

## Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

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



**NEW QUESTION 1**

CORRECT TEXT

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

**Answer:** A**Explanation:**

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

**NEW QUESTION 2**

CORRECT TEXT

Score: 7%

No configuration context  change required for this task.

Ensure, however, that you have returned to the base node before starting to work on this task:

```
[student@mk8s-master-0] |
$
exit
```

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`.

Creating a snapshot of the  given instance is expected to complete in seconds.

If the operation seems to hang, something's likely wrong with your command. Use `CTRL + C` to cancel the operation and try again.

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

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

- CA certificate:  
/opt/KUIN00601/ca.crt
- Client certificate:  
/opt/KUIN00601/etcd-client.crt
- Client key:  
/opt/KUIN00601/etcd-client.key

- A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

#backup

```
ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN00601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot save /etc/data/etcd-snapshot.db
```

#restore

```
ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN00601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot restore /var/lib/backup/etcd-snapshot-previoys.db
```

**NEW QUESTION 3**

CORRECT TEXT

Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified):  
nginx + redis + memcached.

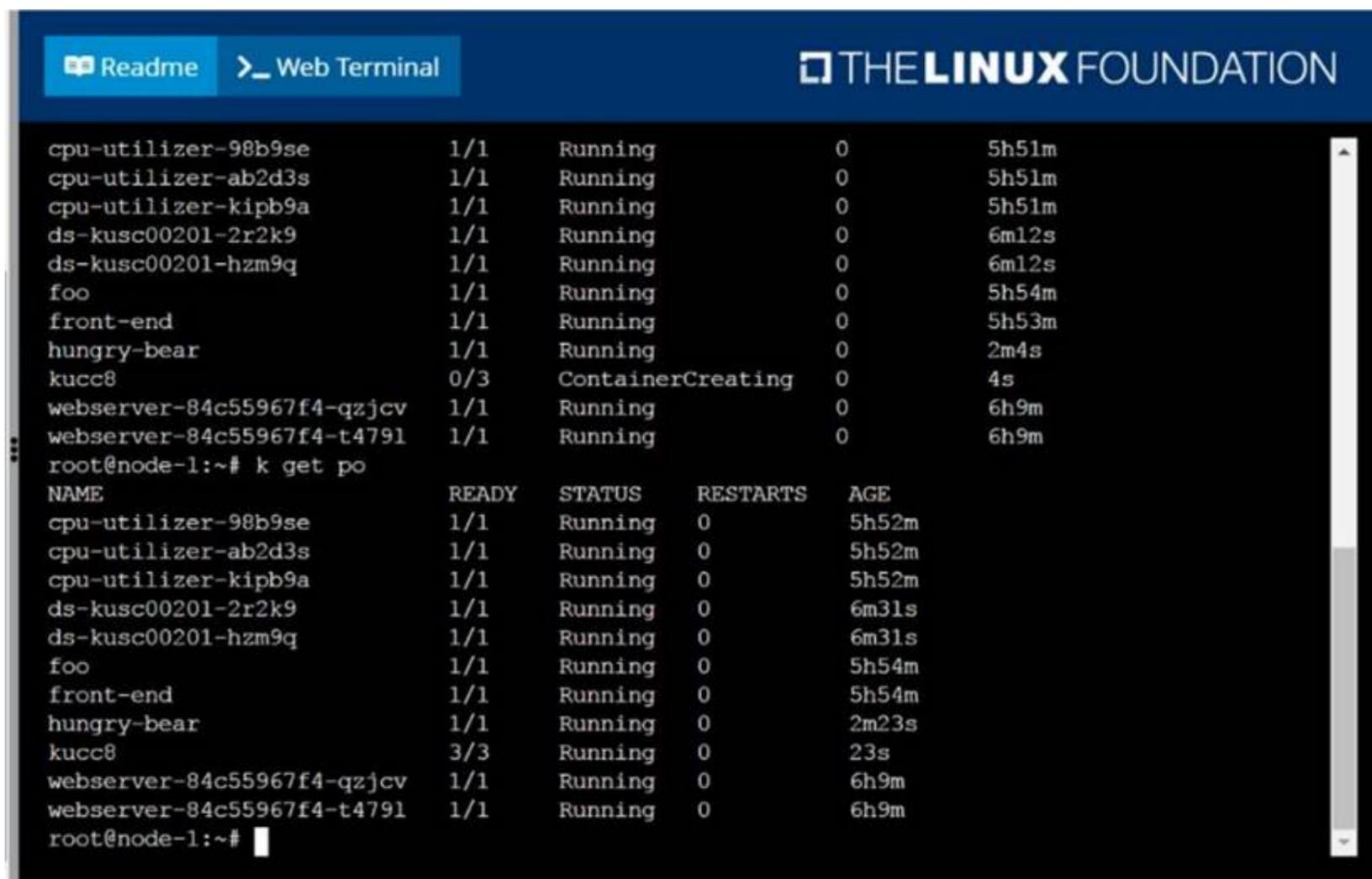
- A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**

