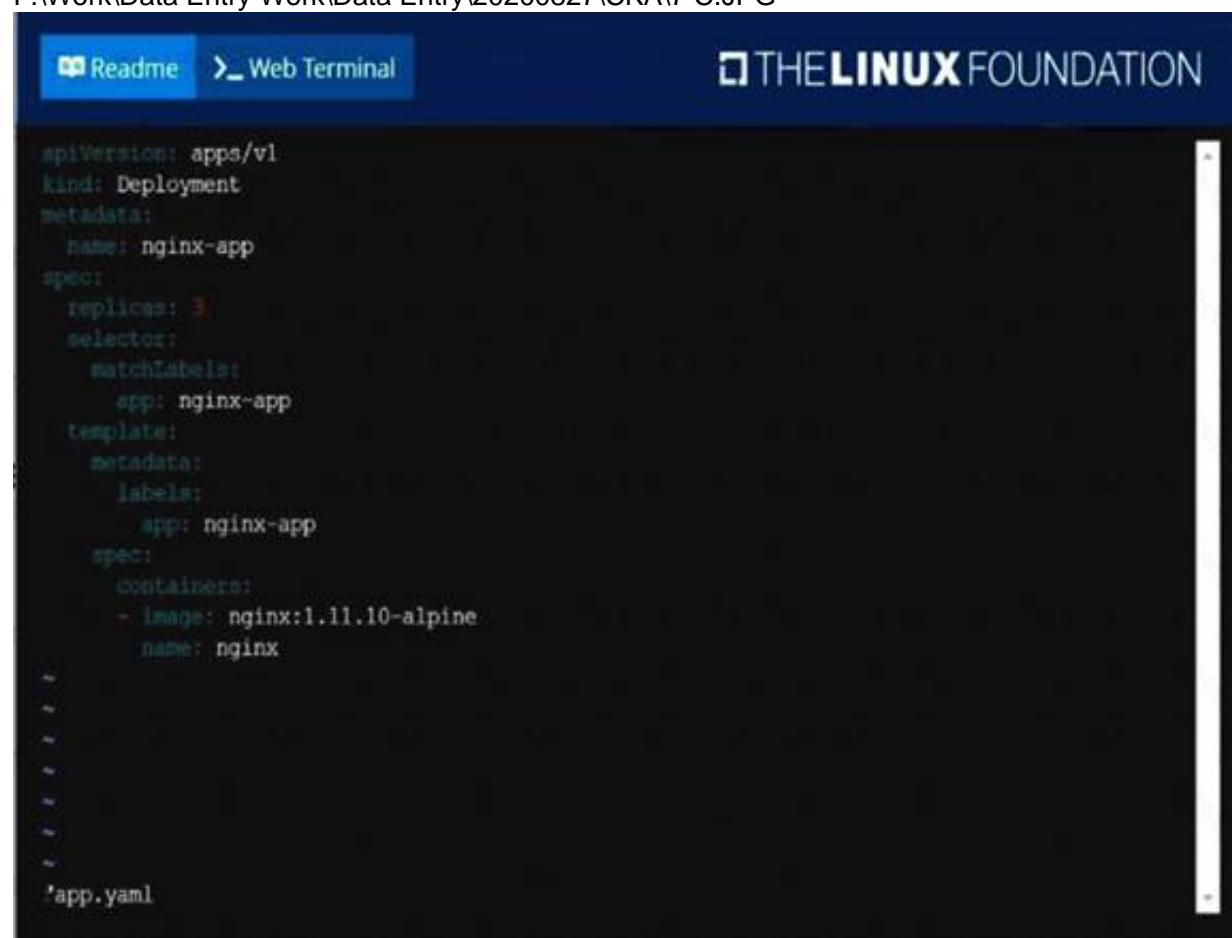


```

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

