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
CORRECT TEXT
List all the pods sorted by name

A. Mastered
B. Not Mastered

Answer: A

Explanation:
```
kubect1 get pods --sort-by=.metadata.name
```

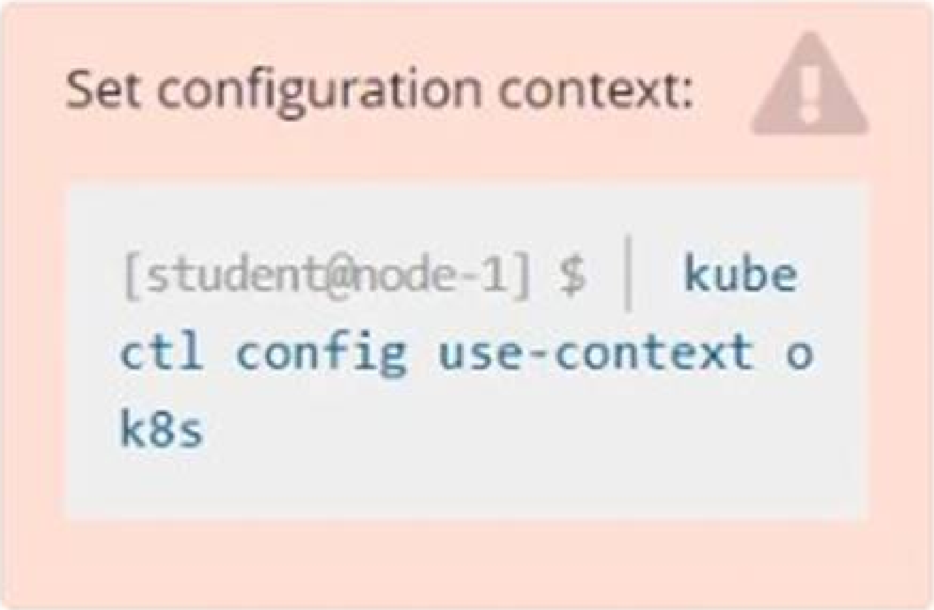
NEW QUESTION 2
CORRECT TEXT
Create a namespace called 'development' and a pod with image nginx called nginx on this namespace.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
```
kubectl create namespace development
kubectl run nginx --image=nginx --restart=Never -n development
```

NEW QUESTION 3
CORRECT TEXT
Score:7%



Set configuration context:

```
[student@node-1] $ kubectl config use-context ok8s
```

Task
Create a new PersistentVolumeClaim
- Name: pv-volume
- Class: csi-hostpath-sc
- Capacity: 10Mi

Create a new Pod which mounts the PersistentVolumeClaim as a volume:
- Name: web-server
- Image: nginx
- Mount path: /usr/share/nginx/html

Configure the new Pod to have ReadWriteOnce access on the volume.
Finally, using kubectl edit or kubectl patch expand the PersistentVolumeClaim to a capacity of 70Mi and record that change.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

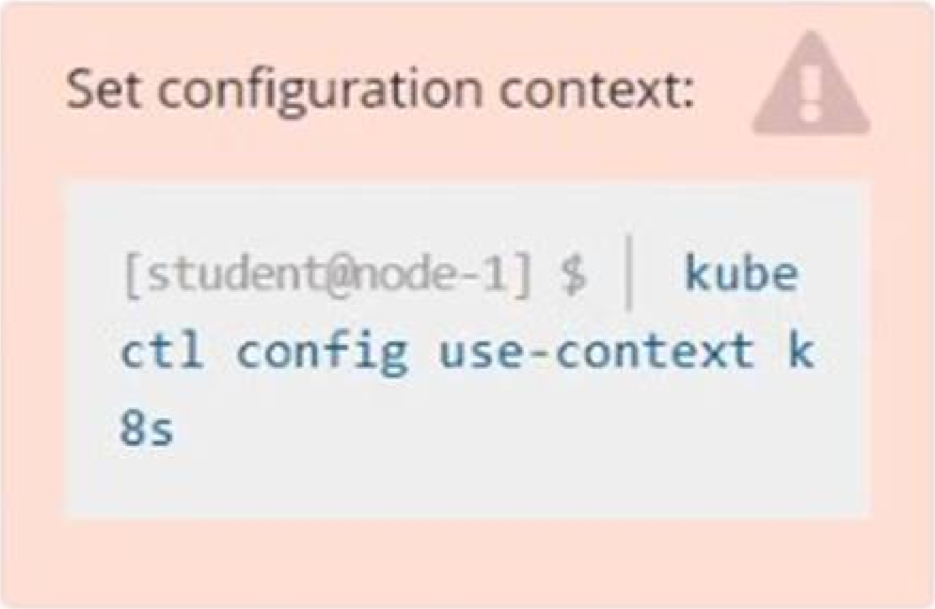
```
vi pvc.yaml
storageclass pvc
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: pv-volume
spec:
accessModes:
- ReadWriteOnce
volumeMode: Filesystem
resources:
requests:
storage: 10Mi
```

```
storageClassName: csi-hostpath-sc
# vi pod-pvc.yaml
apiVersion: v1
kind: Pod
metadata:
name: web-server
spec:
containers:
- name: web-server
image: nginx
volumeMounts:
- mountPath: "/usr/share/nginx/html"
name: my-volume
volumes:
- name: my-volume
persistentVolumeClaim:
claimName: pv-volume
# craete
kubectl create -f pod-pvc.yaml
#edit
kubectl edit pvc pv-volume --record
```

NEW QUESTION 4

CORRECT TEXT

Score: 4%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context k8s
```

Task

Check to see how many nodes are ready (not including nodes tainted NoSchedule) and write the number to /opt/KUSC00402/kusc00402.txt.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:

```
kubectl describe nodes | grep ready|wc -l
kubectl describe nodes | grep -i taint | grep -i noschedule |wc -l
echo 3 > /opt/KUSC00402/kusc00402.txt
#
kubectl get node | grep -i ready |wc -l
# taintsnoSchedule
kubectl describe nodes | grep -i taints | grep -i noschedule |wc -l
#
echo 2 > /opt/KUSC00402/kusc00402.txt
```

NEW QUESTION 5

CORRECT TEXT

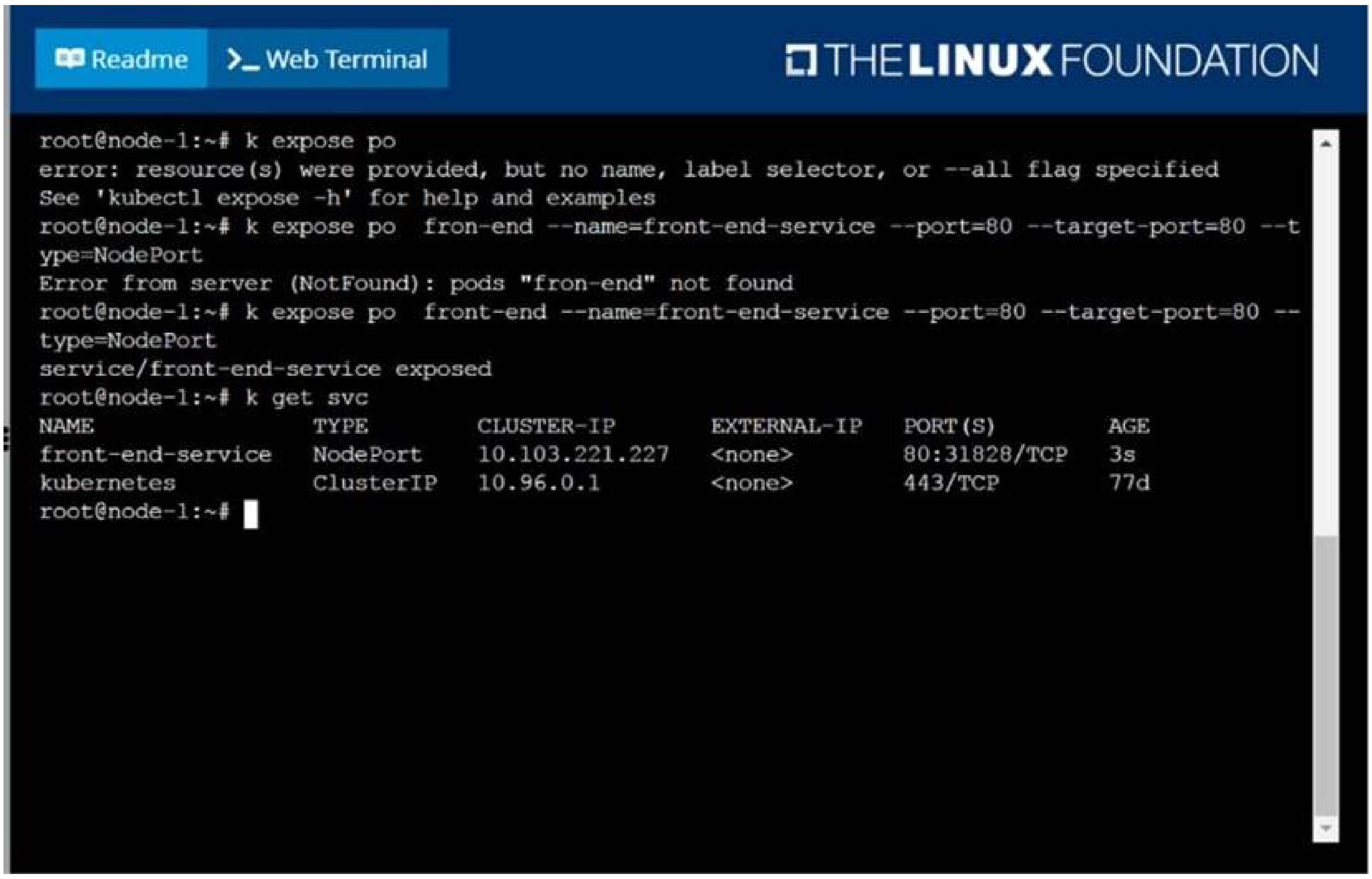
Create and configure the service front-end-service so it's accessible through NodePort and routes to the existing pod named front-end.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

solution



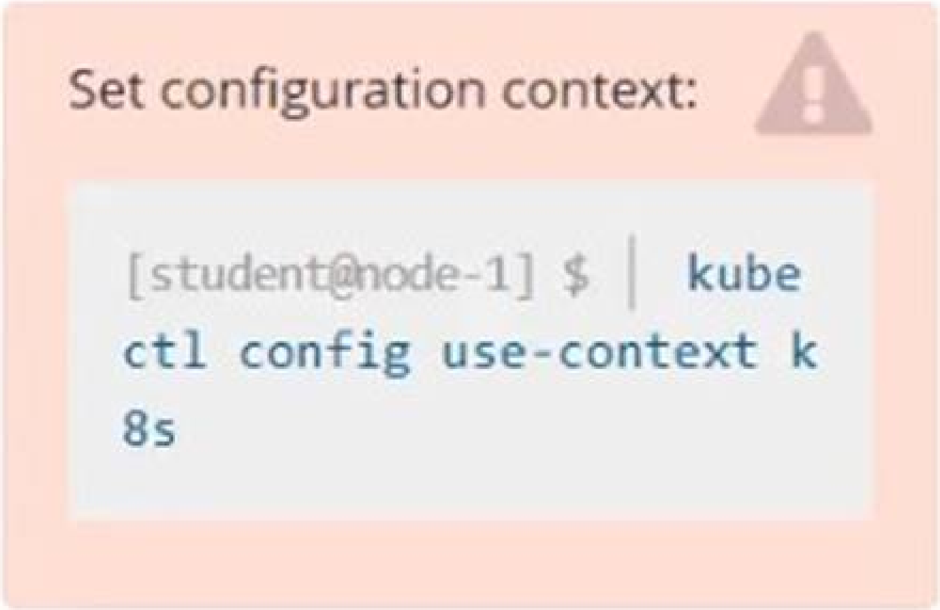
```
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po  fron-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "fron-end" not found
root@node-1:~# k expose po  front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
front-end-service   NodePort    10.103.221.227   <none>        80:31828/TCP   3s
kubernetes          ClusterIP   10.96.0.1        <none>        443/TCP        77d
root@node-1:~#
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\8 B.JPG

NEW QUESTION 6

CORRECT TEXT

Score: 5%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context k8s
```

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).

A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:

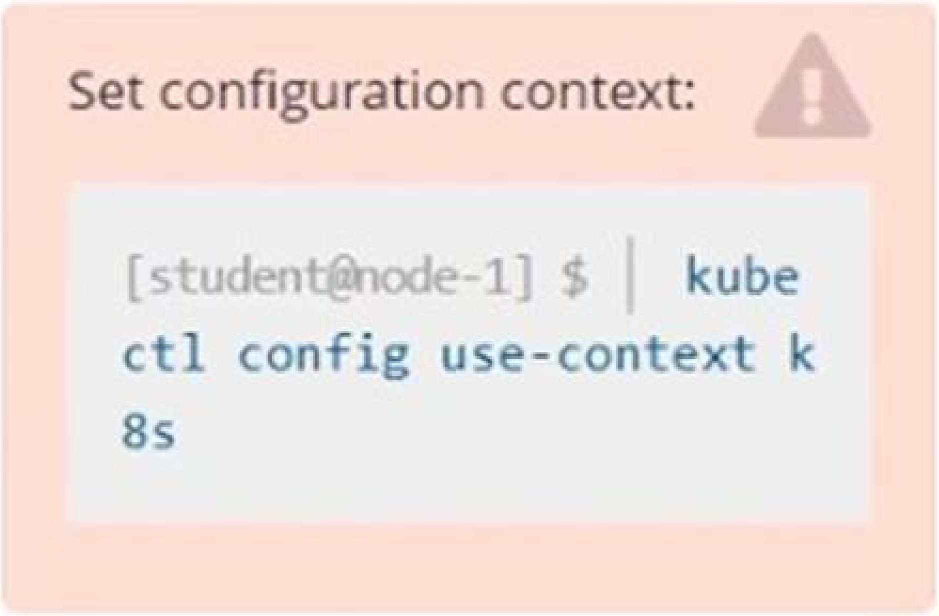
kubectl top -l name=cpu-user -A

echo 'pod name' >> /opt/KUT00401/KUT00401.txt

NEW QUESTION 7

CORRECT TEXT

Score: 4%



Set configuration context:

```
[student@node-1] $ kubectl config use-context k8s
```

Context
You have been asked to create a new ClusterRole for a deployment pipeline and bind it to a specific ServiceAccount scoped to a specific namespace.
Task
Create a new ClusterRole named deployment-clusterrole, which only allows to create the following resource types:
- Deployment
- StatefulSet
- DaemonSet

Create a new ServiceAccount named cicd-token in the existing namespace app-team1. Bind the new ClusterRole deployment-clusterrole lo the new ServiceAccount cicd-token ,
limited to the namespace app-team1.

A. Mastered
B. Not Mastered

Answer: A

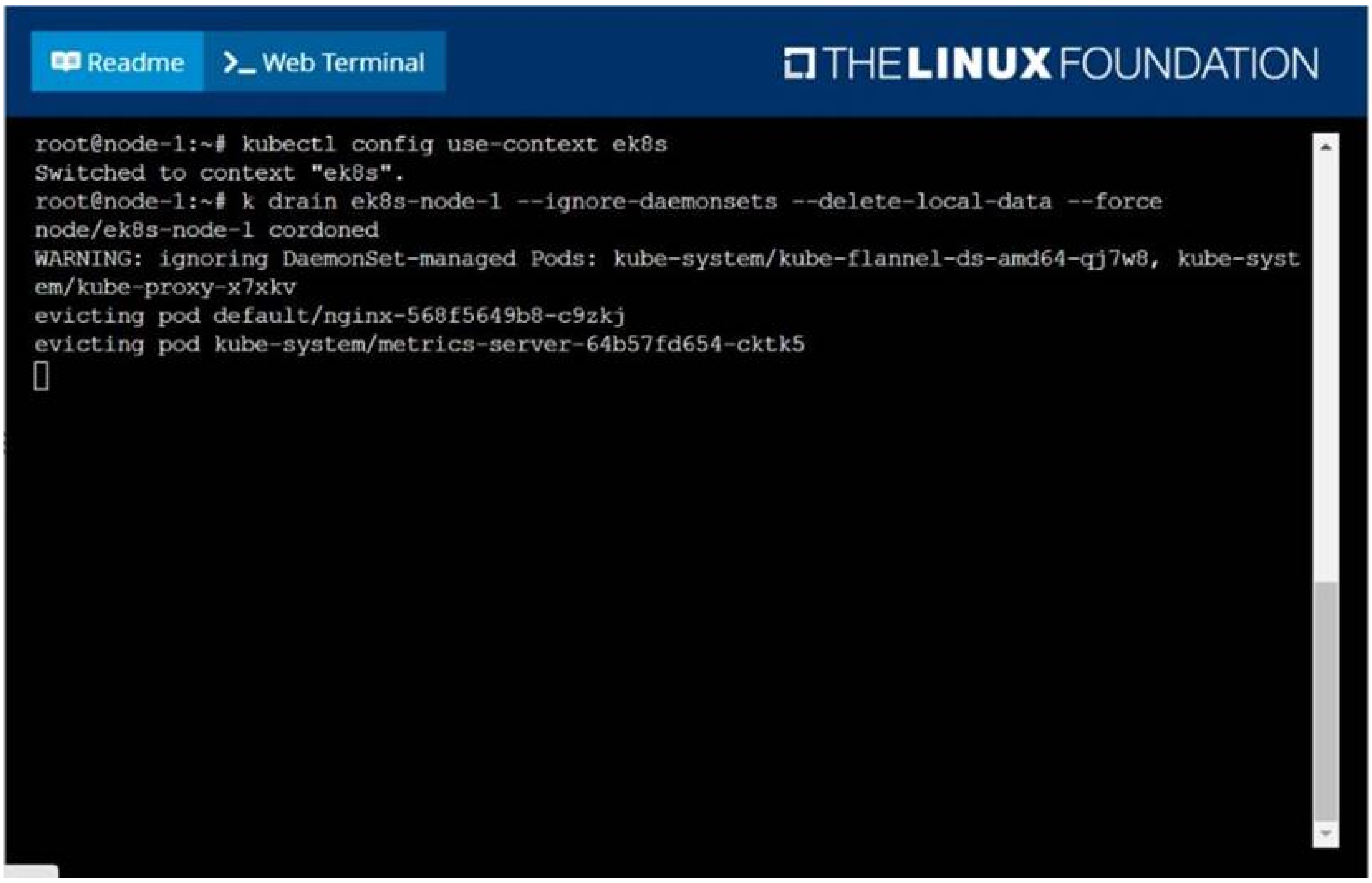
Explanation:
Solution:
Task should be complete on node k8s -1 master, 2 worker for this connect use command [student@node-1] > ssh k8s
kubectl create clusterrole deployment-clusterrole --verb=create -- resource=deployments,statefulsets,daemonsets
kubectl create serviceaccount cicd-token --namespace=app-team1
kubectl create rolebinding deployment-clusterrole --clusterrole=deployment-clusterrole -- serviceaccount=default:cicd-token --namespace=app-team1

NEW QUESTION 8
CORRECT TEXT
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



```
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-syst
em/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zkj
evicting pod kube-system/metrics-server-64b57fd654-cktk5
```

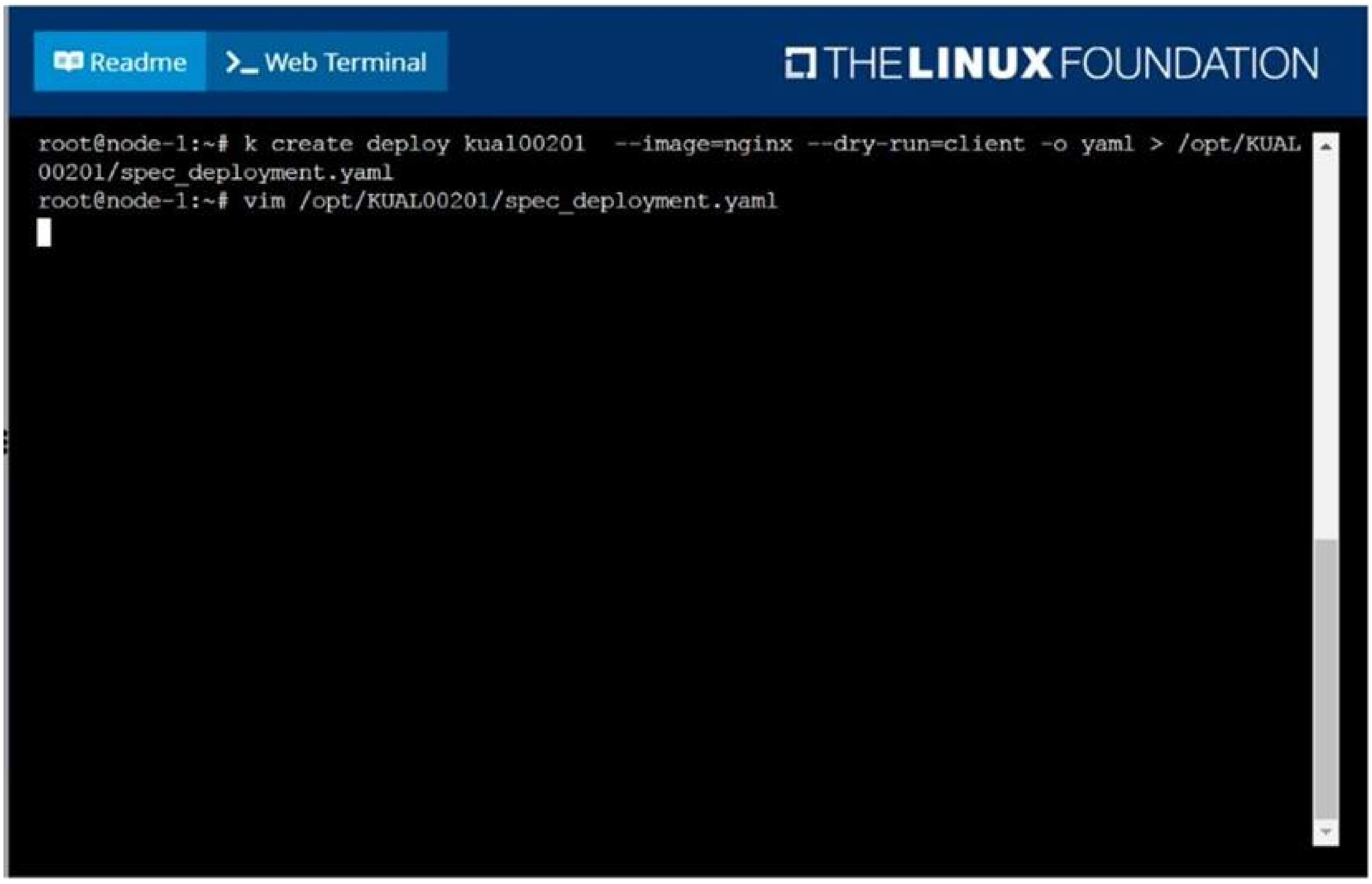
F:\Work\Data Entry Work\Data Entry\20200827\CKA\19 B.JPG

NEW QUESTION 9
CORRECT TEXT
Create a deployment spec file that will:
? Launch 7 replicas of the nginx Image with the labelapp_runtime_stage=dev
? deployment name: kual00201
Save a copy of this spec file to /opt/KUAL00201/spec_deployment.yaml
(or /opt/KUAL00201/spec_deployment.json).
When you are done, clean up (delete) any new Kubernetes API object that you produced during this task.

A. Mastered
B. Not Mastered

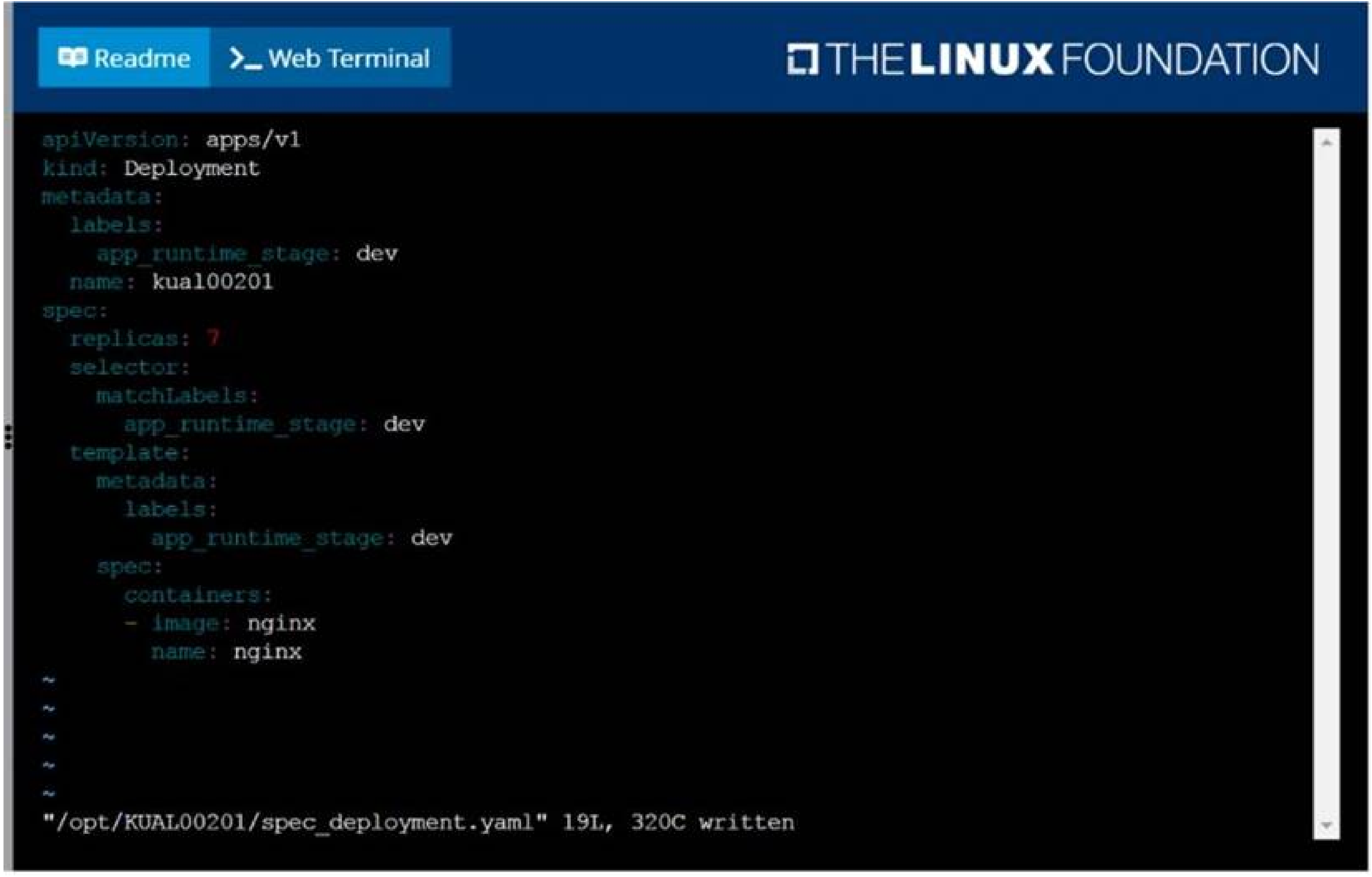
Answer: A

Explanation:
solution



```
root@node-1:~# k create deploy kual00201  --image=nginx --dry-run=client -o yaml > /opt/KUAL
00201/spec_deployment.yaml
root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\10 B.JPG



```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app_runtime_stage: dev
  name: kual00201
spec:
  replicas: 7
  selector:
    matchLabels:
      app_runtime_stage: dev
  template:
    metadata:
      labels:
        app_runtime_stage: dev
    spec:
      containers:
      - image: nginx
        name: nginx
~
~
~
~
~
"/opt/KUAL00201/spec_deployment.yaml" 19L, 320C written
```

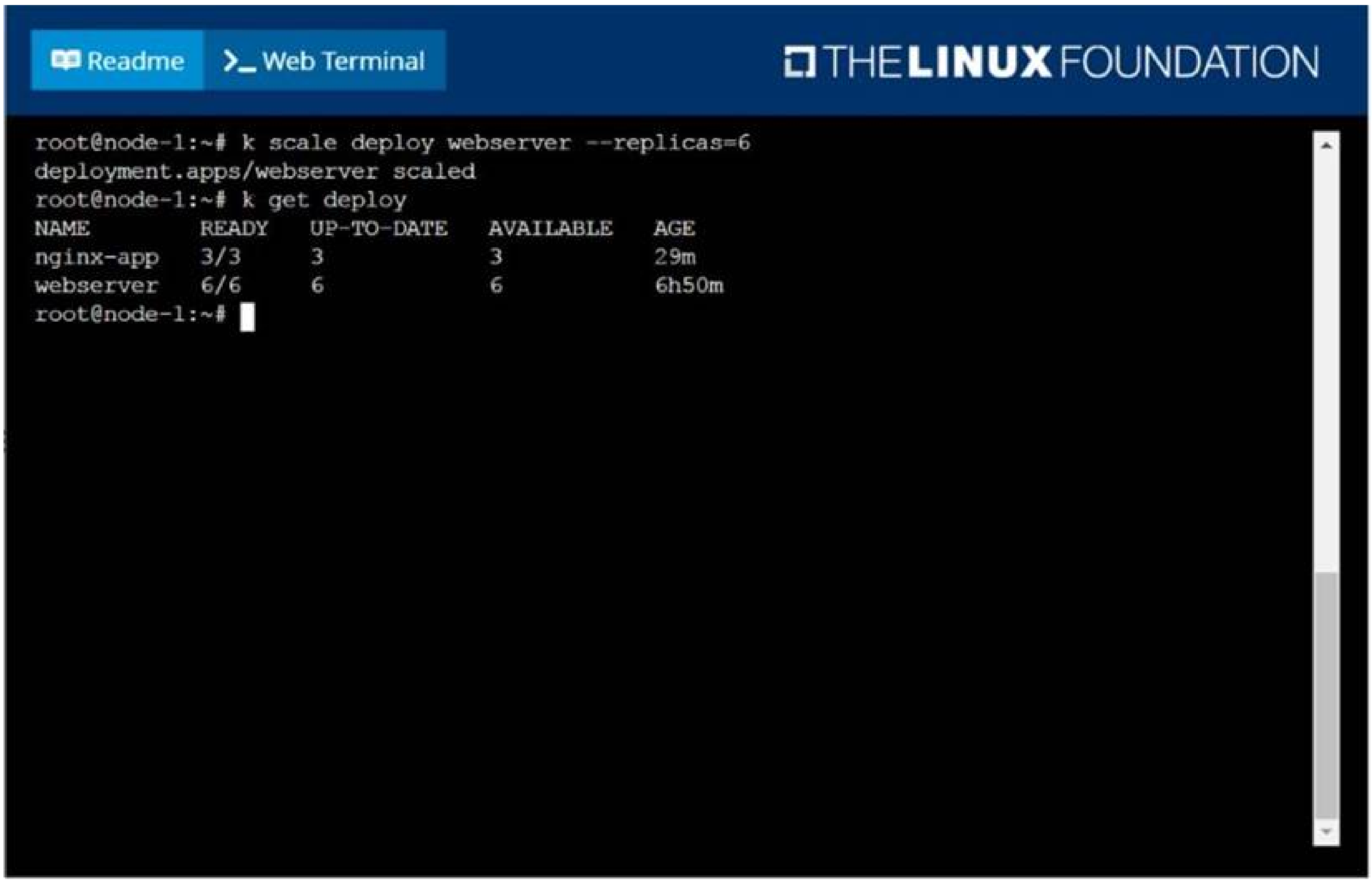
F:\Work\Data Entry Work\Data Entry\20200827\CKA\10 C.JPG

NEW QUESTION 10
CORRECT TEXT
Scale the deployment webserver to 6 pods.