solution



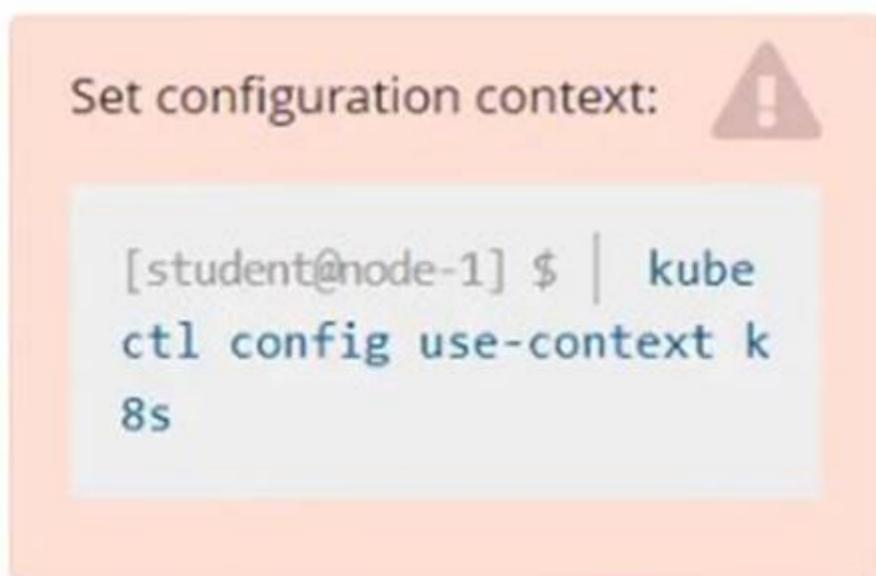


F:\Work\Data Entry Work\Data Entry\20200827\CKA\5 D.JPG

**NEW QUESTION 4**

CORRECT TEXT

Score: 4%



Task

Scale the deployment presentation to 6 pods.

- A. Mastered
- B. Not Mastered

**Answer: A**

**Explanation:**

Solution:

kubectl get deployment

kubectl scale deployment.apps/presentation --replicas=6

**NEW QUESTION 5**

CORRECT TEXT

Create a deployment as follows:

? Name: nginx-app

? Using container nginx with version 1.11.10-alpine

? The deployment should contain 3 replicas

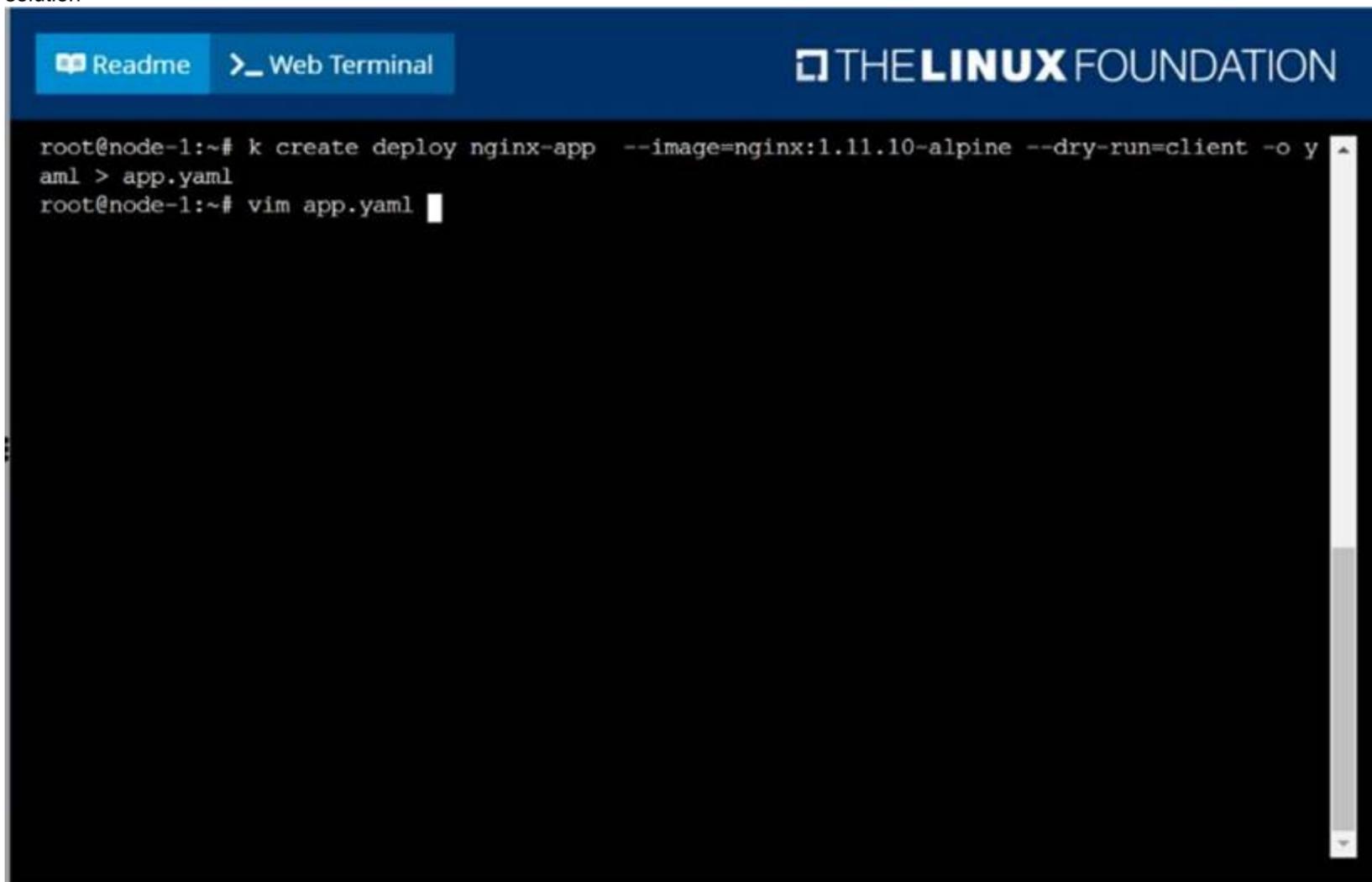
Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update.