```

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

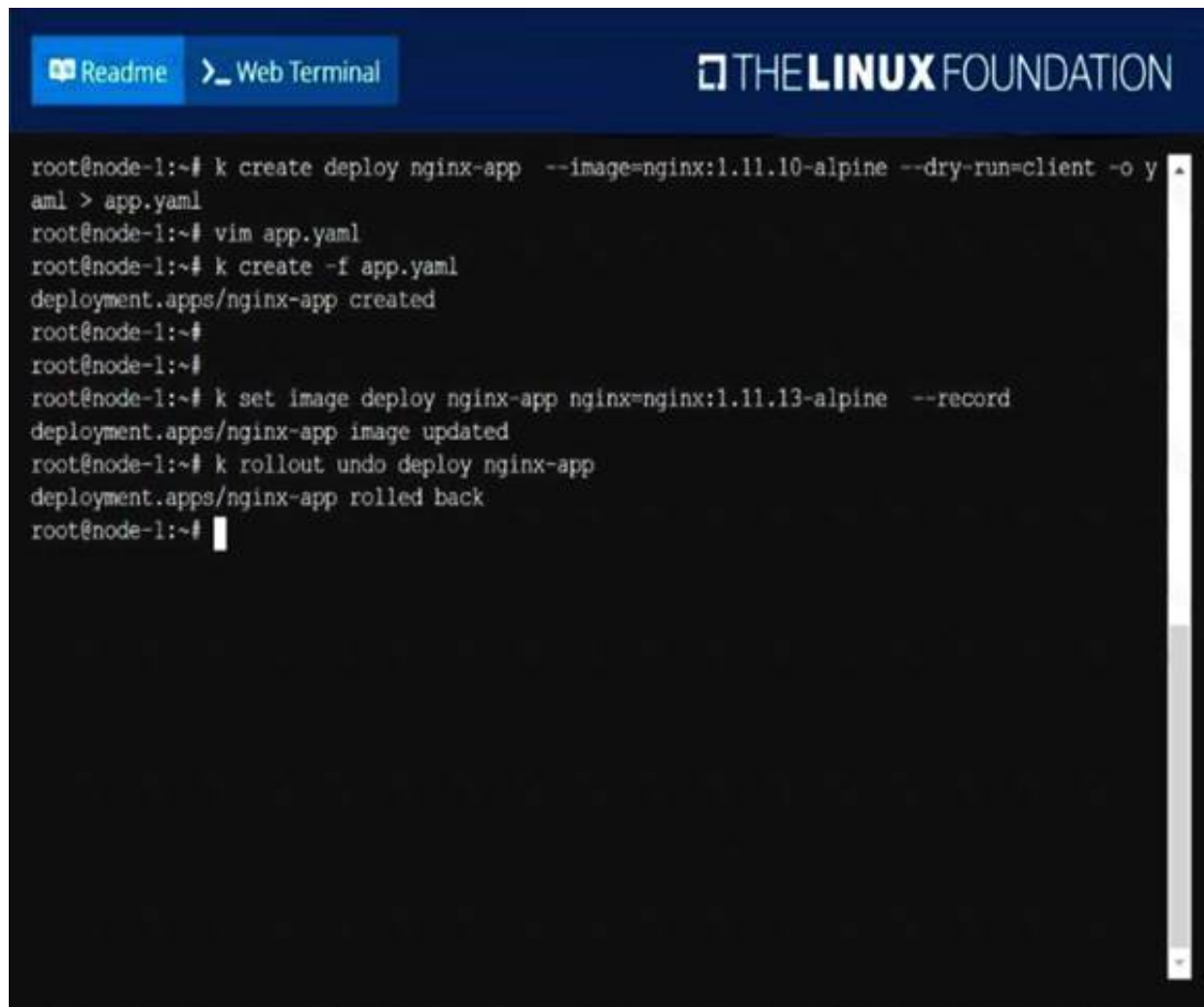


```

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

```

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



```

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 2

Create a nginx pod with label env=test in engineering namespace

- A. Mastered
- B. Not Mastered

**Answer:** A

### Explanation:

kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml > nginx-pod.yaml  
 kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml | kubectl create -nengineering-f ?C  
 YAML File: apiVersion: v1 kind: Pod metadata: name: nginx  
 namespace: engineering labels:  
 env: test spec: containers:  
 - name: nginx image: nginx  
 imagePullPolicy: IfNotPresent restartPolicy: Never  
 kubectl create -f nginx-pod.yaml

## NEW QUESTION 3

Create a persistent volume with nameapp-data, of capacity2Giandaccess modeReadWriteMany. Thetype of volume ishostPathand itslocation is/srv/app-data.

- A. Mastered
- B. Not Mastered

**Answer:** A

### Explanation:

solution  
 Persistent Volume  
 A persistent volume is a piece of storage in aKubernetes 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 knowthe 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: PersistentVolumeapiVersion: v1metadata:name:app-dataspec:capacity: # defines the capacity of PV we are creatingstorage:2Gi#the amount of storage we are tying to claimaccessModes: # defines the rights of the volumewe are creating-ReadWriteManyhostPath:path: "/srv/app-data" # path to which we are creating the volume  
 Challenge  
 > Create a Persistent Volume namedapp-data, with access modeReadWriteMany, storage classname shared,2Giof 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: shared
```

"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: PersistentVolumeapiVersion: v1metadata:name:app-data spec:

accessModes:-ReadWriteManyresources:

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: v1kind: Podmetadata:creationTimestamp: nullname: app-dataspec:volumes:- name: configpvcpersistentVolumeClaim:claimName: app-datacontainers:-

image: nginxname: app-volumeMounts:- mountPath: "/srv/app-data"name: configpvc

#### NEW QUESTION 4

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubect1 get pods --sort-by=.metadata.name

#### NEW QUESTION 5

List pod logs named ??frontend?? and search for the pattern ??started?? and write it to a file ??/opt/error-logs??