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:~#
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\14 B.JPG

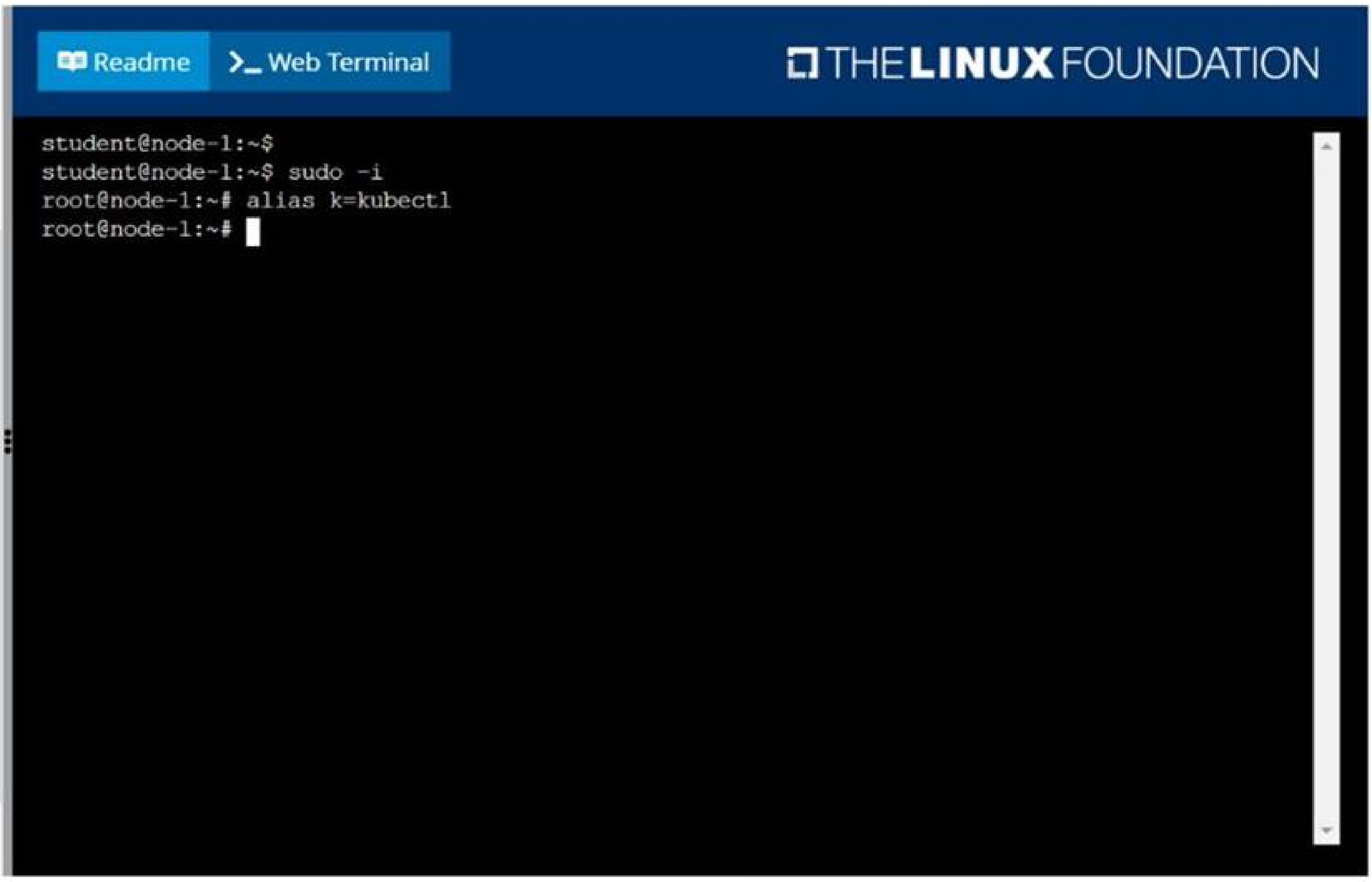
NEW QUESTION 10

CORRECT TEXT
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

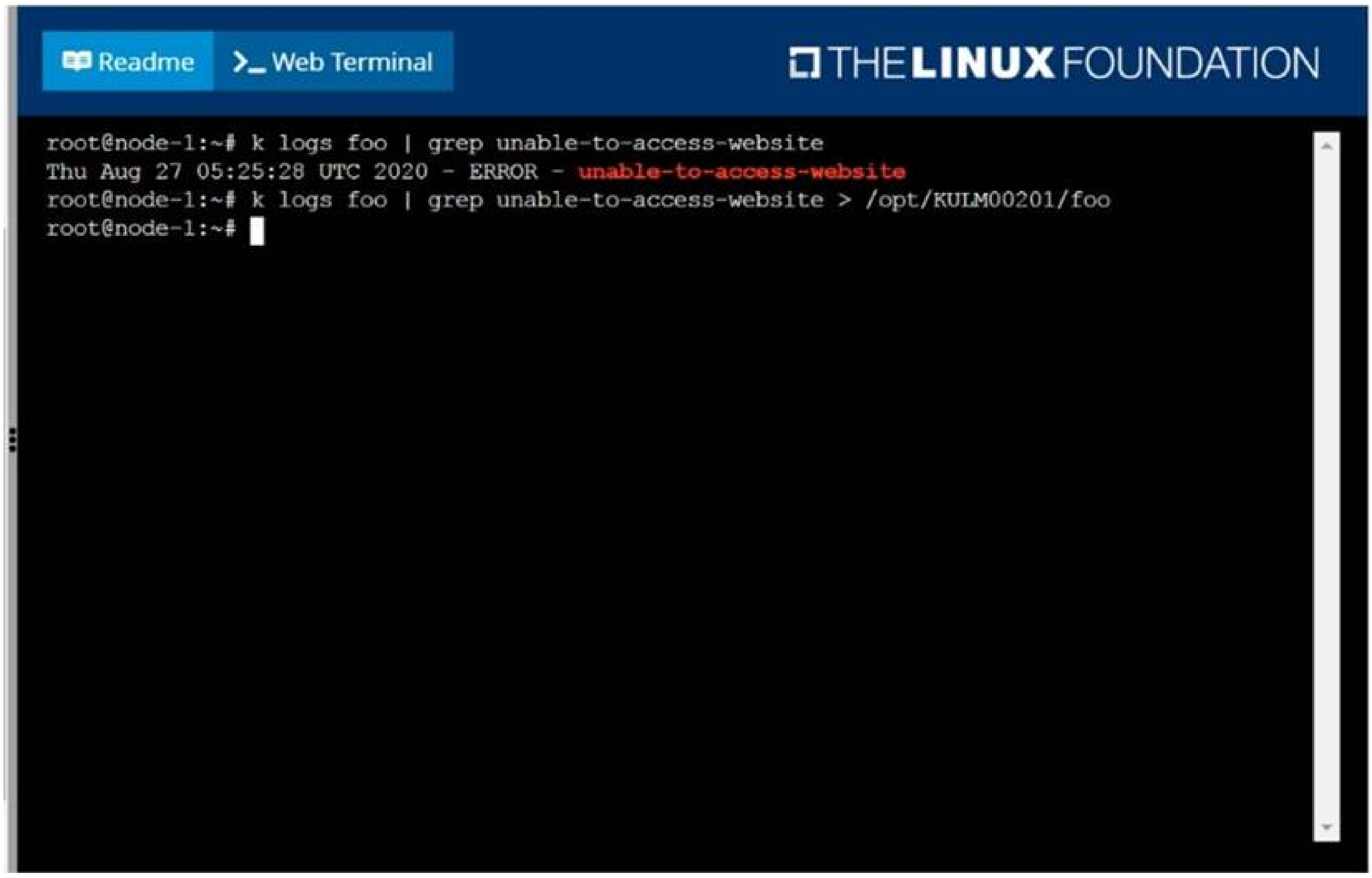
Answer: A

Explanation:
solution



```
student@node-1:~$
student@node-1:~$ sudo -i
root@node-1:~# alias k=kubectl
root@node-1:~#
```

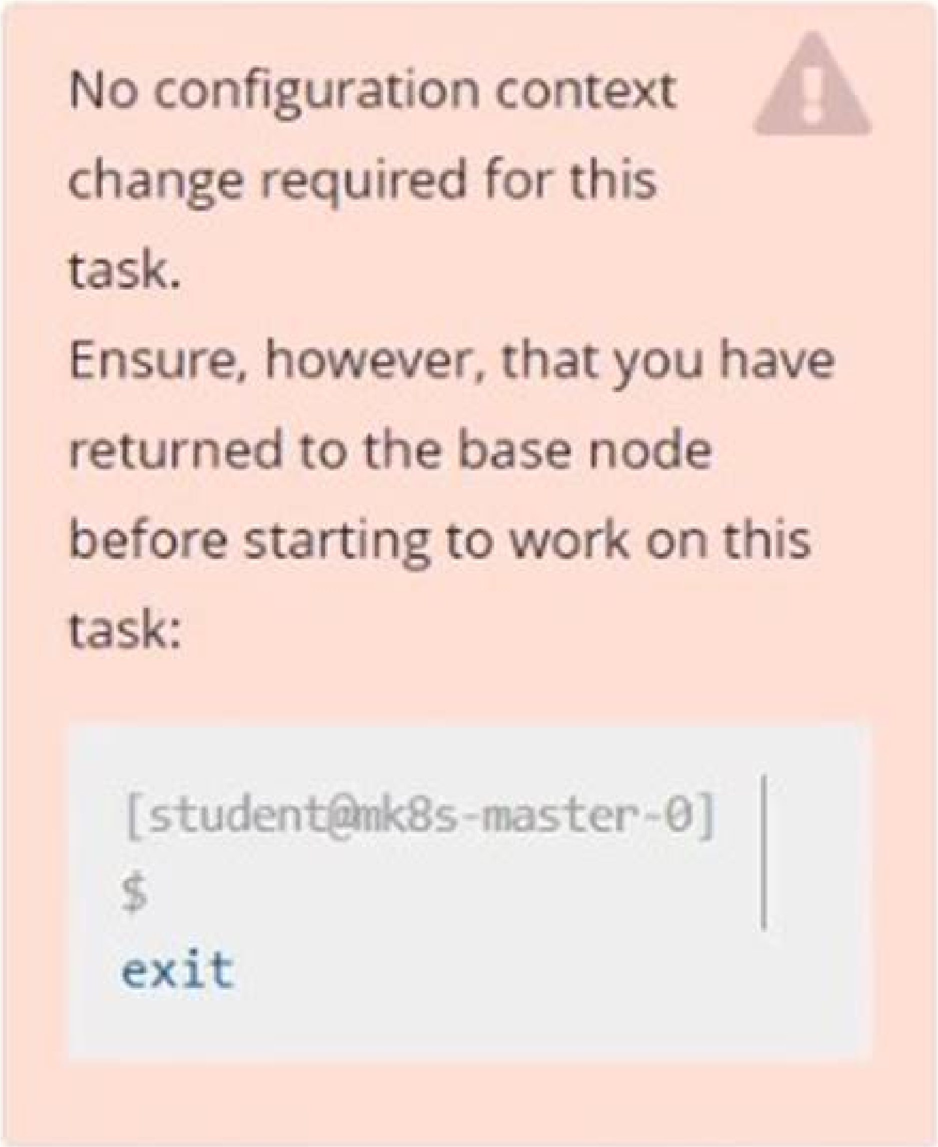
F:\Work\Data Entry Work\Data Entry\20200827\CKA\1 B.JPG



```
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:~#
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\1 C.JPG

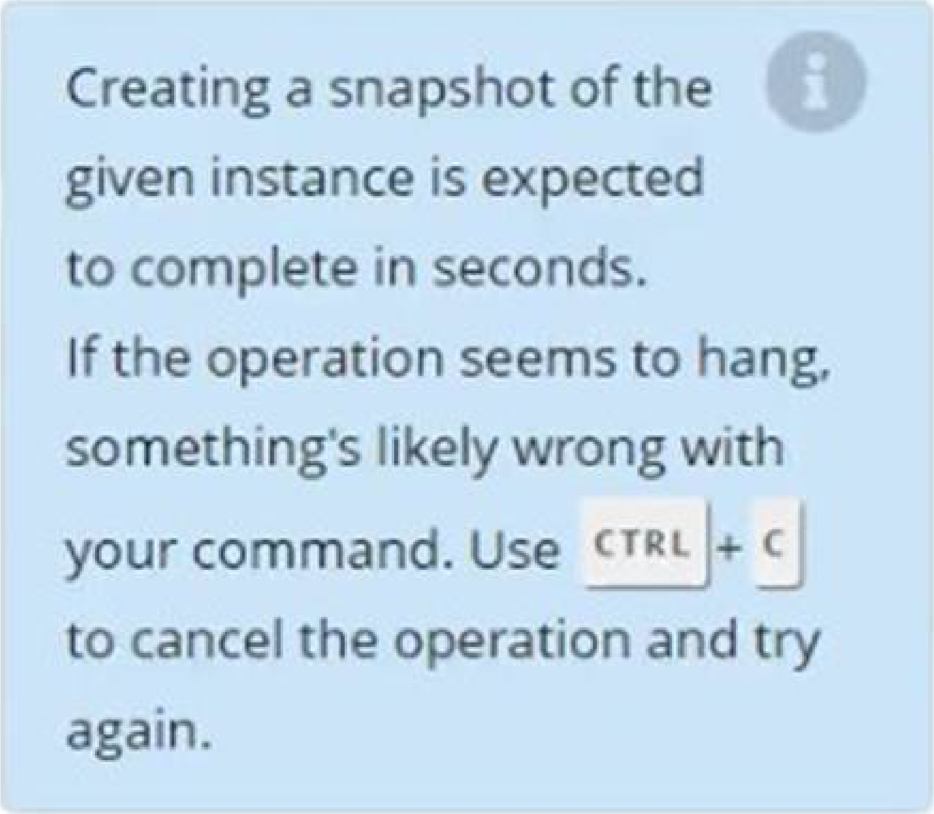
NEW QUESTION 13
CORRECT TEXT
Score: 7%


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:

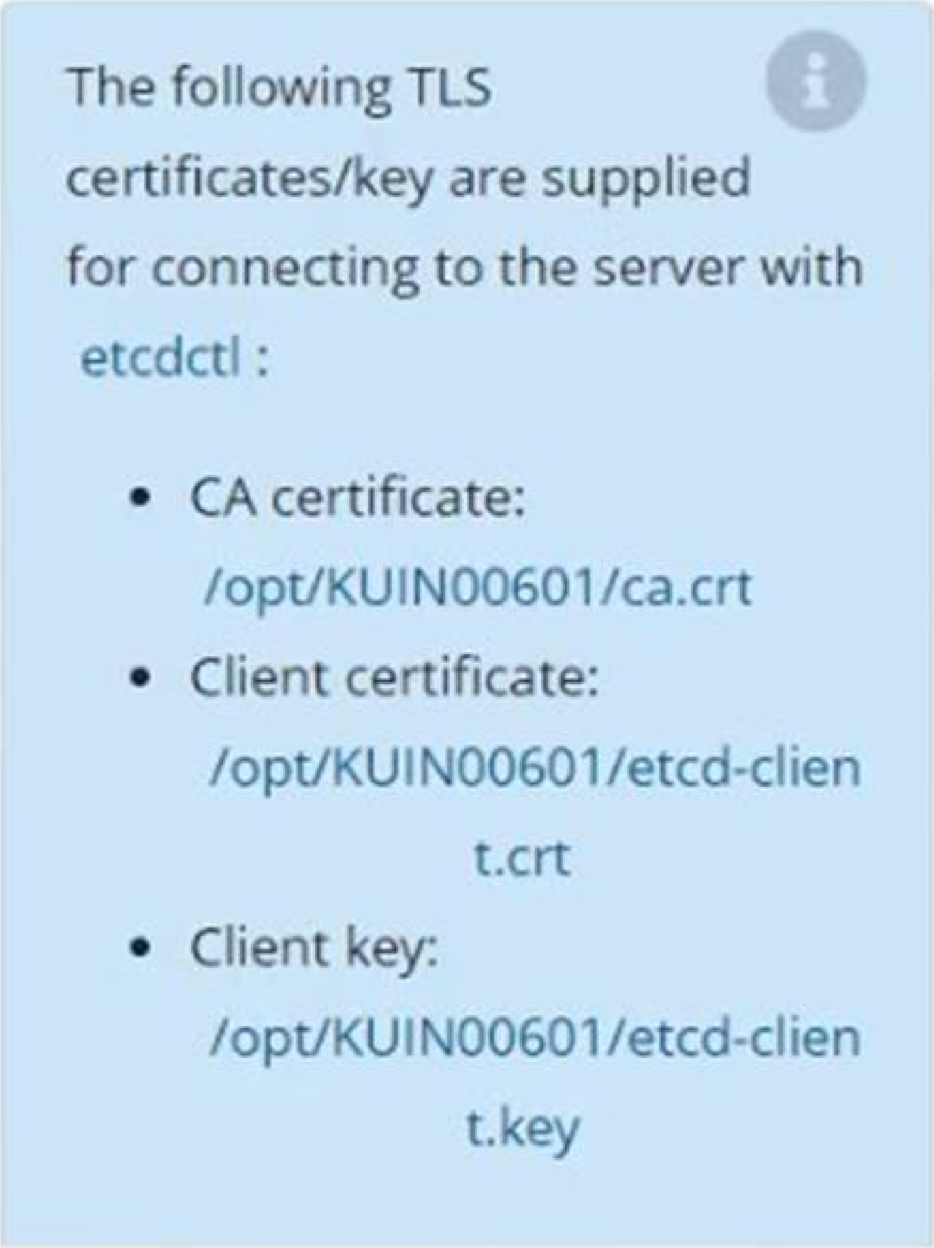
```
[student@mk8s-master-0] $ exit
```

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-previo 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/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
```