Finally, rollback that update to the previous version 1.11.10-alpine.

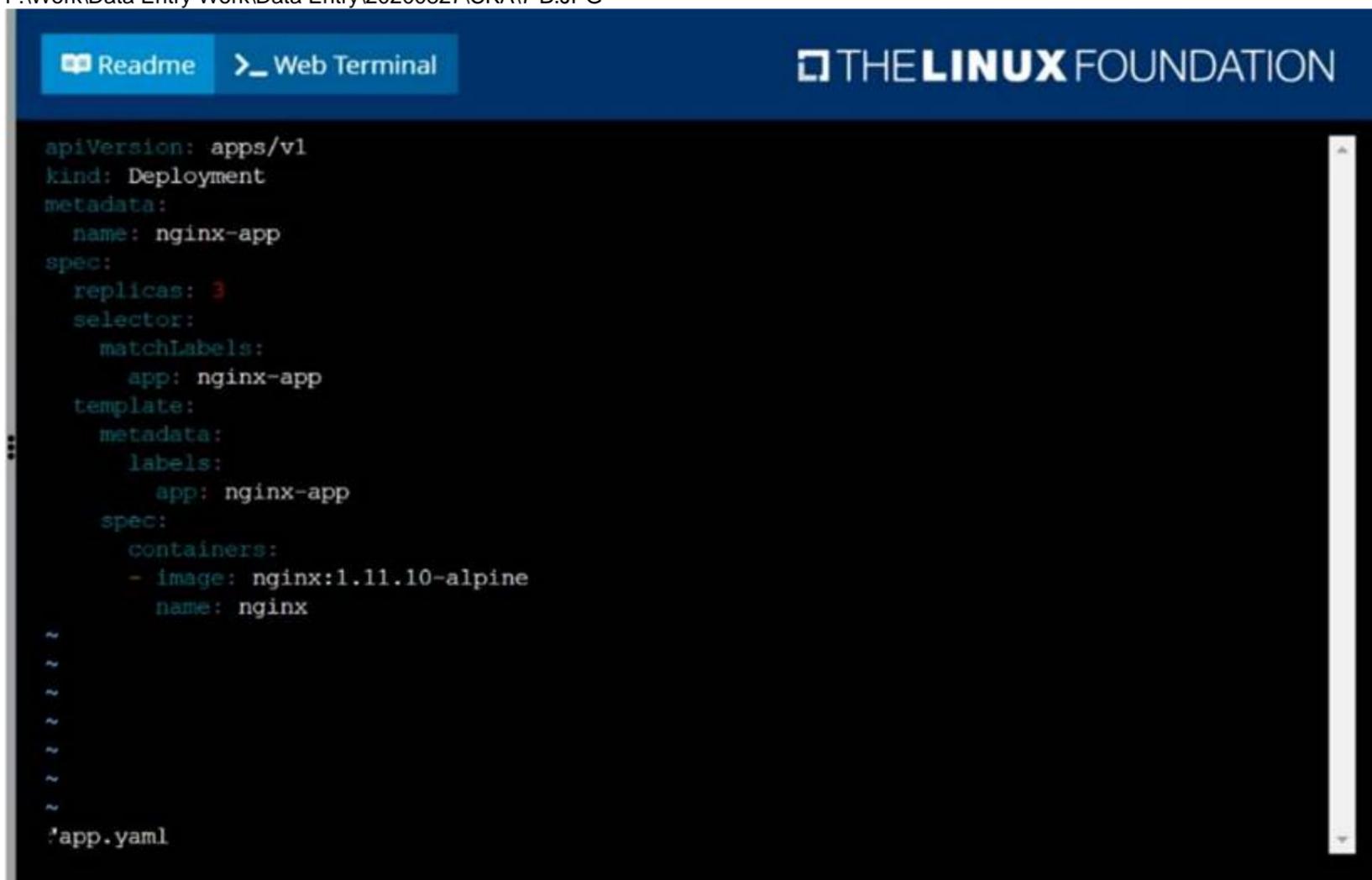
- A. Mastered
- B. Not Mastered

Answer: A

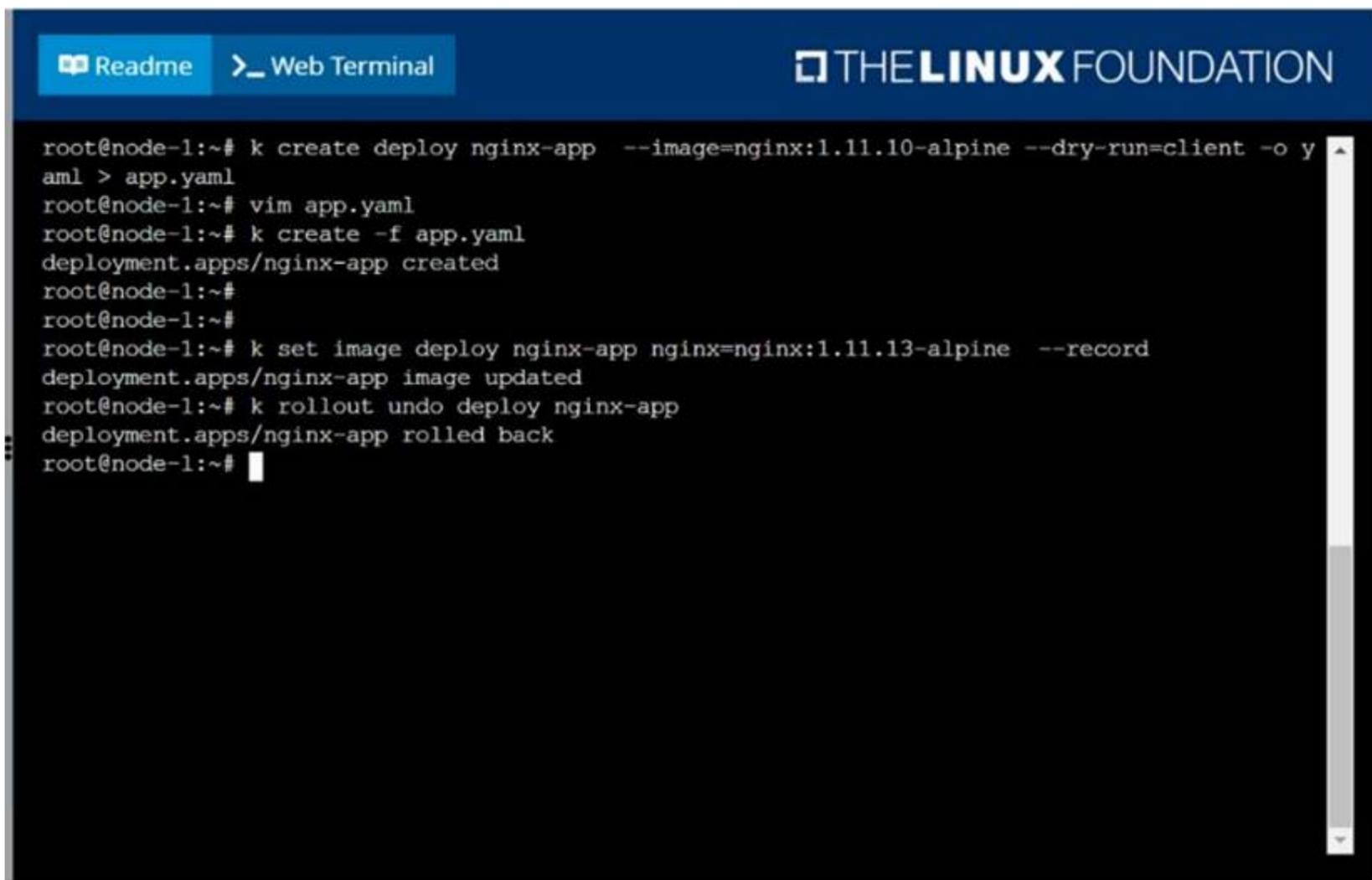
Explanation:  
solution



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



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



```
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:~#
```

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

#### NEW QUESTION 6

CORRECT TEXT

Ensure a single instance of pod nginx is running on each node of the Kubernetes cluster where nginx also represents the Image name which has to be used. Do not override any taints currently in place.