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Kubectl logs frontend | grep -i ??started?? > /opt/error-logs

#### NEW QUESTION 6

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 thestagingnamespace and the volumemust notbe persistent.

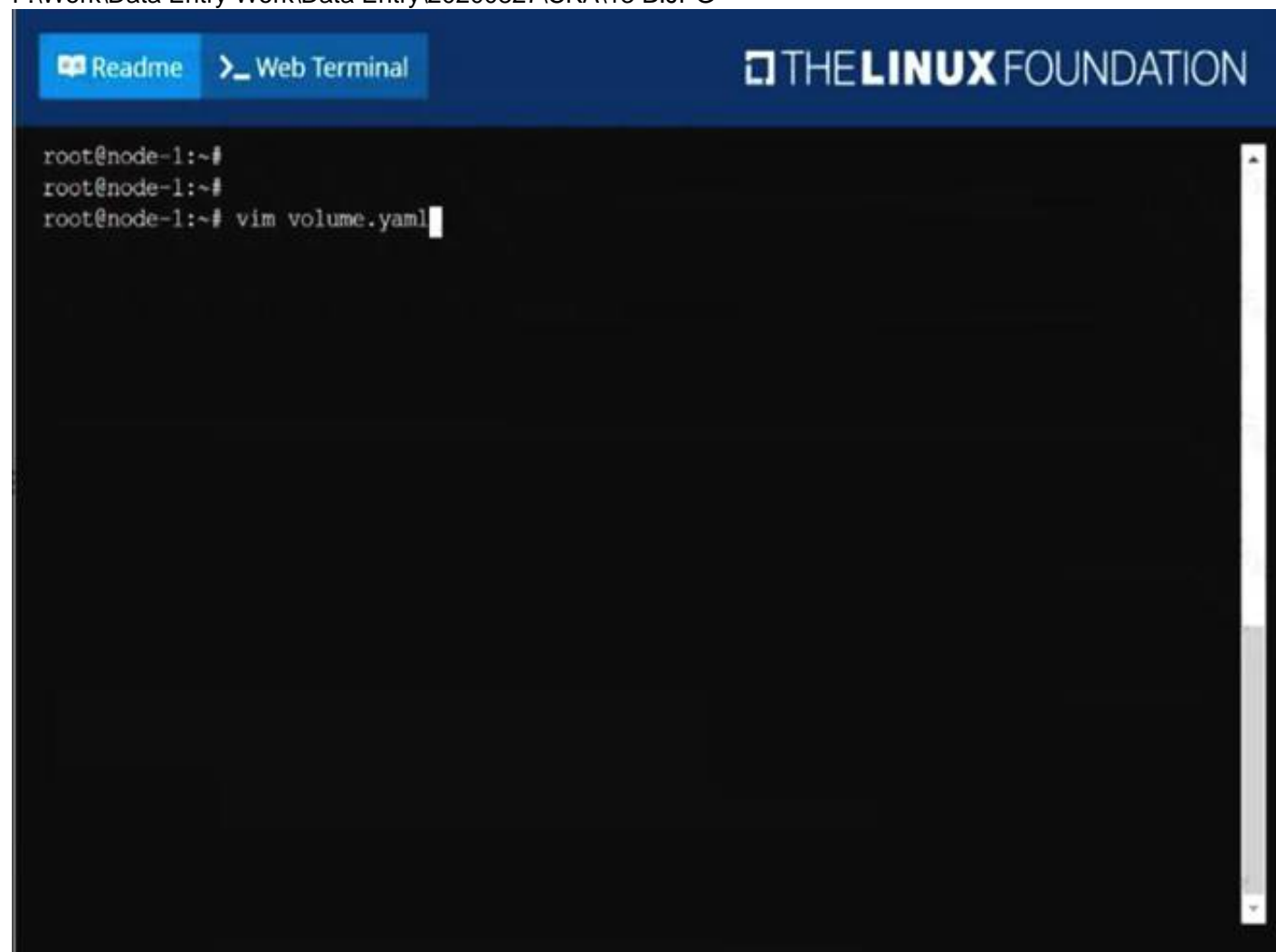
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

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



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





UseDaemonSetto complete thistask and usesd-kusc00201asDaemonSet name.