NEW QUESTION 17
CORRECT TEXT
Check the Image version of nginx-dev pod using jsonpath

A. Mastered
B. Not Mastered

Answer: A

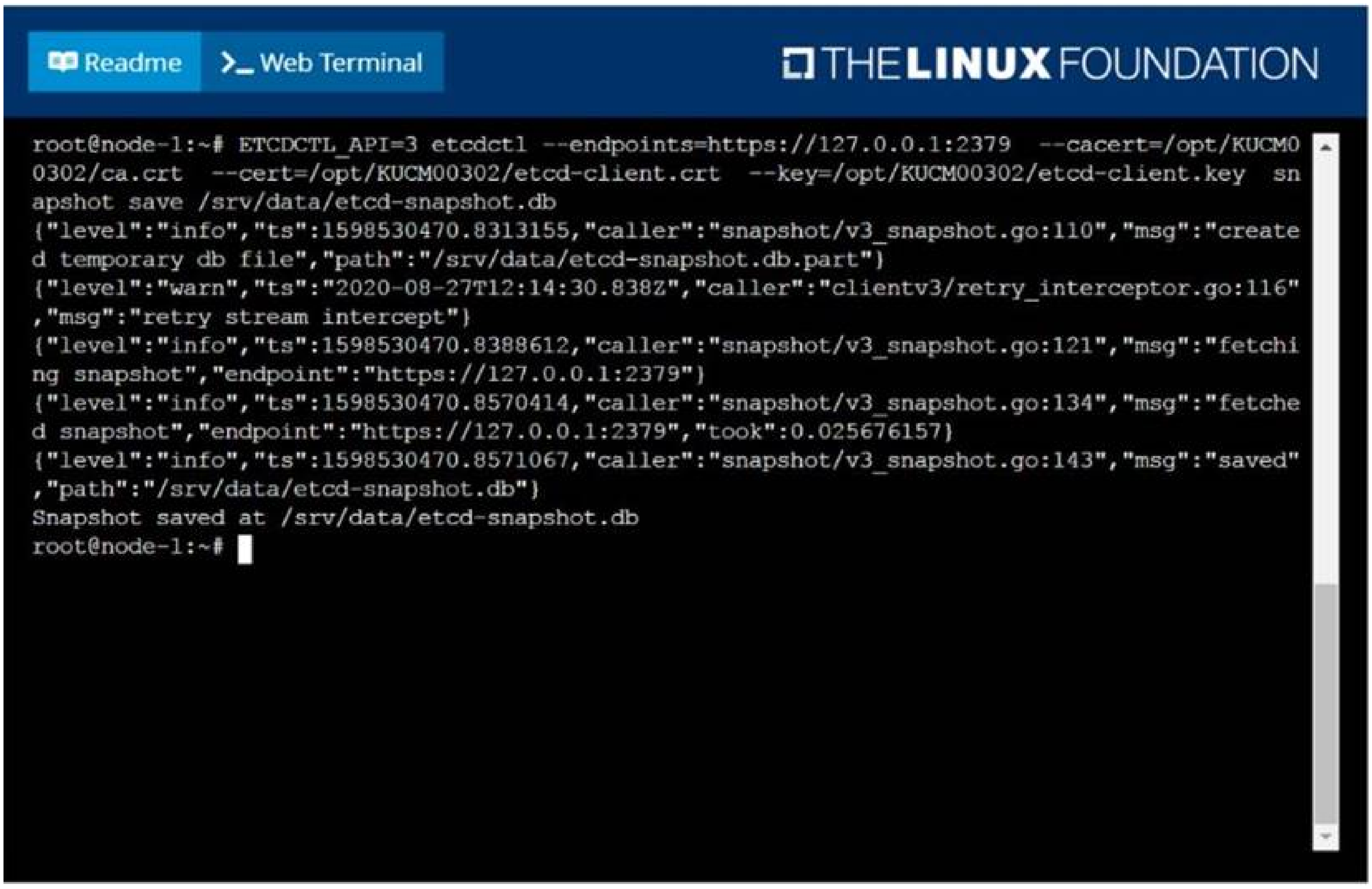
Explanation:
kubect1 get po nginx-dev -o
jsonpath='{.spec.containers[].image}{"\n"}'

NEW QUESTION 19
CORRECT TEXT
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: Topt/KUCM00302/etcd-client.key

A. Mastered
B. Not Mastered

Answer: A

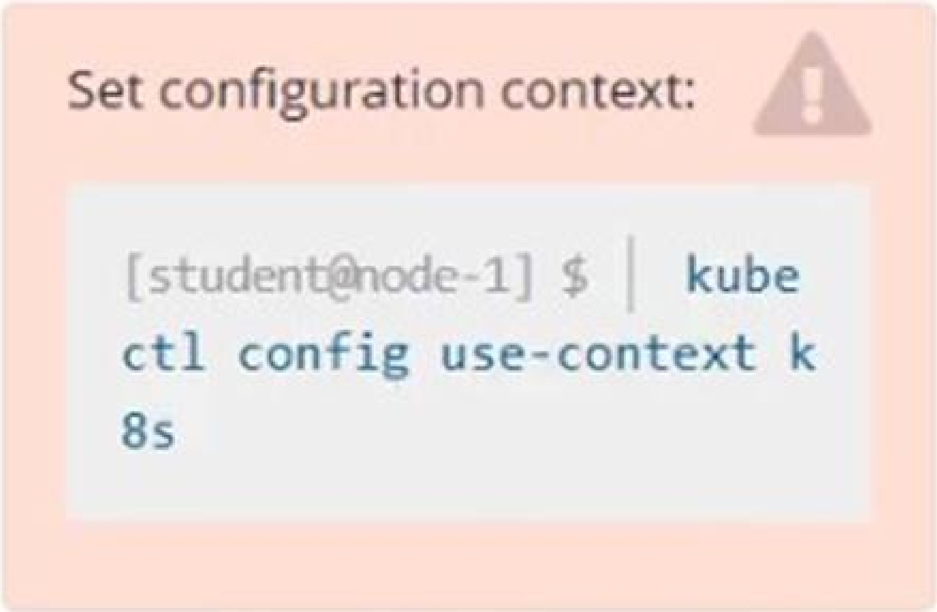
Explanation:
solution



```
root@node-1:~# ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379  --cacert=/opt/KUCM0
0302/ca.crt  --cert=/opt/KUCM00302/etcd-client.crt  --key=/opt/KUCM00302/etcd-client.key  sn
apshot save /srv/data/etcd-snapshot.db
{"level":"info","ts":1598530470.8313155,"caller":"snapshot/v3_snapshot.go:110","msg":"create
d temporary db file","path":"/srv/data/etcd-snapshot.db.part"}
{"level":"warn","ts":"2020-08-27T12:14:30.838Z","caller":"clientv3/retry_interceptor.go:116"
,"msg":"retry stream intercept"}
{"level":"info","ts":1598530470.8388612,"caller":"snapshot/v3_snapshot.go:121","msg":"fetchi
ng snapshot","endpoint":"https://127.0.0.1:2379"}
{"level":"info","ts":1598530470.8570414,"caller":"snapshot/v3_snapshot.go:134","msg":"fetche
d snapshot","endpoint":"https://127.0.0.1:2379","took":0.025676157}
{"level":"info","ts":1598530470.8571067,"caller":"snapshot/v3_snapshot.go:143","msg":"saved"
,"path":"/srv/data/etcd-snapshot.db"}
Snapshot saved at /srv/data/etcd-snapshot.db
root@node-1:~#
```

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

NEW QUESTION 23
CORRECT TEXT
Score:7%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context k8s
```

Context
An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:

```
#
kubectl get pod big-corp-app -o yaml
#
apiVersion: v1
kind: Pod
metadata:
name: big-corp-app
spec:
containers:
- name: big-corp-app
image: busybox
args:
- /bin/sh
- -c
- > i=0;
while true;
do
echo "$(date) INFO $i" >> /var/log/big-corp-app.log;
i=$((i+1));
sleep 1;
done
volumeMounts:
- name: logs
mountPath: /var/log
- name: count-log-1
image: busybox
args: [/bin/sh, -c, 'tail -n+1 -f /var/log/big-corp-app.log']
volumeMounts:
- name: logs
mountPath: /var/log
volumes:
- name: logs
emptyDir: {
}
#
kubectl logs big-corp-app -c count-log-1
```

NEW QUESTION 28
CORRECT TEXT
Create an nginx pod and list the pod with different levels of verbosity

A. Mastered
B. Not Mastered

Answer: A

Explanation:

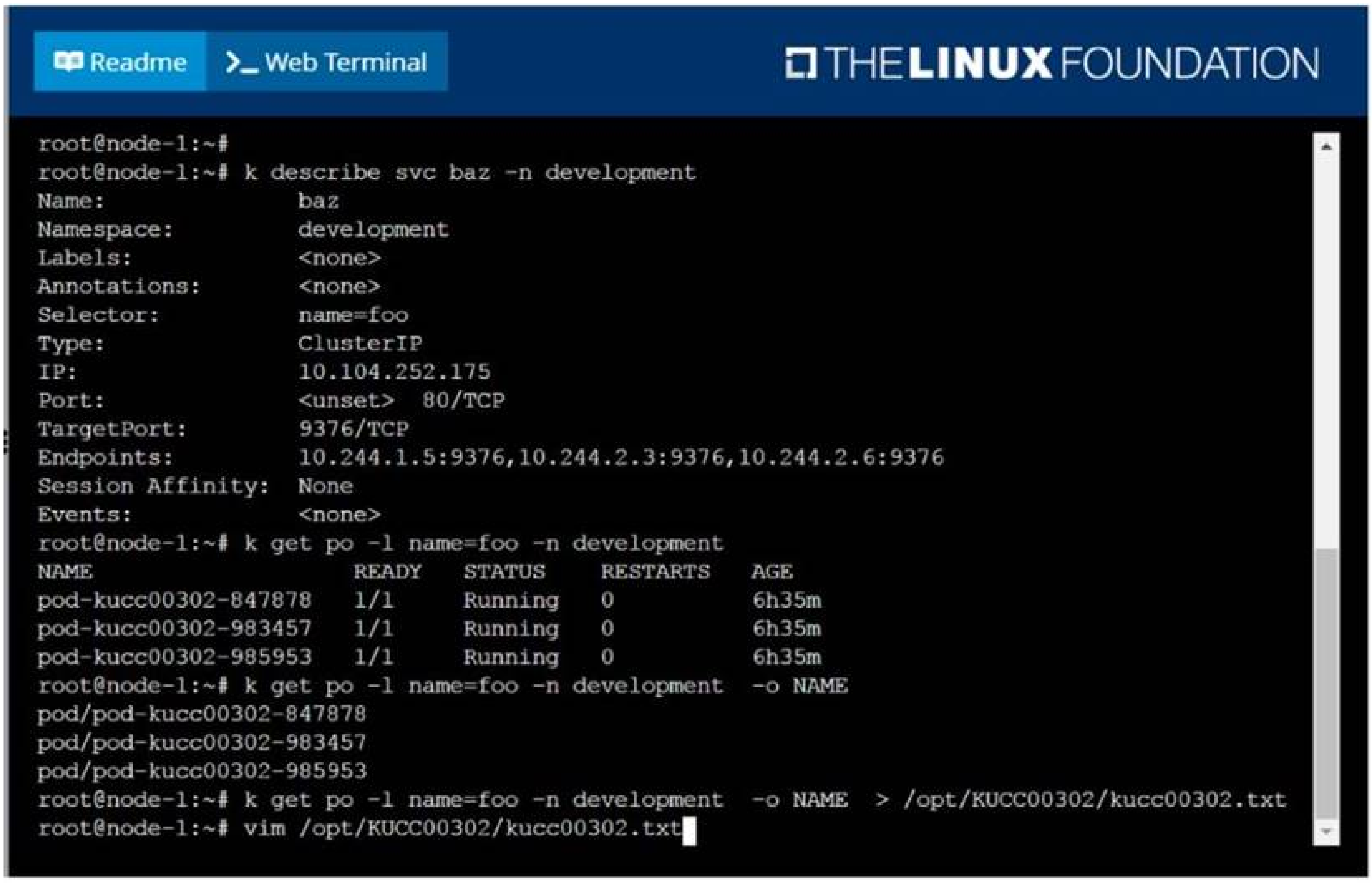
```
// create a pod
kubectl run nginx --image=nginx --restart=Never --port=80
// List the pod with different verbosity
kubectl get po nginx --v=7
kubectl get po nginx --v=8
kubectl get po nginx --v=9
```

NEW QUESTION 32
CORRECT TEXT
Create a file:
/opt/KUCC00302/kucc00302.txt that lists all pods that implement service baz in namespace development.
The format of the file should be one pod name per line.

A. Mastered
B. Not Mastered

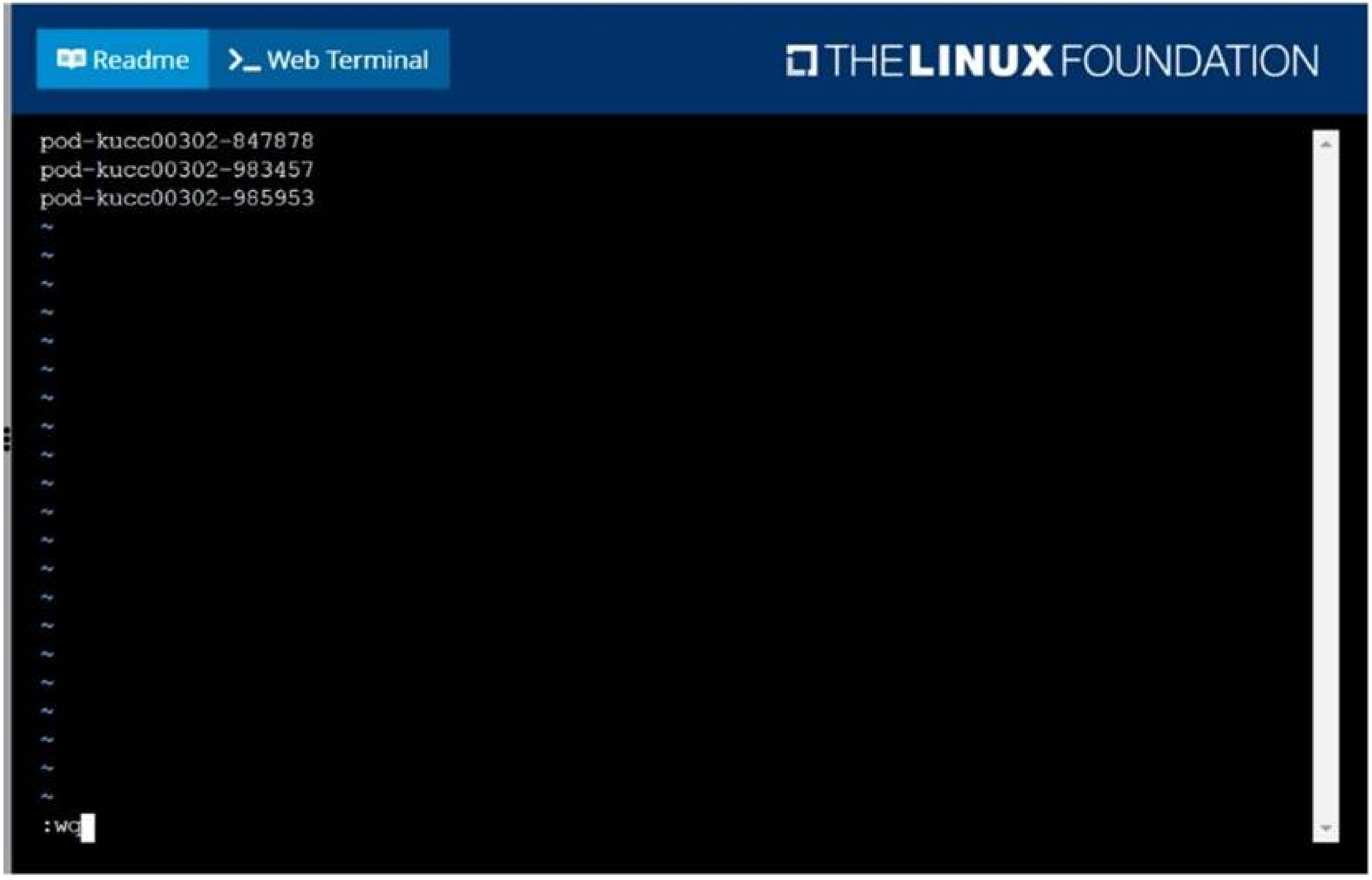
Answer: A

Explanation:
solution



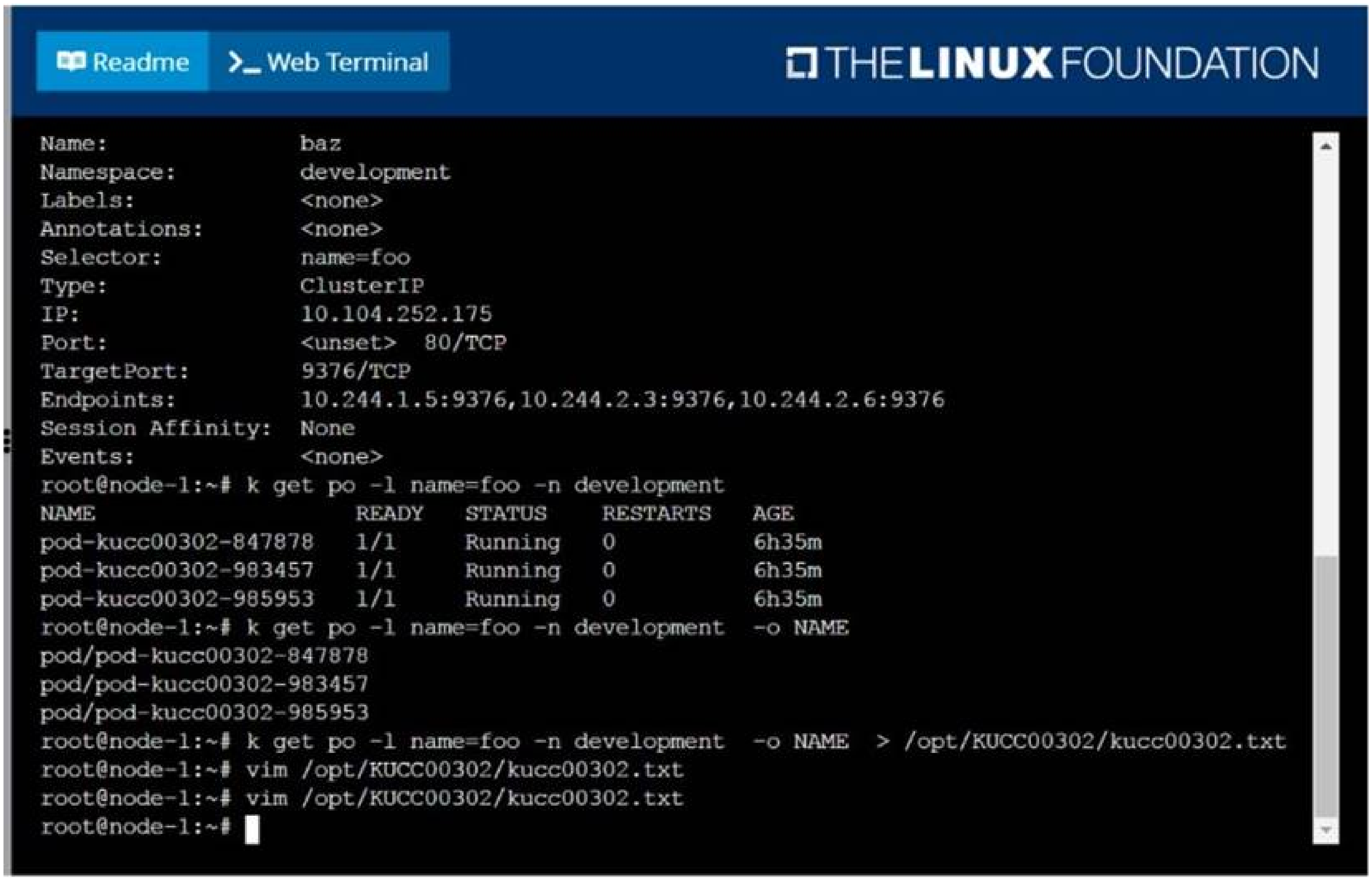
```
root@node-1:~#
root@node-1:~# k describe svc baz -n development
Name:              baz
Namespace:         development
Labels:            <none>
Annotations:       <none>
Selector:          name=foo
Type:              ClusterIP
IP:                10.104.252.175
Port:              <unset>  80/TCP
TargetPort:        9376/TCP
Endpoints:         10.244.1.5:9376,10.244.2.3:9376,10.244.2.6:9376
Session Affinity:  None
Events:            <none>
root@node-1:~# k get po -l name=foo -n development
NAME                   READY   STATUS    RESTARTS   AGE
pod-kucc00302-847878   1/1     Running   0          6h35m
pod-kucc00302-983457   1/1     Running   0          6h35m
pod-kucc00302-985953   1/1     Running   0          6h35m
root@node-1:~# k get po -l name=foo -n development  -o NAME
pod/pod-kucc00302-847878
pod/pod-kucc00302-983457
pod/pod-kucc00302-985953
root@node-1:~# k get po -l name=foo -n development  -o NAME  > /opt/KUCC00302/kucc00302.txt
root@node-1:~# vim /opt/KUCC00302/kucc00302.txt
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 B.JPG



```
pod-kucc00302-847878
pod-kucc00302-983457
pod-kucc00302-985953
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 C.JPG



```
Name:              baz
Namespace:         development
Labels:            <none>
Annotations:       <none>
Selector:          name=foo
Type:              ClusterIP
IP:                10.104.252.175
Port:              <unset>  80/TCP
TargetPort:        9376/TCP
Endpoints:         10.244.1.5:9376,10.244.2.3:9376,10.244.2.6:9376
Session Affinity:  None
Events:            <none>
root@node-1:~# k get po -l name=foo -n development
NAME                    READY   STATUS    RESTARTS   AGE
pod-kucc00302-847878    1/1     Running   0          6h35m
pod-kucc00302-983457    1/1     Running   0          6h35m
pod-kucc00302-985953    1/1     Running   0          6h35m
root@node-1:~# k get po -l name=foo -n development  -o NAME
pod/pod-kucc00302-847878
pod/pod-kucc00302-983457
pod/pod-kucc00302-985953
root@node-1:~# k get po -l name=foo -n development  -o NAME  > /opt/KUCC00302/kucc00302.txt
root@node-1:~# vim /opt/KUCC00302/kucc00302.txt
root@node-1:~# vim /opt/KUCC00302/kucc00302.txt
root@node-1:~# 
```

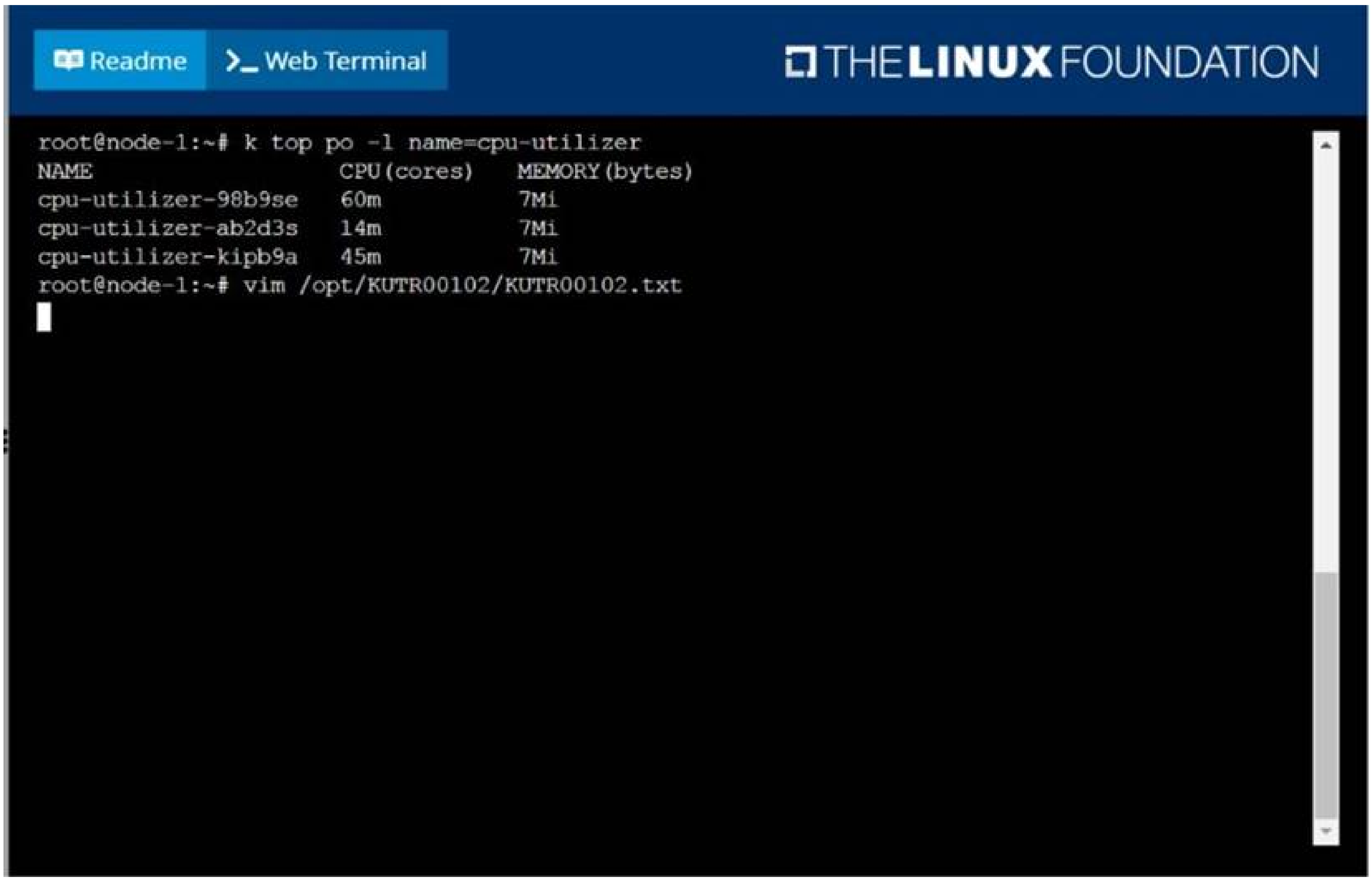
F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 D.JPG

NEW QUESTION 33
CORRECT TEXT
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/KUTR00102/KUTR00102.txt
(which already exists).

A.

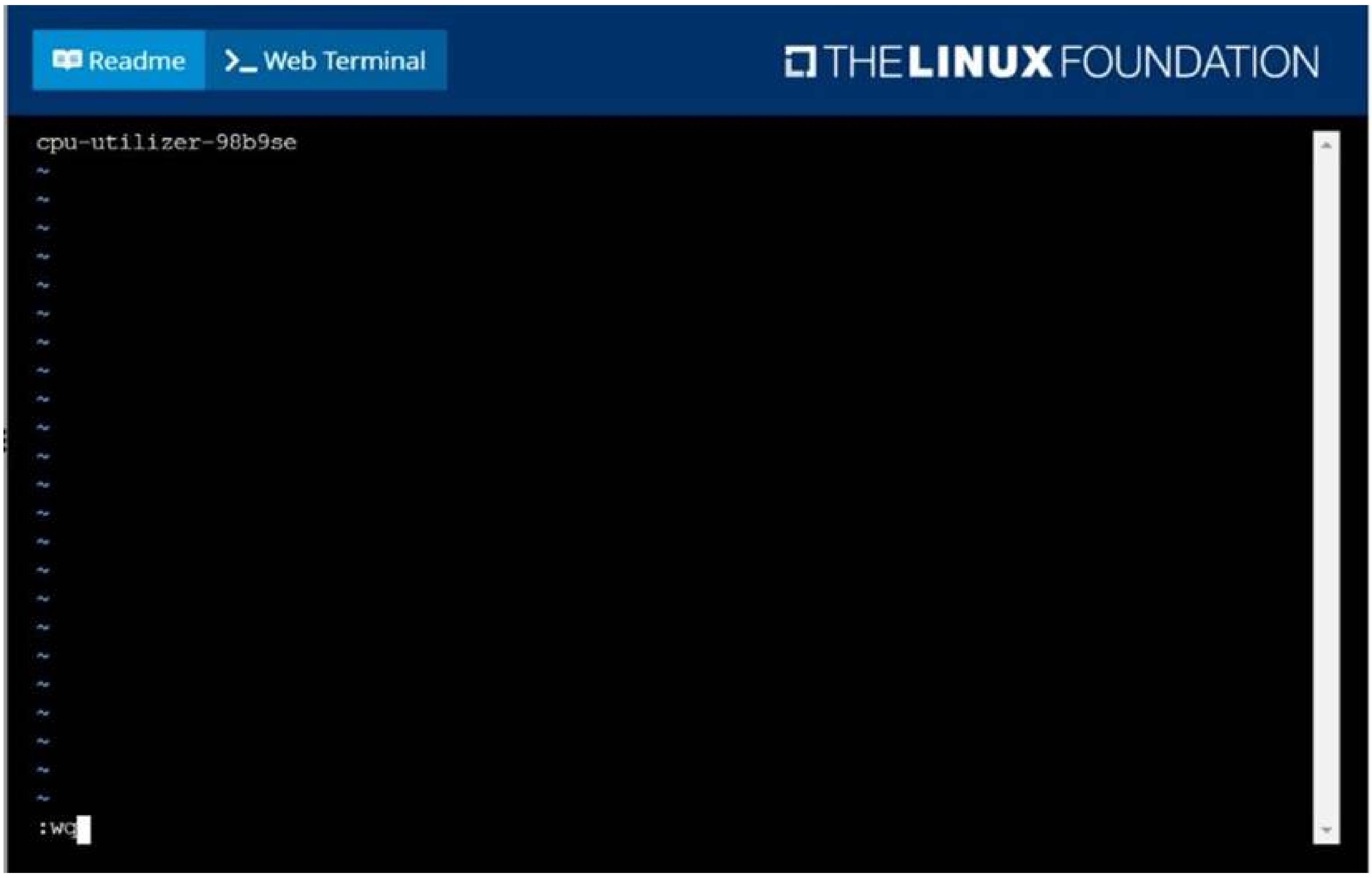
Answer: Seethesolutionbelow.

Explanation:
solution