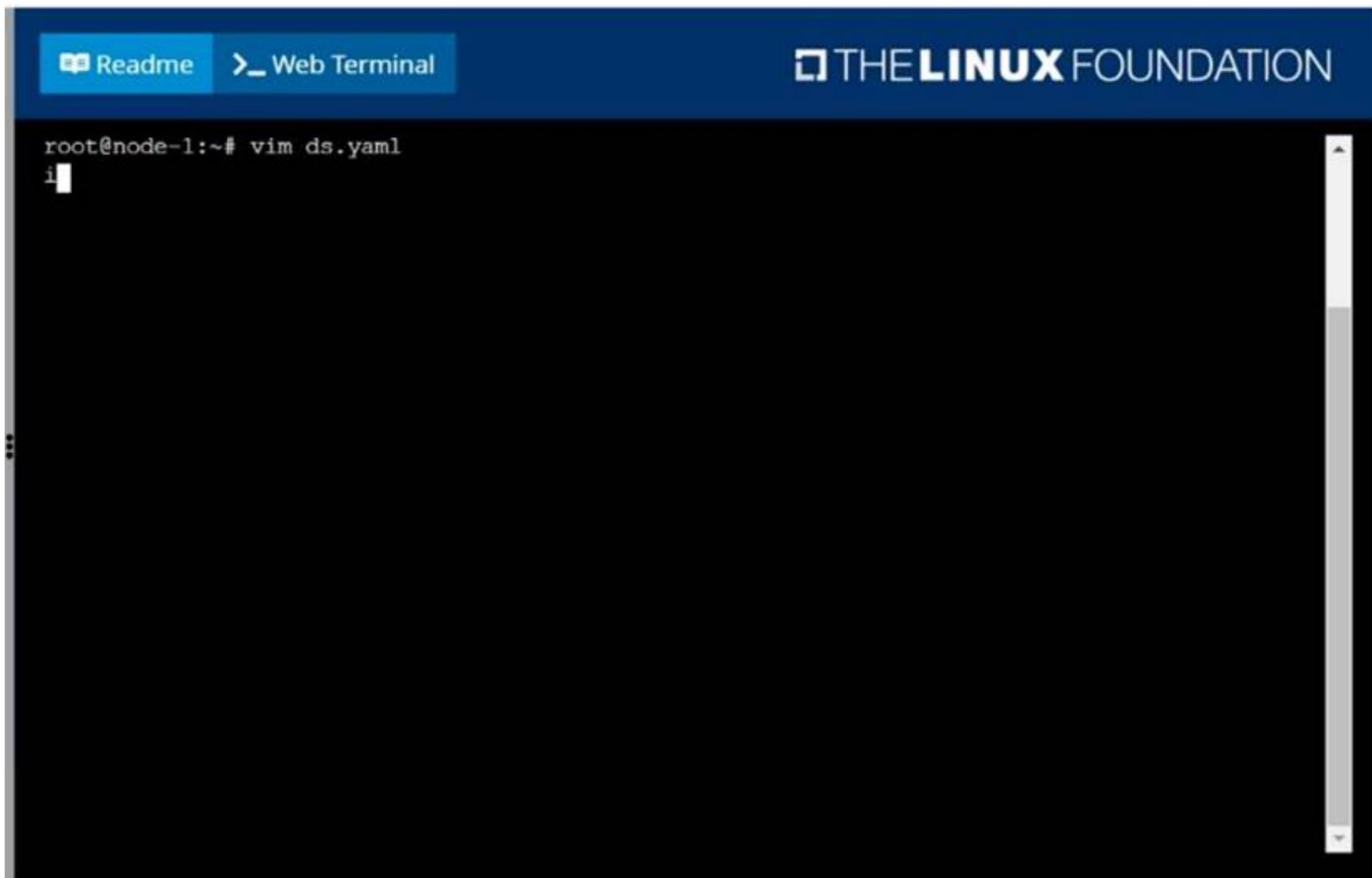
Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