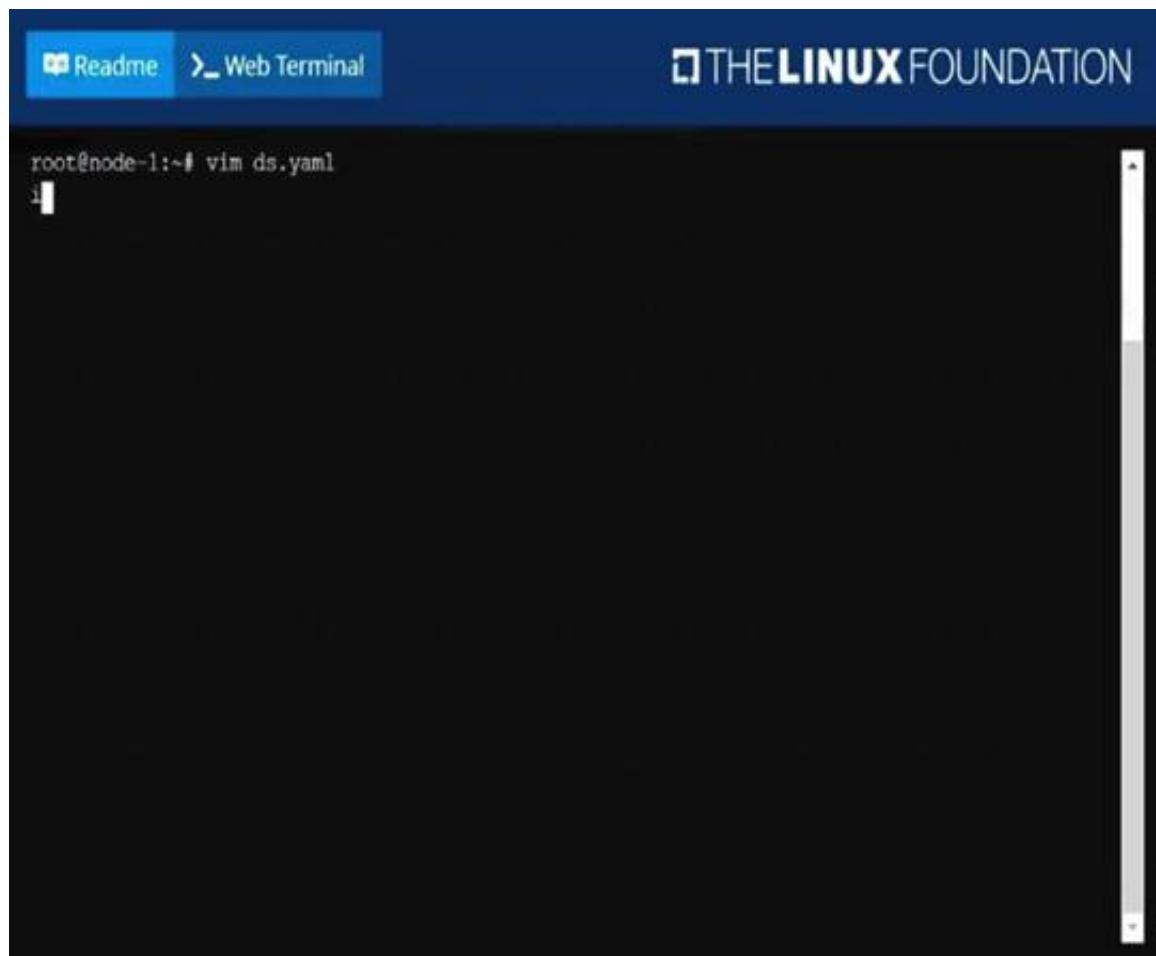
- A. Mastered  
B. Not Mastered

**Answer: A**

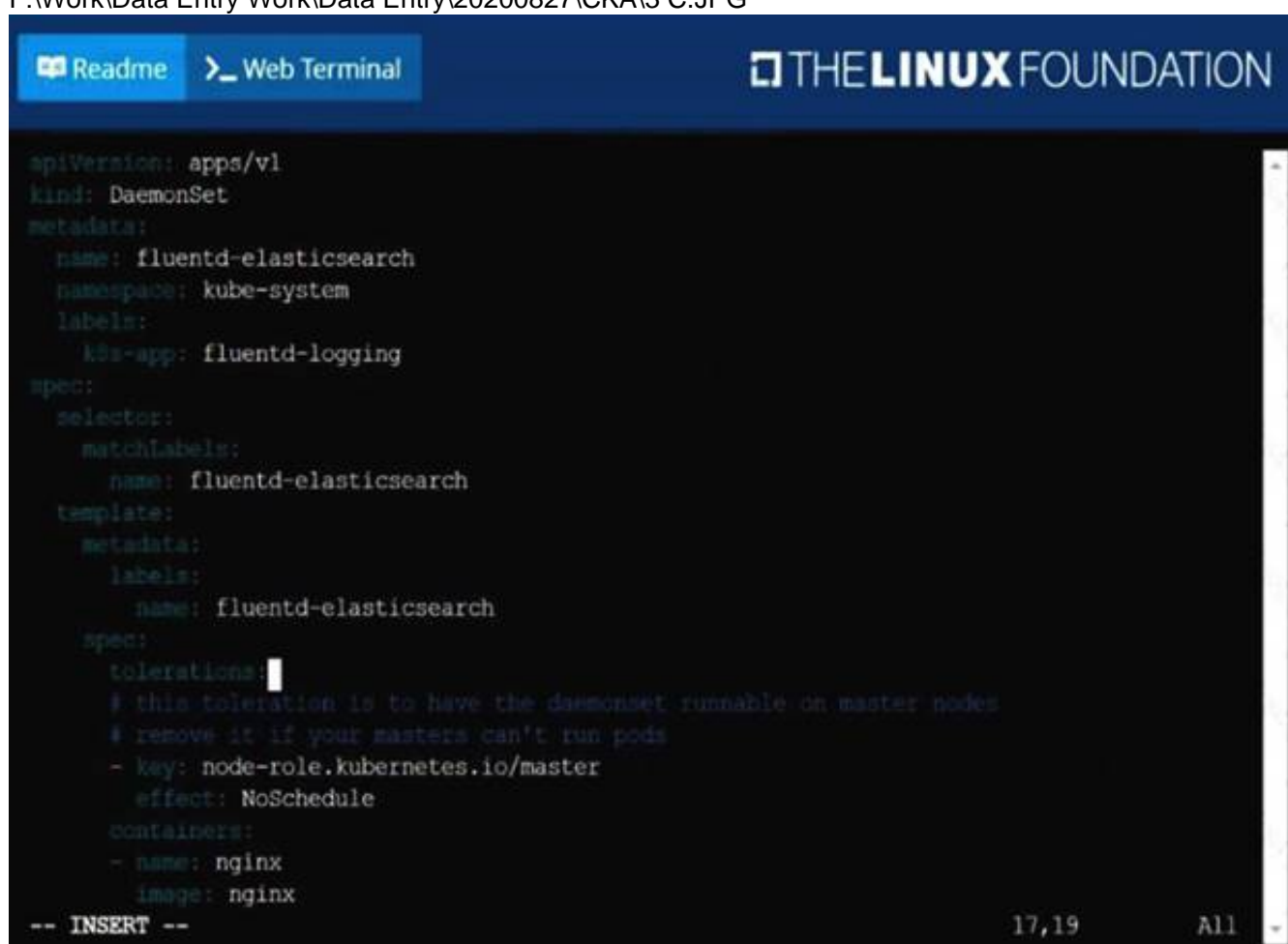