```
root@node-1:~# k top po -l name=cpu-utilizer
NAME                  CPU(cores)   MEMORY(bytes)
cpu-utilizer-98b9se   60m          7Mi
cpu-utilizer-ab2d3s   14m          7Mi
cpu-utilizer-kipb9a   45m          7Mi
root@node-1:~# vim /opt/KUTR00102/KUTR00102.txt
```

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



```
cpu-utilizer-98b9se
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

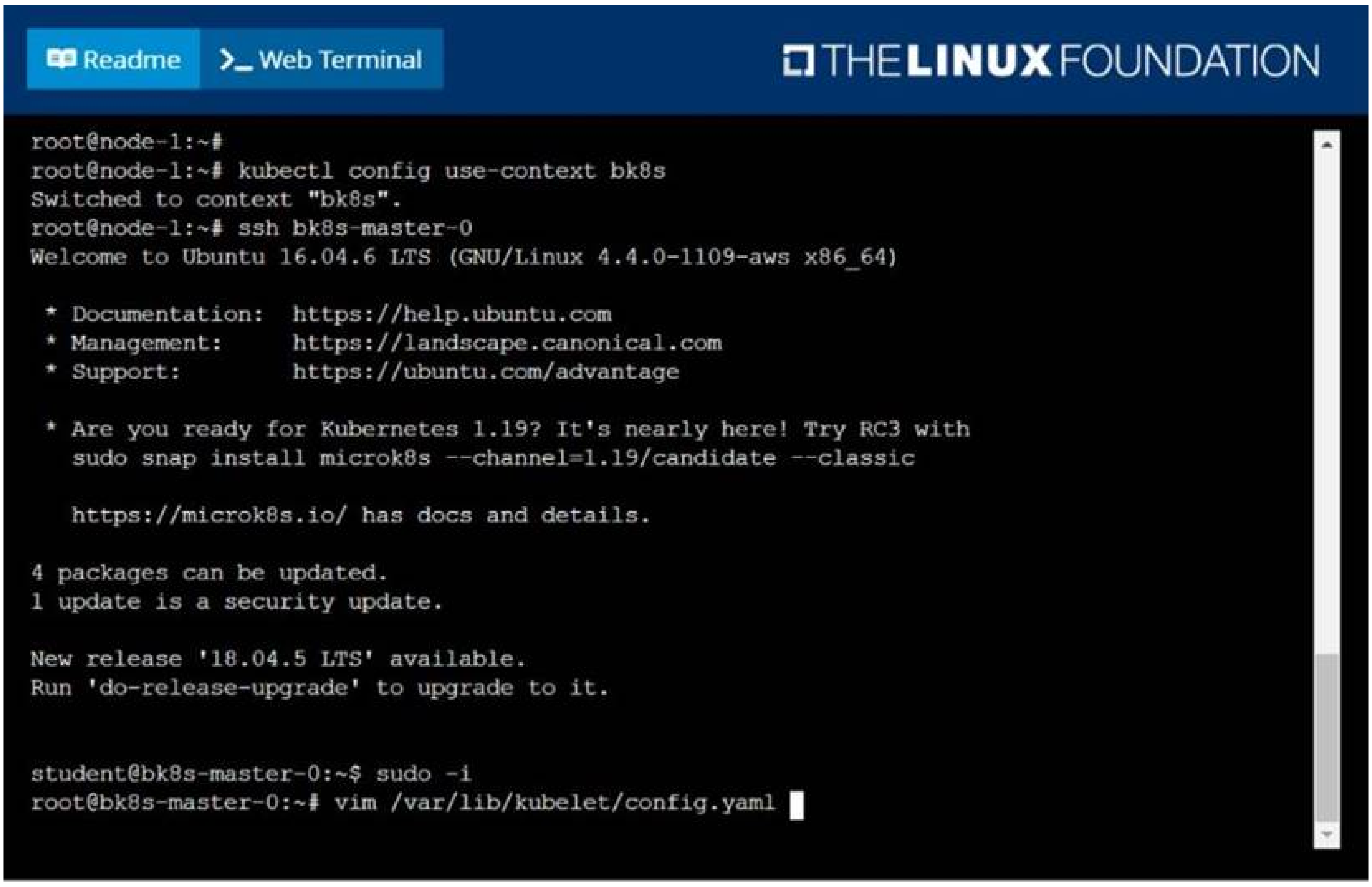
F:\Work\Data Entry Work\Data Entry\20200827\CKA\16 C.JPG

NEW QUESTION 34
CORRECT TEXT
Given a partially-functioning Kubernetes cluster, identify symptoms of failure on the cluster.
Determine the node, the failing service, and take actions to bring up the failed service and restore the health of the cluster. Ensure that any changes are made permanently.
You can ssh to the relevant I nodes (bk8s-master-0 or bk8s-node-0) using:
[student@node-1] $ ssh <nodename>
You can assume elevated privileges on any node in the cluster with the following command:
[student@nodename] $ | sudo –i

A. Mastered
B. Not Mastered

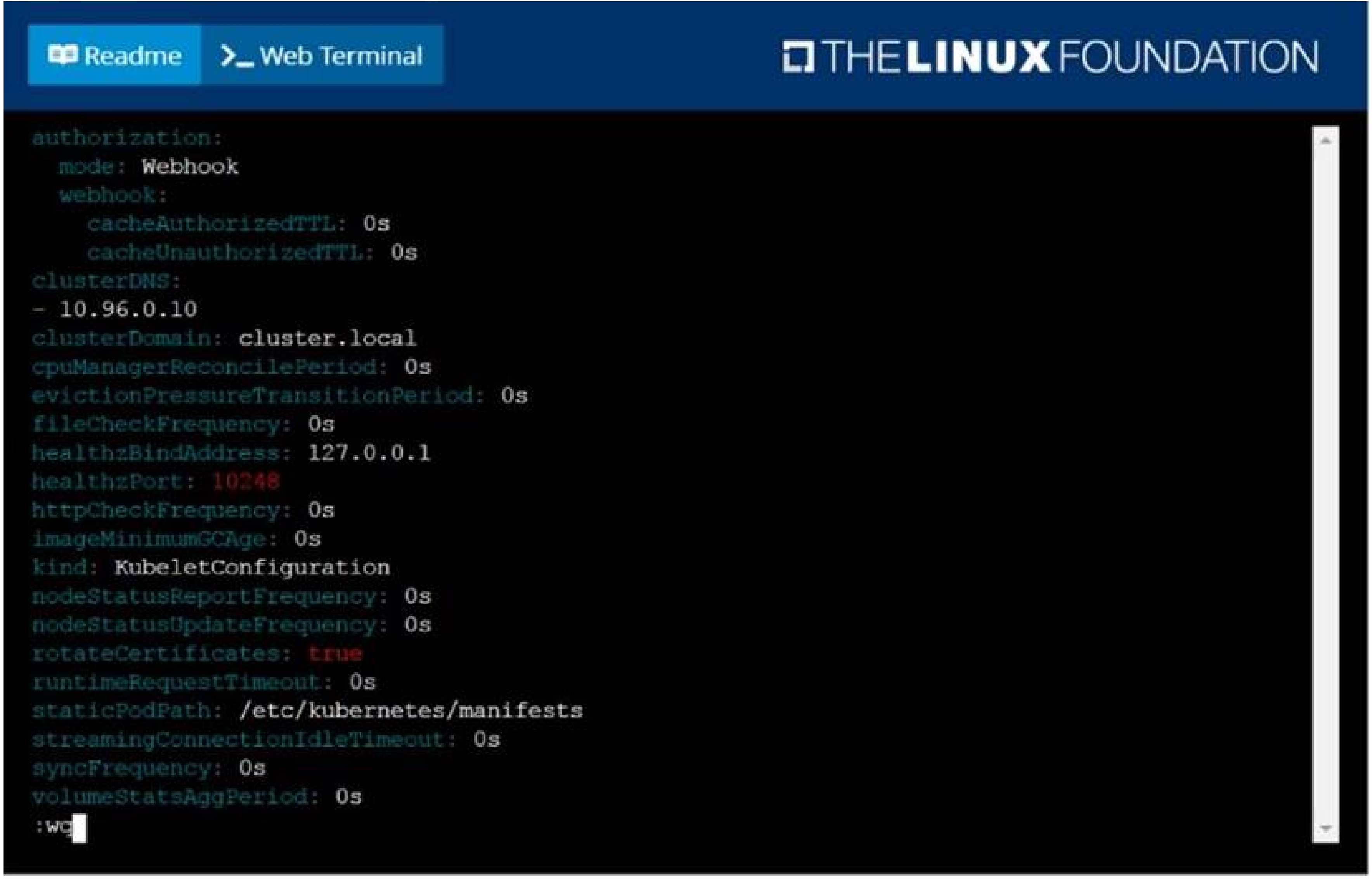
Answer: A

Explanation:
solution



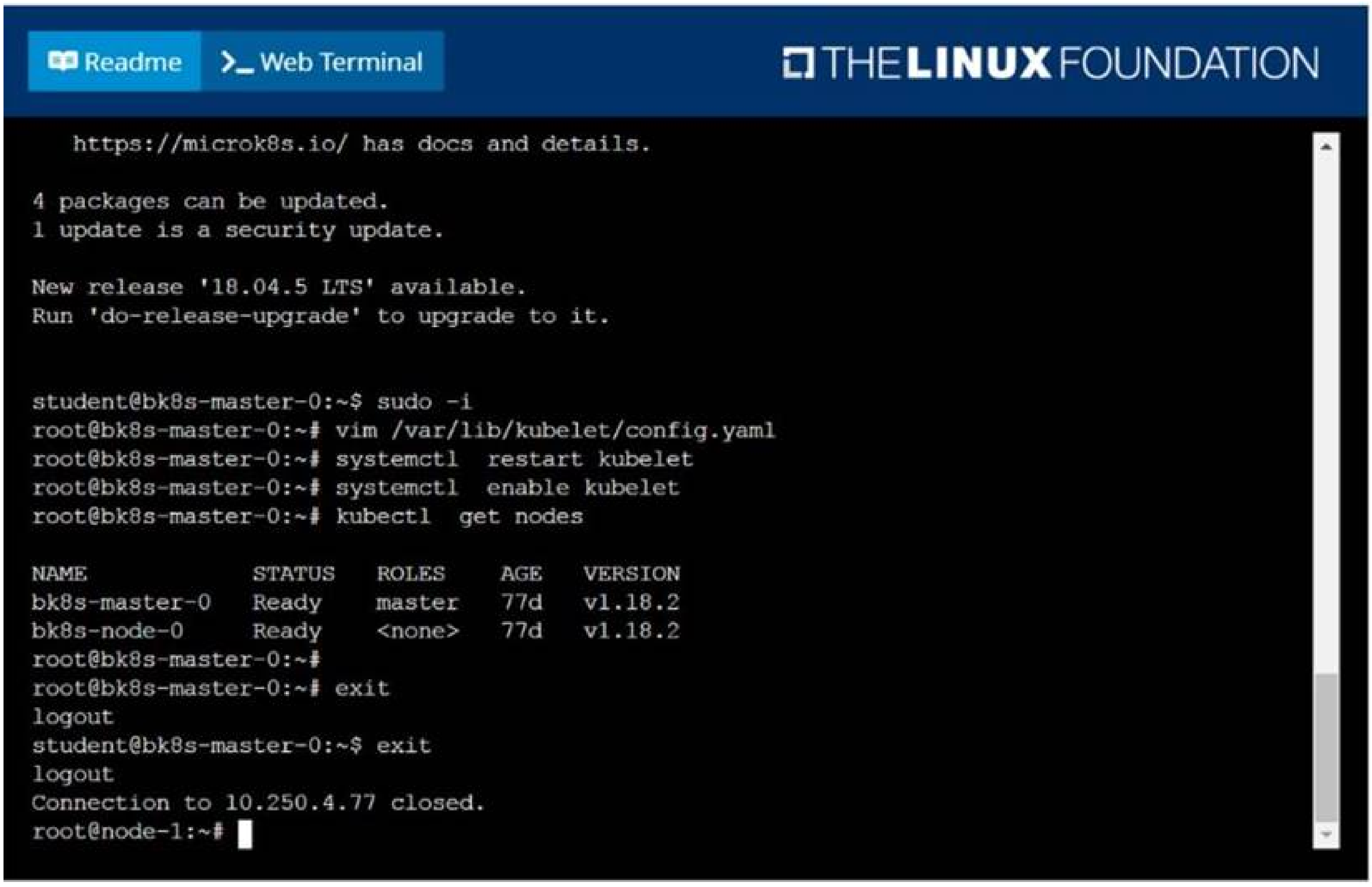
```
root@node-1:~#
root@node-1:~# kubectl config use-context bk8s
Switched to context "bk8s".
root@node-1:~# ssh bk8s-master-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

 * 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@bk8s-master-0:~$ sudo -i
root@bk8s-master-0:~# vim /var/lib/kubelet/config.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\23 C.JPG



```
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
volumeStatsAggPeriod: 0s
:wq
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\23 D.JPG



```
   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@bk8s-master-0:~$ sudo -i
root@bk8s-master-0:~# vim /var/lib/kubelet/config.yaml
root@bk8s-master-0:~# systemctl  restart kubelet
root@bk8s-master-0:~# systemctl  enable kubelet
root@bk8s-master-0:~# kubectl  get nodes

NAME             STATUS   ROLES    AGE   VERSION
bk8s-master-0    Ready    master   77d   v1.18.2
bk8s-node-0      Ready    <none>   77d   v1.18.2
root@bk8s-master-0:~#
root@bk8s-master-0:~# exit
logout
student@bk8s-master-0:~$ exit
logout
Connection to 10.250.4.77 closed.
root@node-1:~# 
```

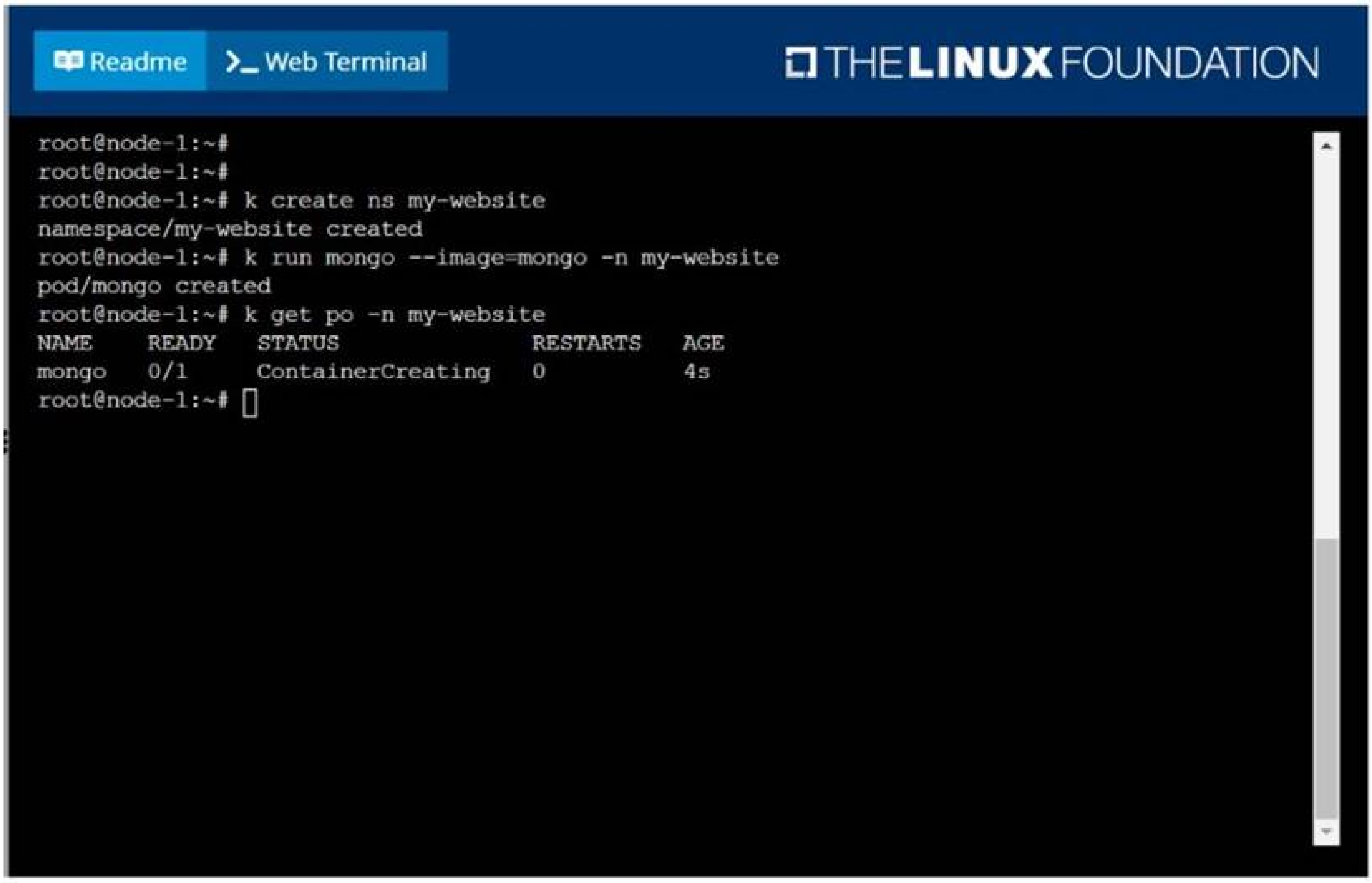
F:\Work\Data Entry Work\Data Entry\20200827\CKA\23 E.JPG