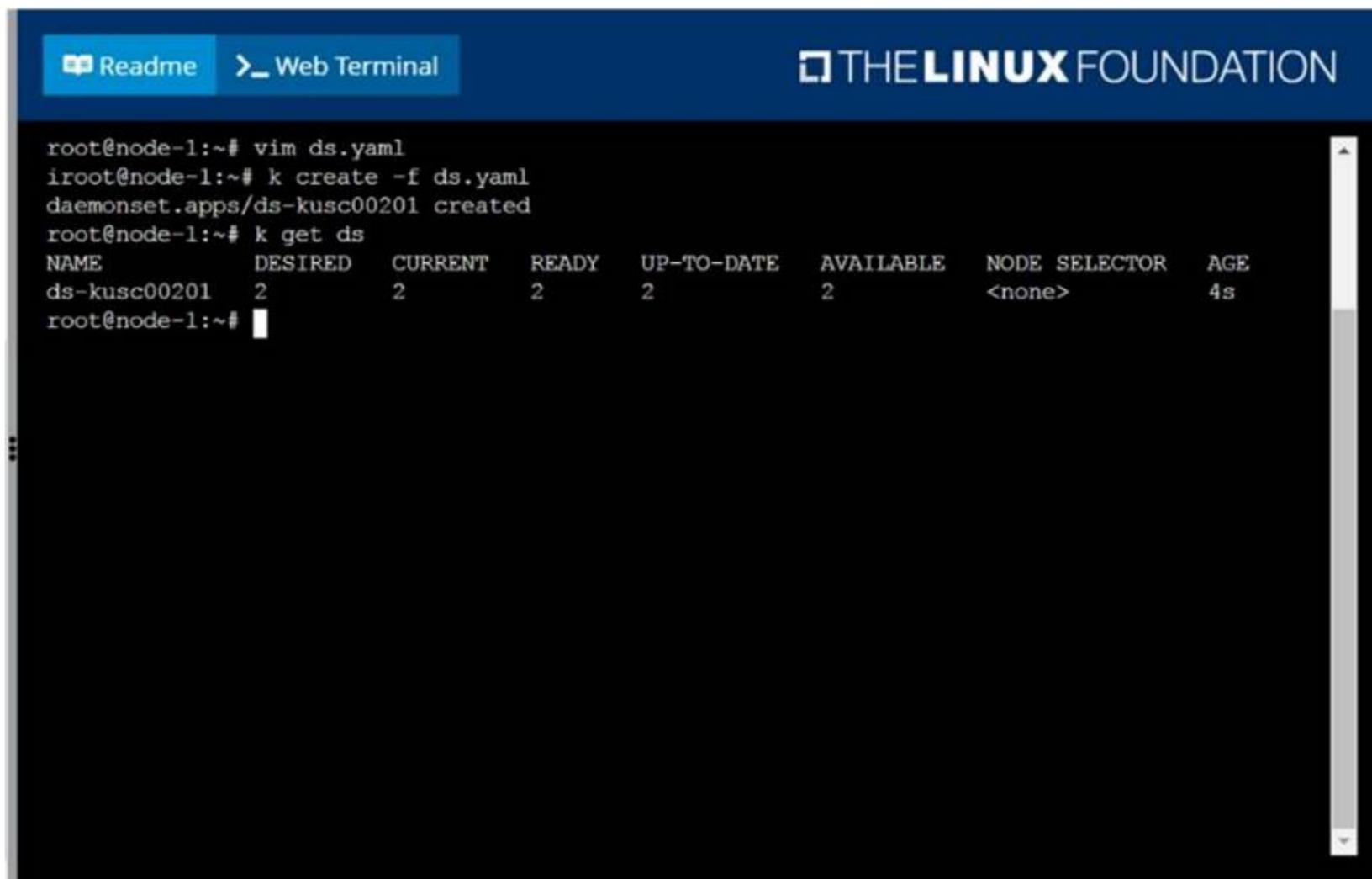
solution



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

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 B.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:~#
```

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

#### NEW QUESTION 7

CORRECT TEXT

Get list of all the pods showing name and namespace with a jsonpath expression.

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

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

#### NEW QUESTION 8

CORRECT TEXT

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.

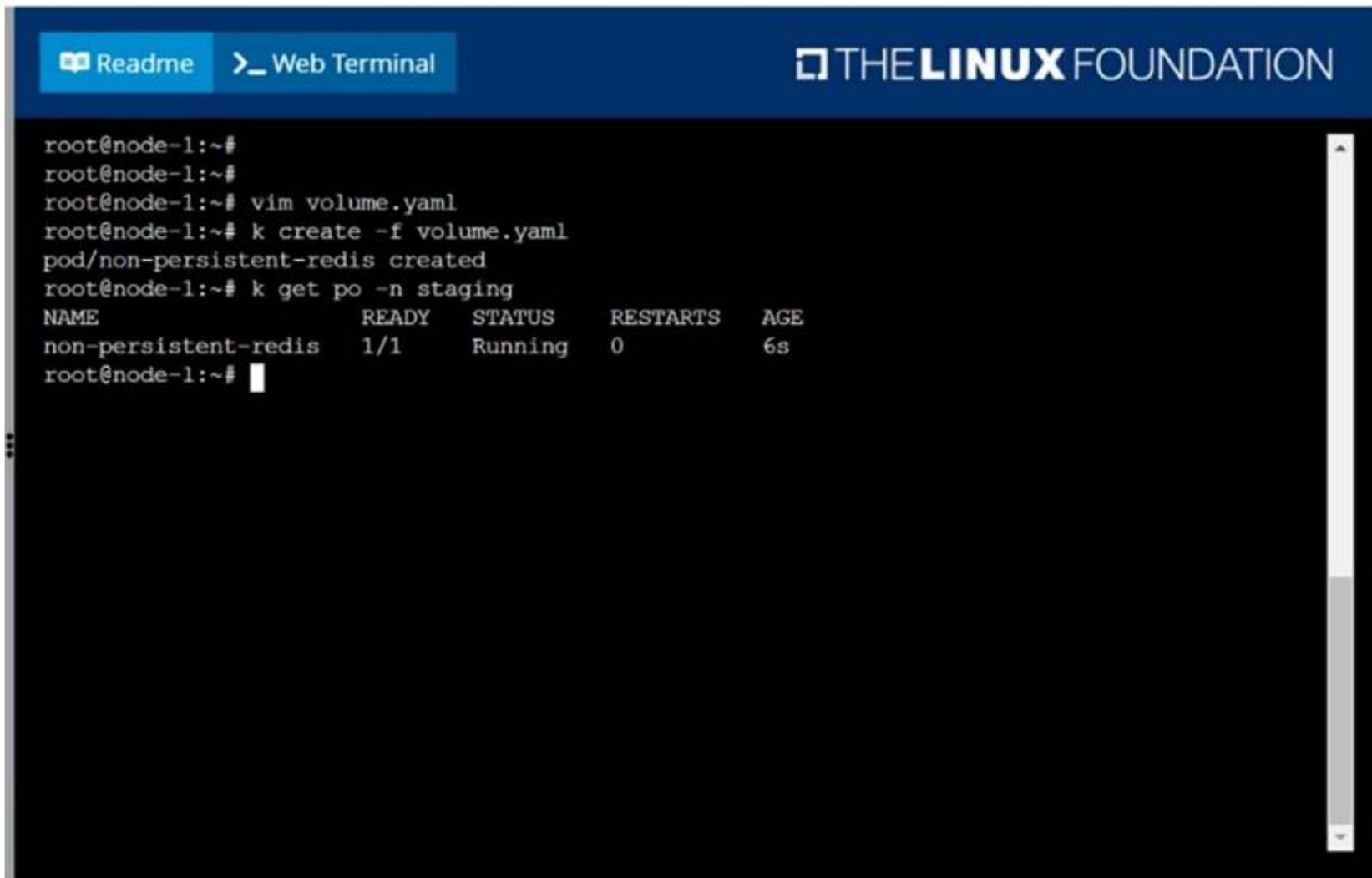
- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

solution





The screenshot shows a web terminal interface with a dark background. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active. The terminal content shows a series of commands and their outputs:

```
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:~#
```

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

#### NEW QUESTION 9

CORRECT TEXT

Score: 13%



Task

A Kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a Ready state, ensuring that any changes are made permanent.

You can ssh to the failed node using: 

```
[student@node-1] $ | ssh  
wk8s-node-0
```

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

```
[student@wk8s-node-0] $ |  
sudo -i
```

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

```
sudo -i  
systemctl status kubelet  
systemctl start kubelet  
systemctl enable kubelet
```

#### NEW QUESTION 10

CORRECT TEXT

Score: 4%

Set configuration context: 

```
[student@node-1] $ | kube  
ctl config use-context k  
8s
```

Task

Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified): nginx + redis + memcached .

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
Solution:
kubectl run kucc8 --image=nginx --dry-run -o yaml > kucc8.yaml
# vi kucc8.yaml
apiVersion: v1
kind: Pod
metadata:
creationTimestamp: null
name: kucc8
spec:
containers:
- image: nginx
name: nginx
- image: redis
name: redis
- image: memcached
name: memcached
- image: consul
name: consul
#
kubectl create -f kucc8.yaml
#12.07
```

**NEW QUESTION 10****CORRECT TEXT**

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 httpd named 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:

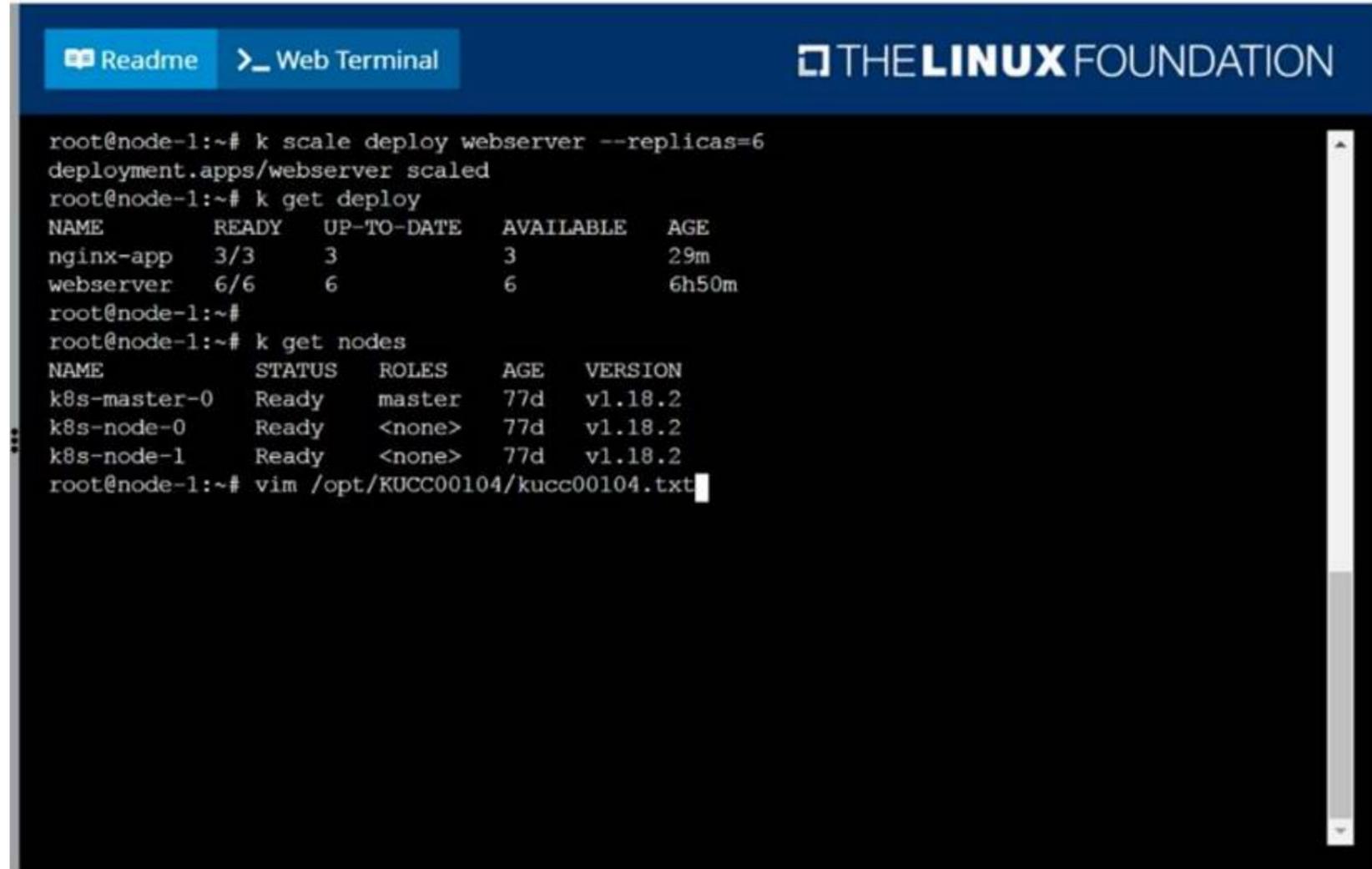
```
[student@wk8s-node-1] $ | sudo -i
```