**Explanation:**

### Explain solution

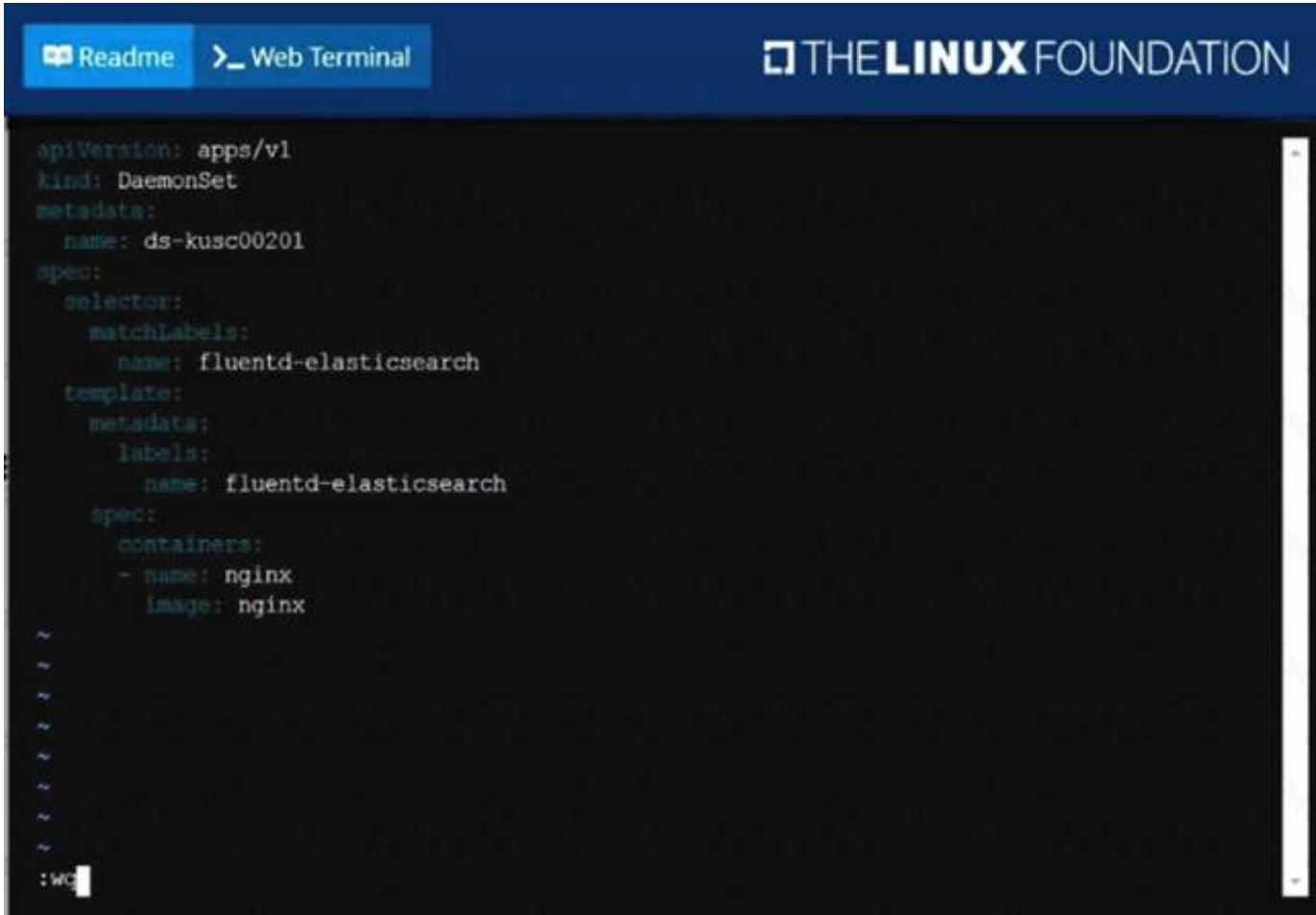
F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 B.JPG