NEW QUESTION 37
CORRECT TEXT
Create a pod as follows:
? Name: mongo
? Using Image: mongo
? In a new Kubernetes namespace named: my-website

A. Mastered
B. Not Mastered

Answer: A

Explanation:
solution



```
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:~# 
```

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

NEW QUESTION 40
CORRECT TEXT
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

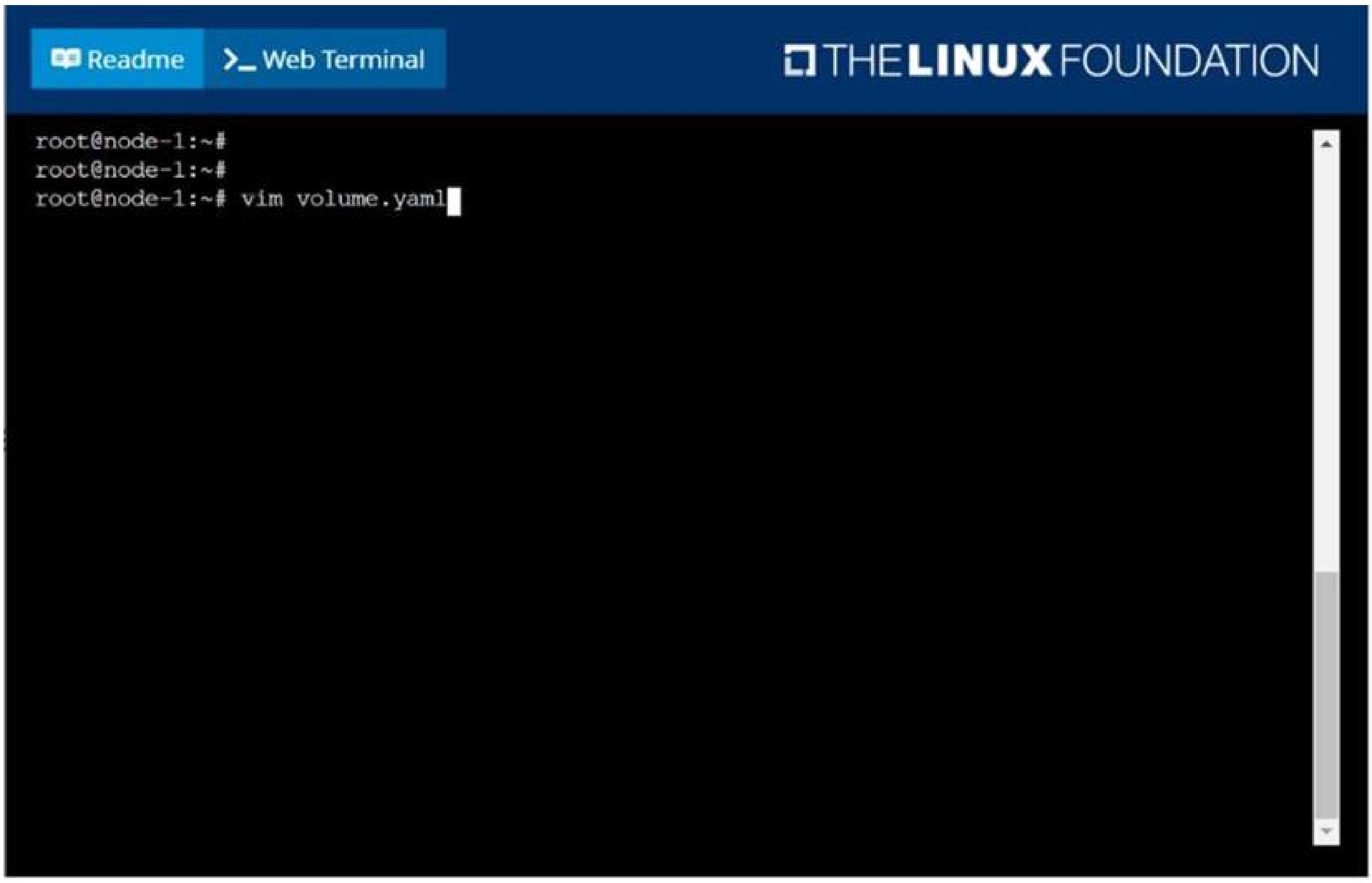
Explanation:
kubectl run busybox --image=busybox -it --rm --restart=Never --
/bin/sh -c 'echo hello world'
kubectl get po # You shouldn't see pod with the name "busybox"

NEW QUESTION 45
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

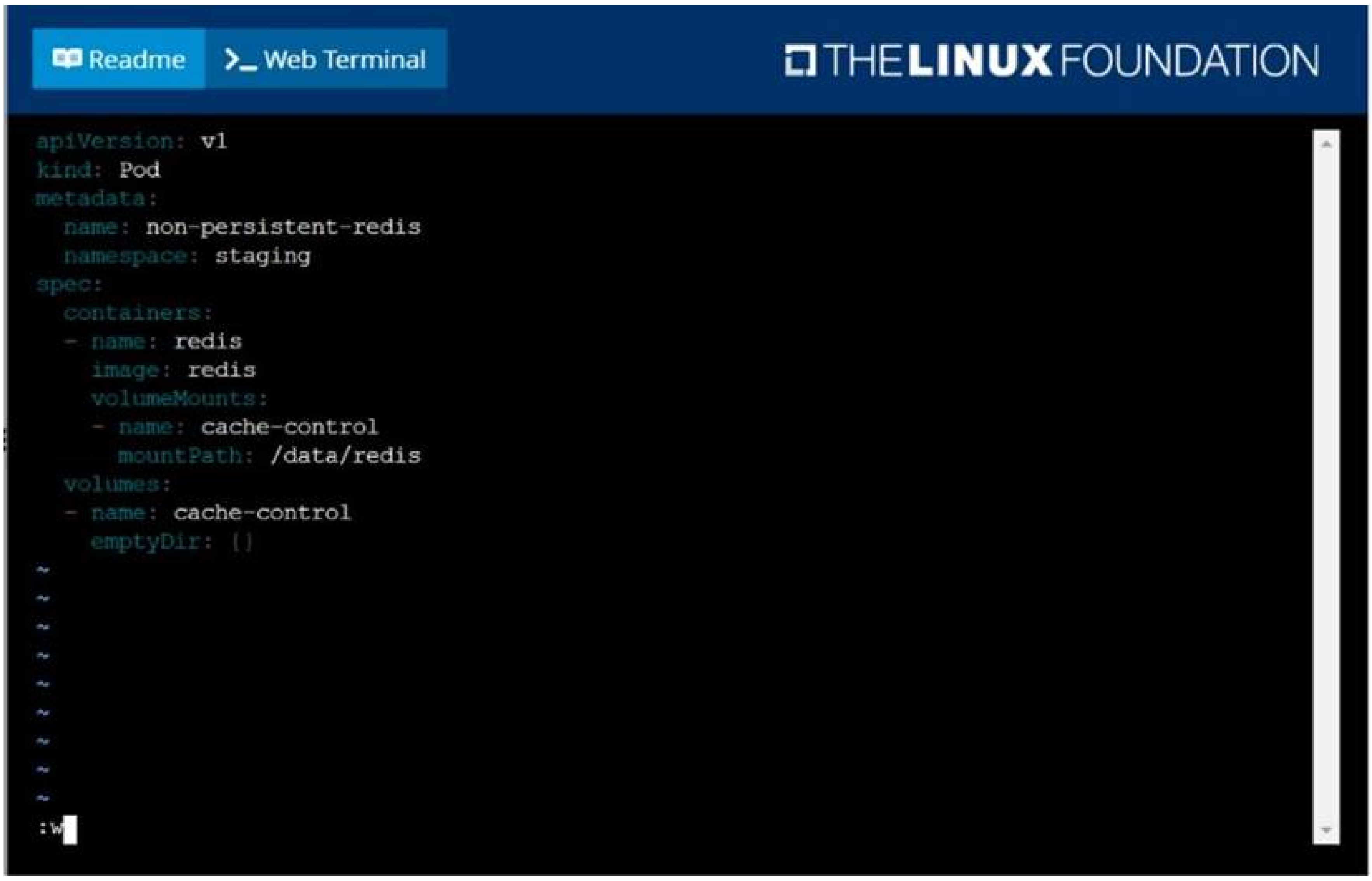
Answer: A

Explanation:
solution



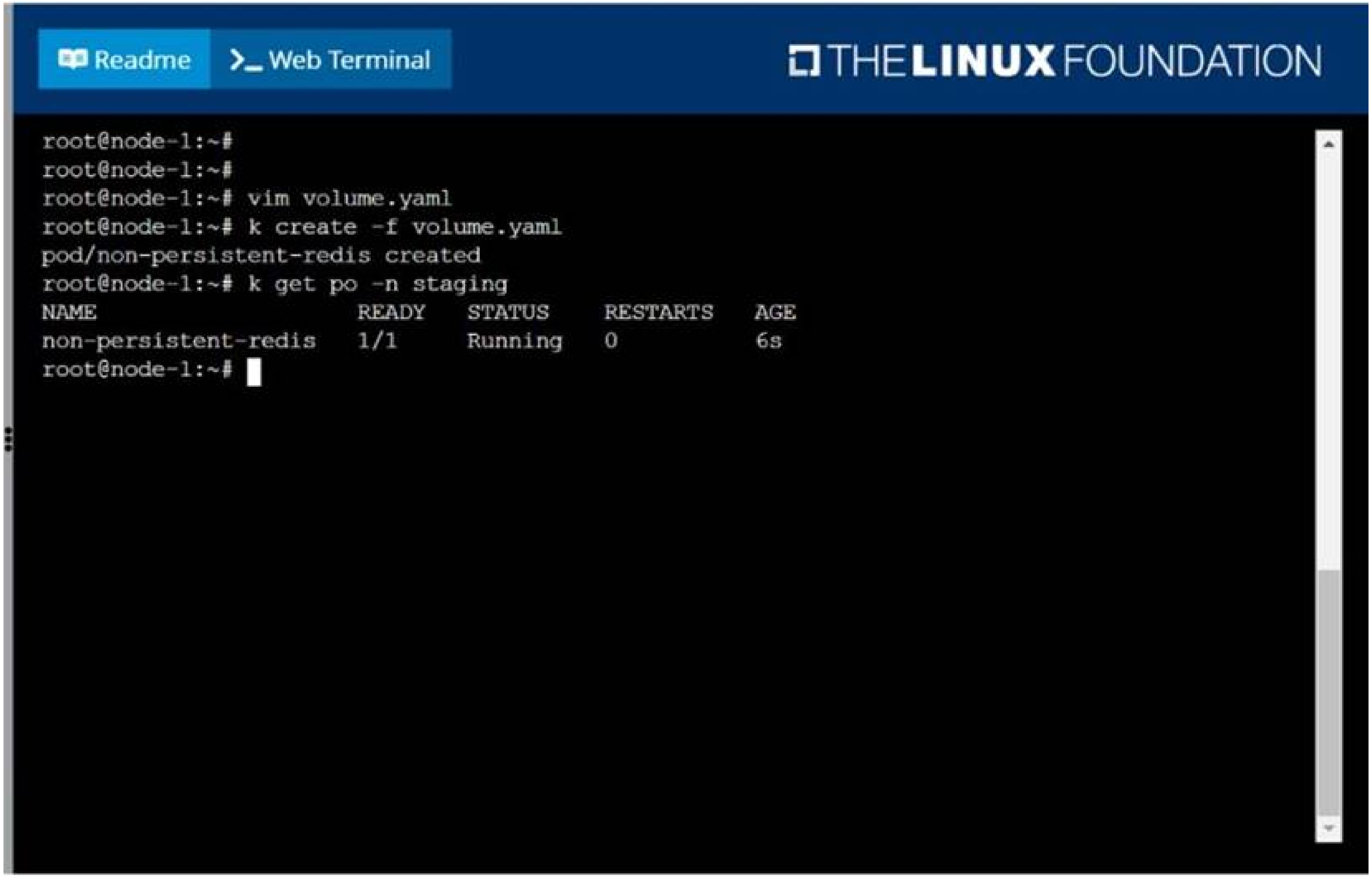
```
root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml
```

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



```
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: {}
~
~
~
~
~
~
~
~
~
:w
```

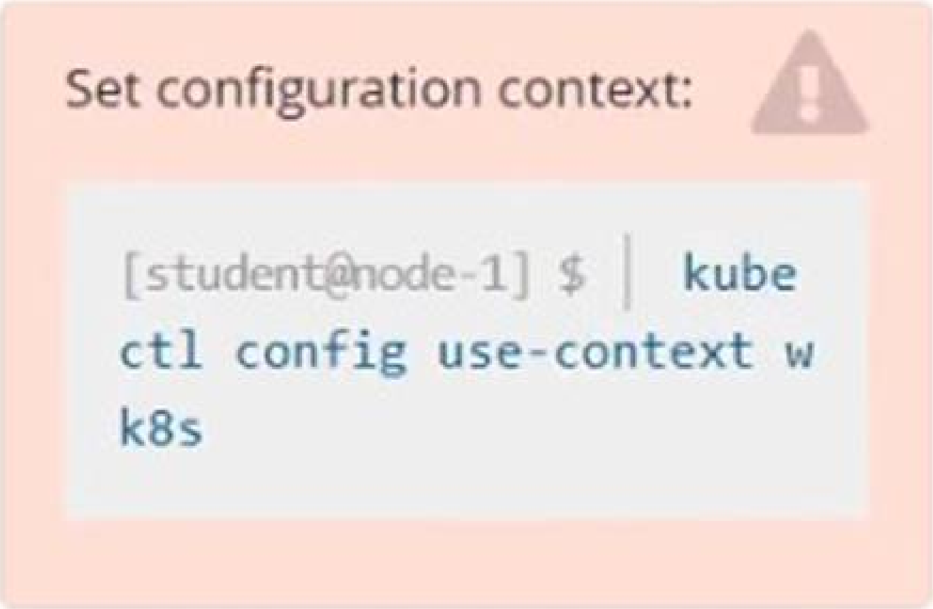
F:\Work\Data Entry Work\Data Entry\20200827\CKA\13 C.JPG



```
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:~#
```