- A. Mastered
- B. Not Mastered

**Answer:** A

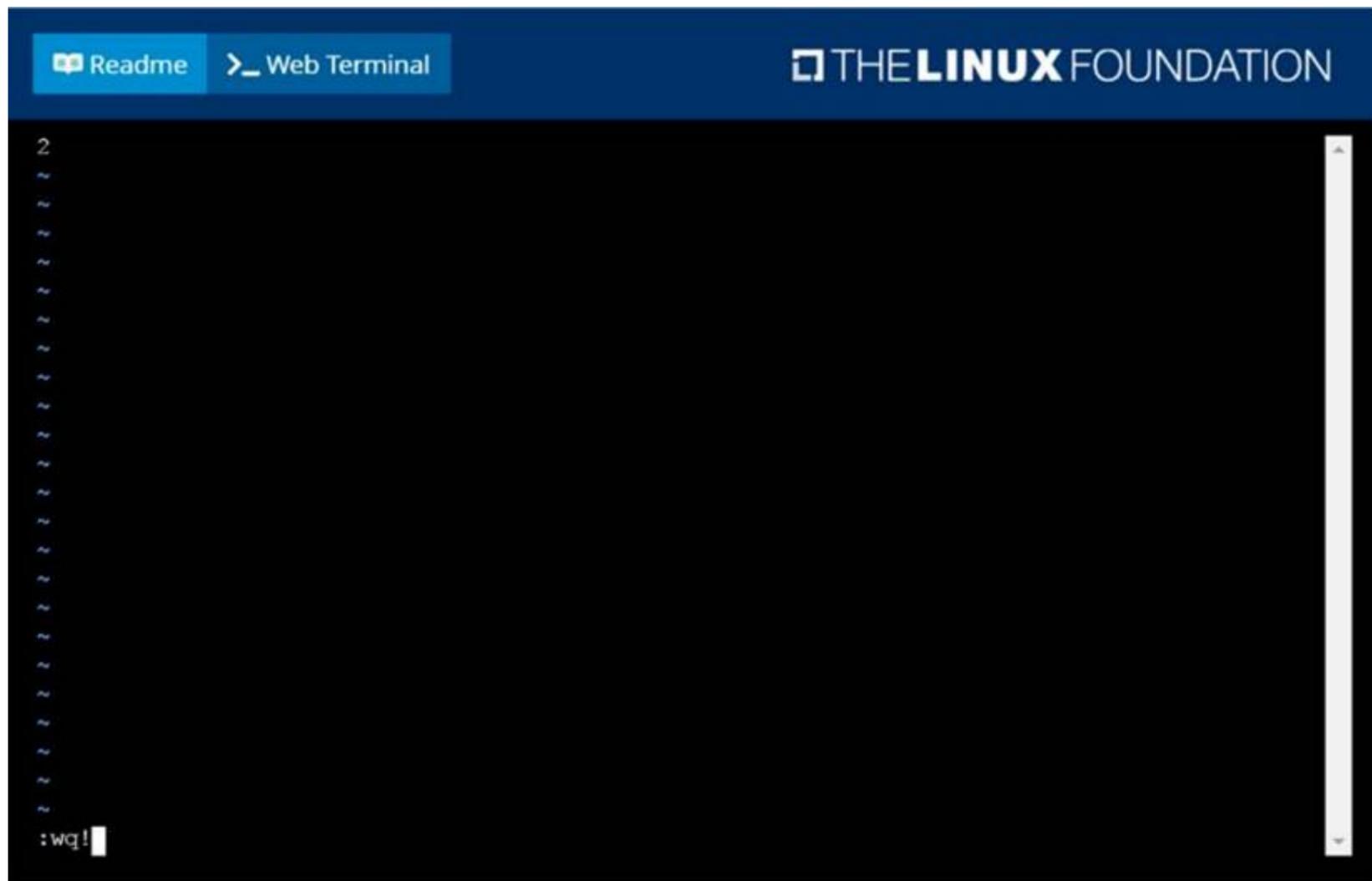
**Explanation:**

solution



```
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 B.JPG



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

#### NEW QUESTION 12

CORRECT TEXT

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubect! get pods --sort-by=.metadata.name

#### NEW QUESTION 15

CORRECT TEXT

List all persistent volumes sorted by capacity, saving the full kubect! output to /opt/KUCC00102/volume\_list. Use kubect! 's own functionality for sorting the output, and do not manipulate it any further.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

```
Readme Web Terminal THE LINUX FOUNDATION

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:~#
```

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

#### NEW QUESTION 18

CORRECT TEXT

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 Persistent Volume

```
kind: PersistentVolumeapiVersion: v1metadata: name:app-dataspec: 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 classname shared, 2Gi of storage capacity and the host path /srv/app-data.



### NEW QUESTION 21

CORRECT TEXT

List the nginx pod with custom columns POD\_NAME and POD\_STATUS

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

### NEW QUESTION 26

CORRECT TEXT

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

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
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
```

### NEW QUESTION 27

CORRECT TEXT

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 31

.....

## 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