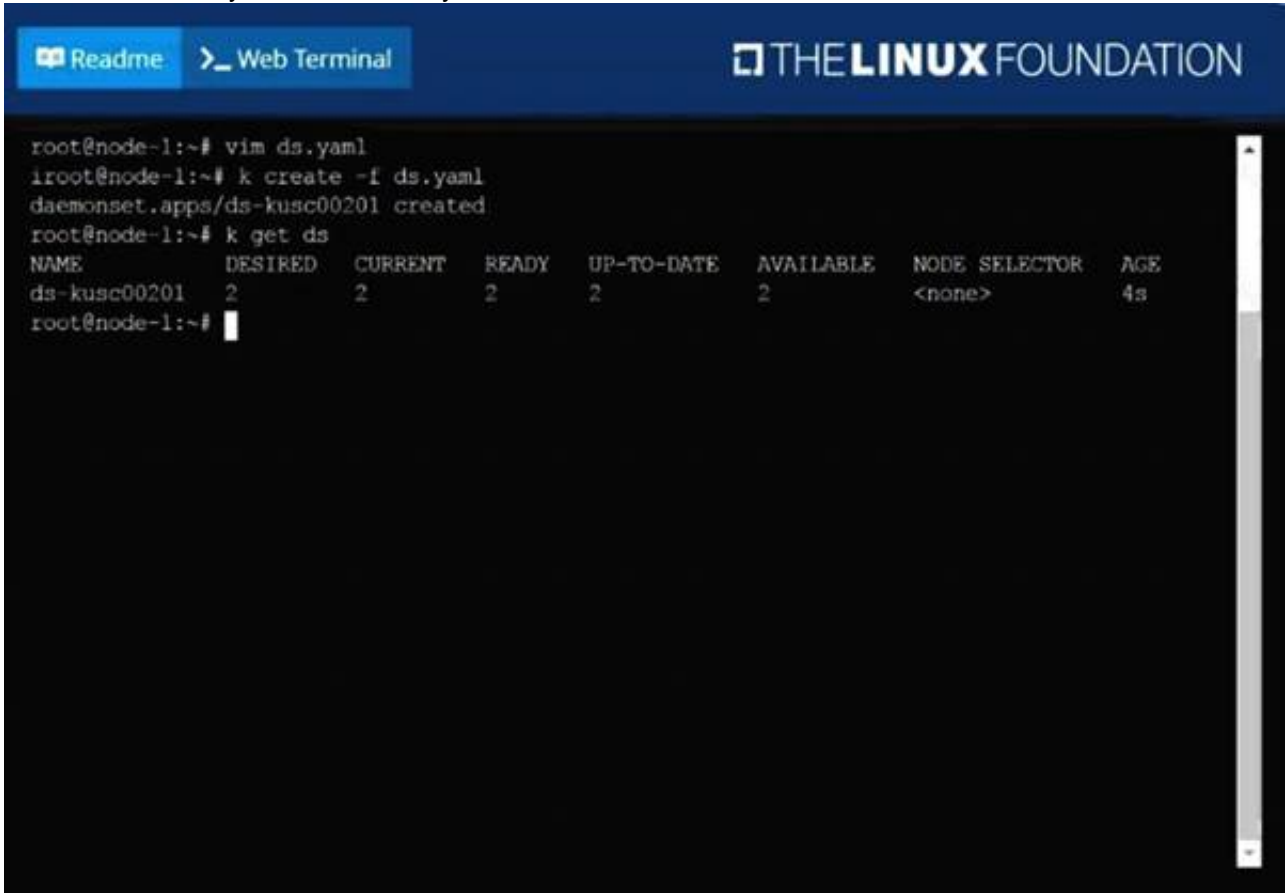
F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 D.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 E.JPG



NEW QUESTION 8

Perform the following tasks:

- > Add an init container tohungry-bear(which has beendefined in spec file /opt/KUCC00108/pod-spec-KUCC00108.yaml)
- > The init container should createan empty file named/workdir/calm.txt
- > If/workdir/calm.txtis notdetected, the pod should exit
- > Once the spec file has beenupdatedwith the init containerdefinition, the pod should becreated

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

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



ReadmeWeb Terminal

THELINUX 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
```

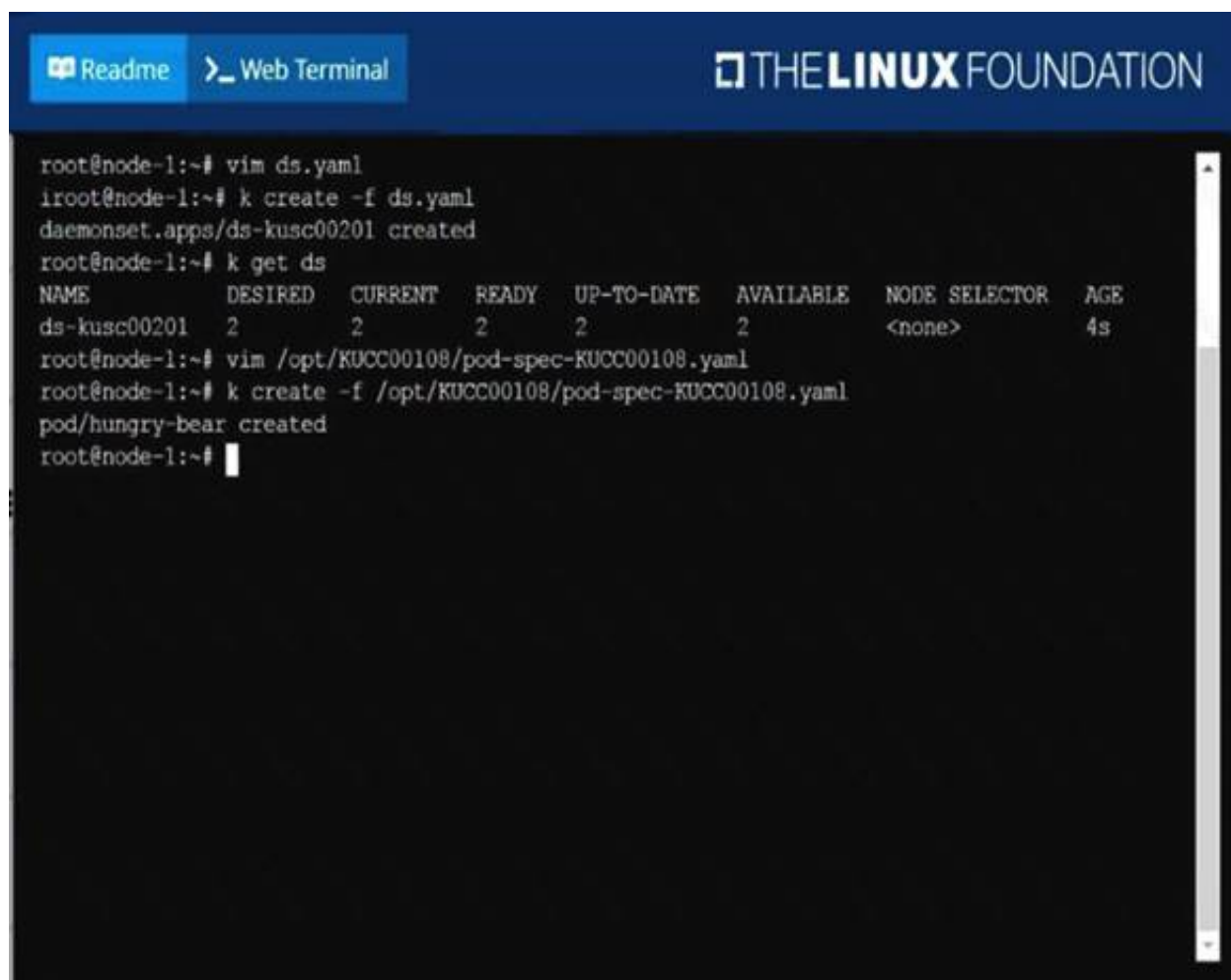
F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 C.JPG

ReadmeWeb Terminal

THELINUX 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
```

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



```
root@node-1:~# vim ds.yaml
root@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 9

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 10

Scale the deploymentwebserverto6pods.

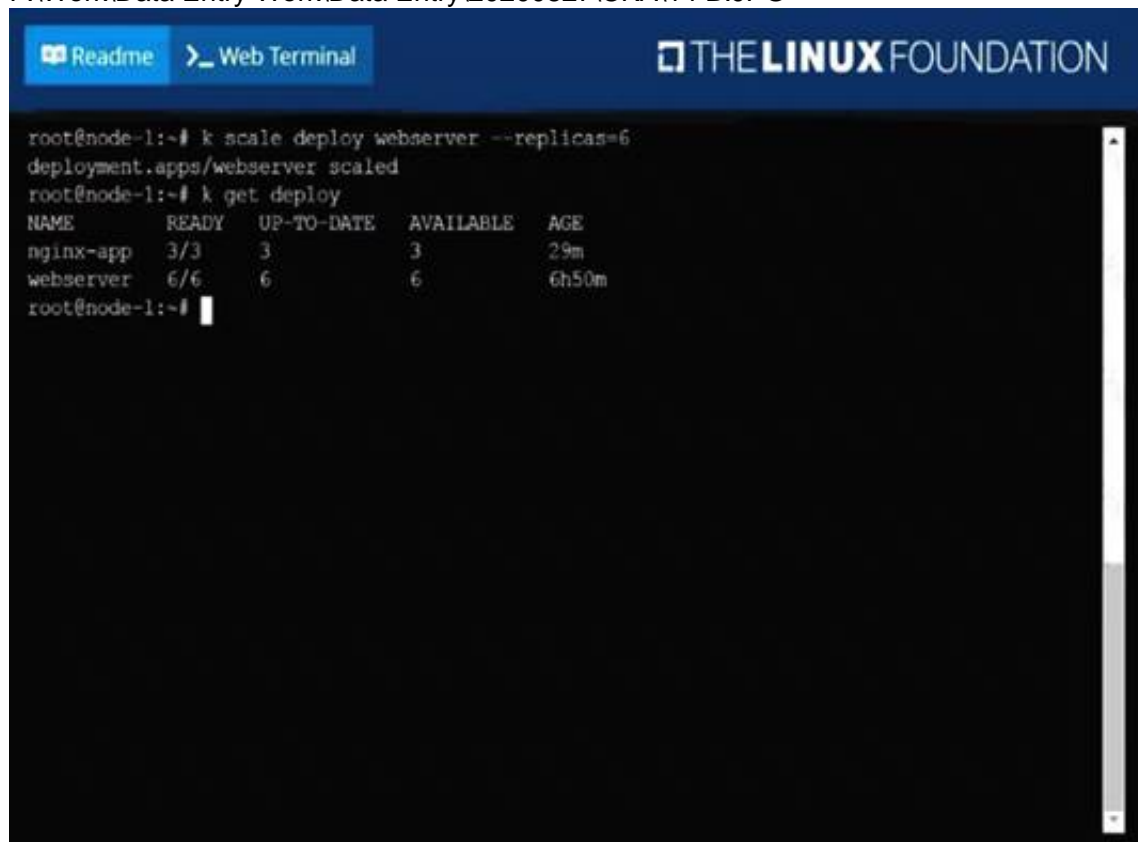
- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\14 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:~#
```

#### NEW QUESTION 10

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual CKA Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the CKA Product From:

<https://www.2passeasy.com/dumps/CKA/>

## Money Back Guarantee

### CKA Practice Exam Features:

- \* CKA Questions and Answers Updated Frequently
- \* CKA Practice Questions Verified by Expert Senior Certified Staff
- \* CKA Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* CKA Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year