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

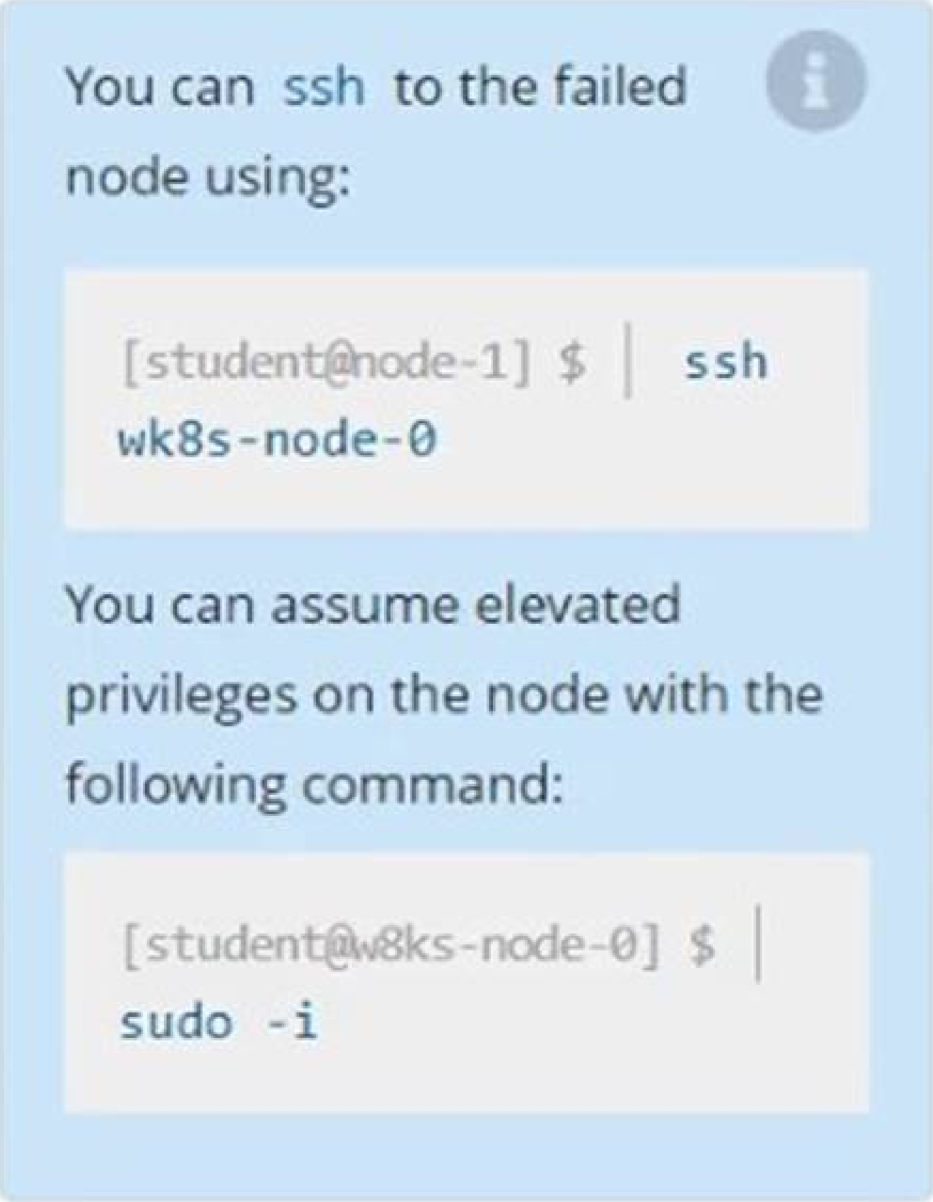
NEW QUESTION 48
CORRECT TEXT
Score: 13%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context wk8s
```

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@w8ks-node-0] $ | sudo -i
```

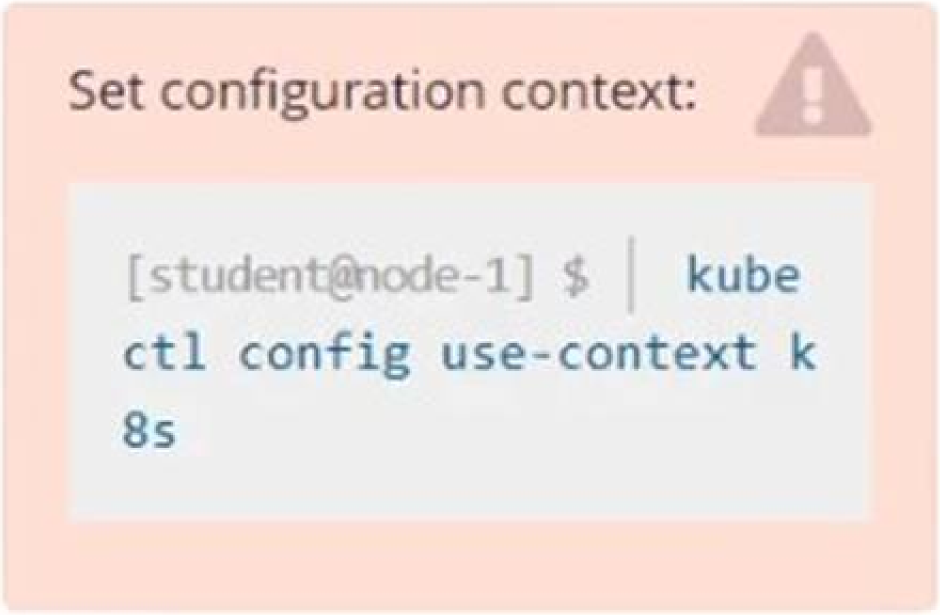
A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
sudo -i
systemctl status kubelet
systemctl start kubelet
systemctl enable kubelet

NEW QUESTION 53
CORRECT TEXT
Score: 4%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context k8s
```

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

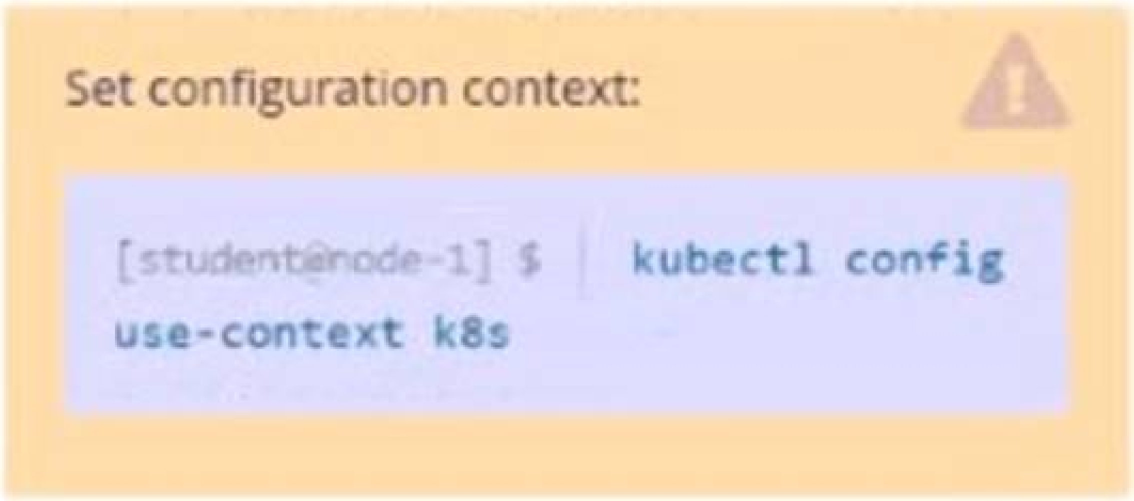
Answer: A

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 57
CORRECT TEXT
Task Weight: 4%



Set configuration context:

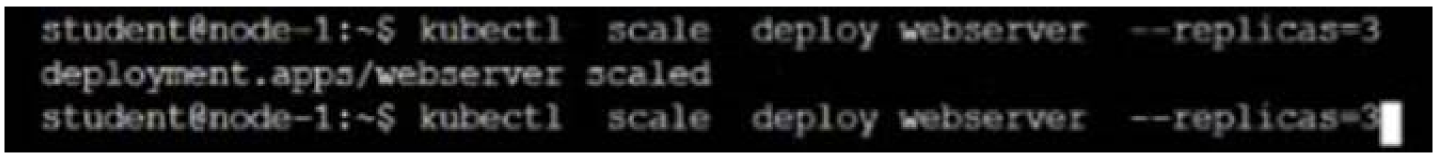
```
[student@node-1] $ | kubectl config use-context k8s
```

Task
Scale the deployment webserver to 3 pods.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:



```
student@node-1:~$ kubectl  scale  deploy webserver  --replicas=3
deployment.apps/webserver scaled
student@node-1:~$ kubectl  scale  deploy webserver  --replicas=3
```

NEW QUESTION 58

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

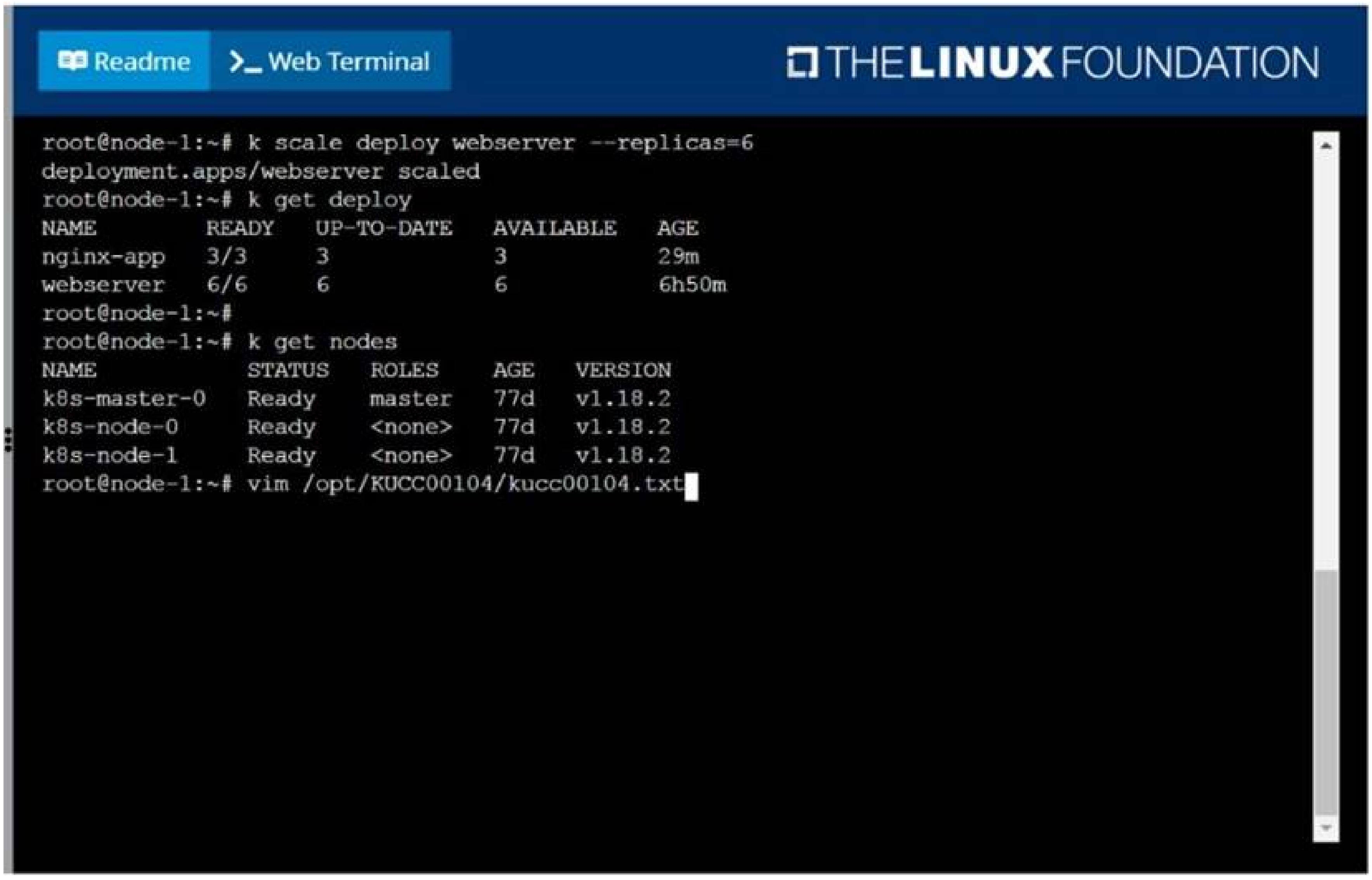
A. Mastered

B. Not Mastered

Answer: A

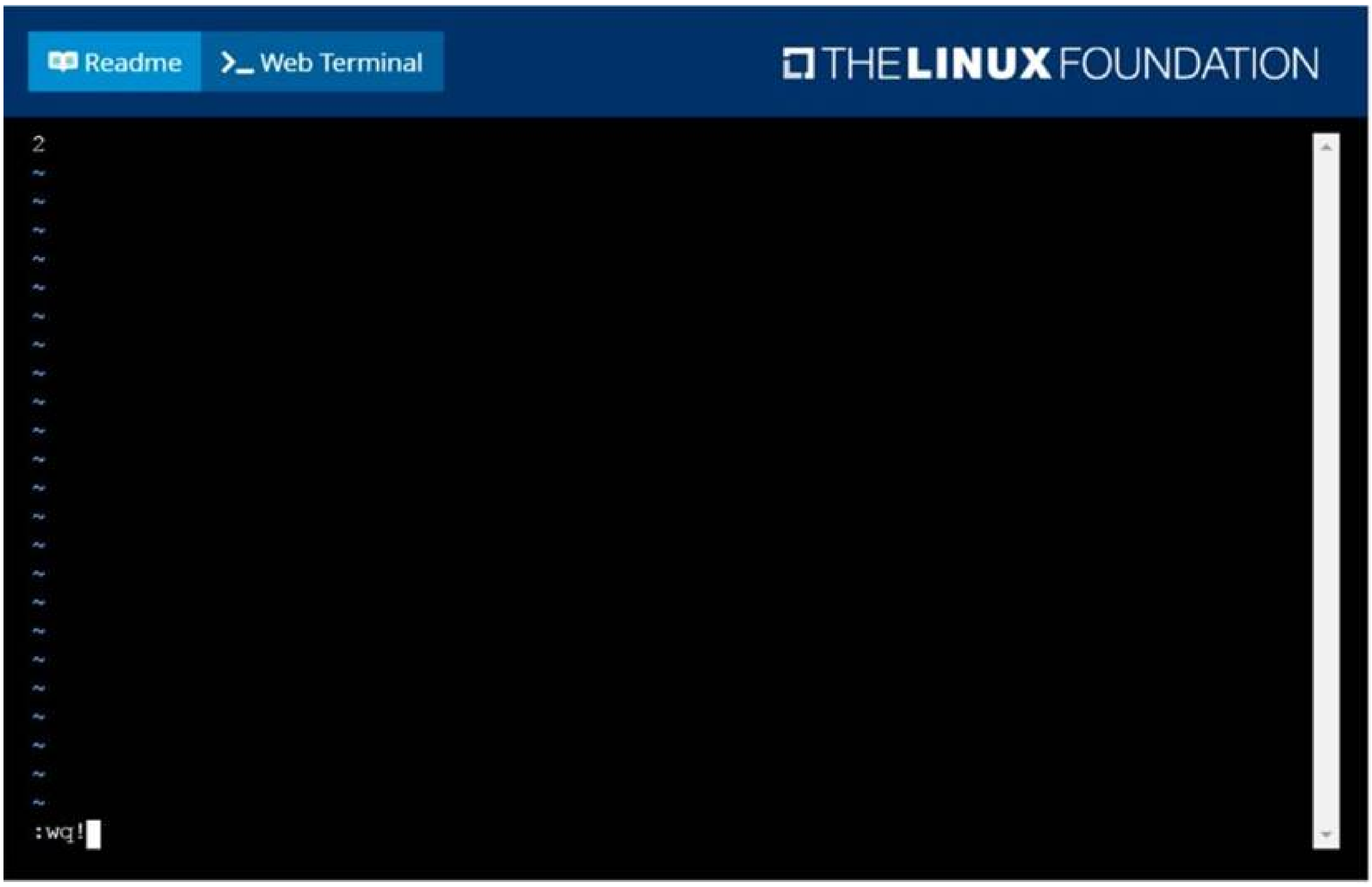
Explanation:

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



```
2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

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

NEW QUESTION 59

CORRECT TEXT

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@w8ks-node-0] $ | sudo –i

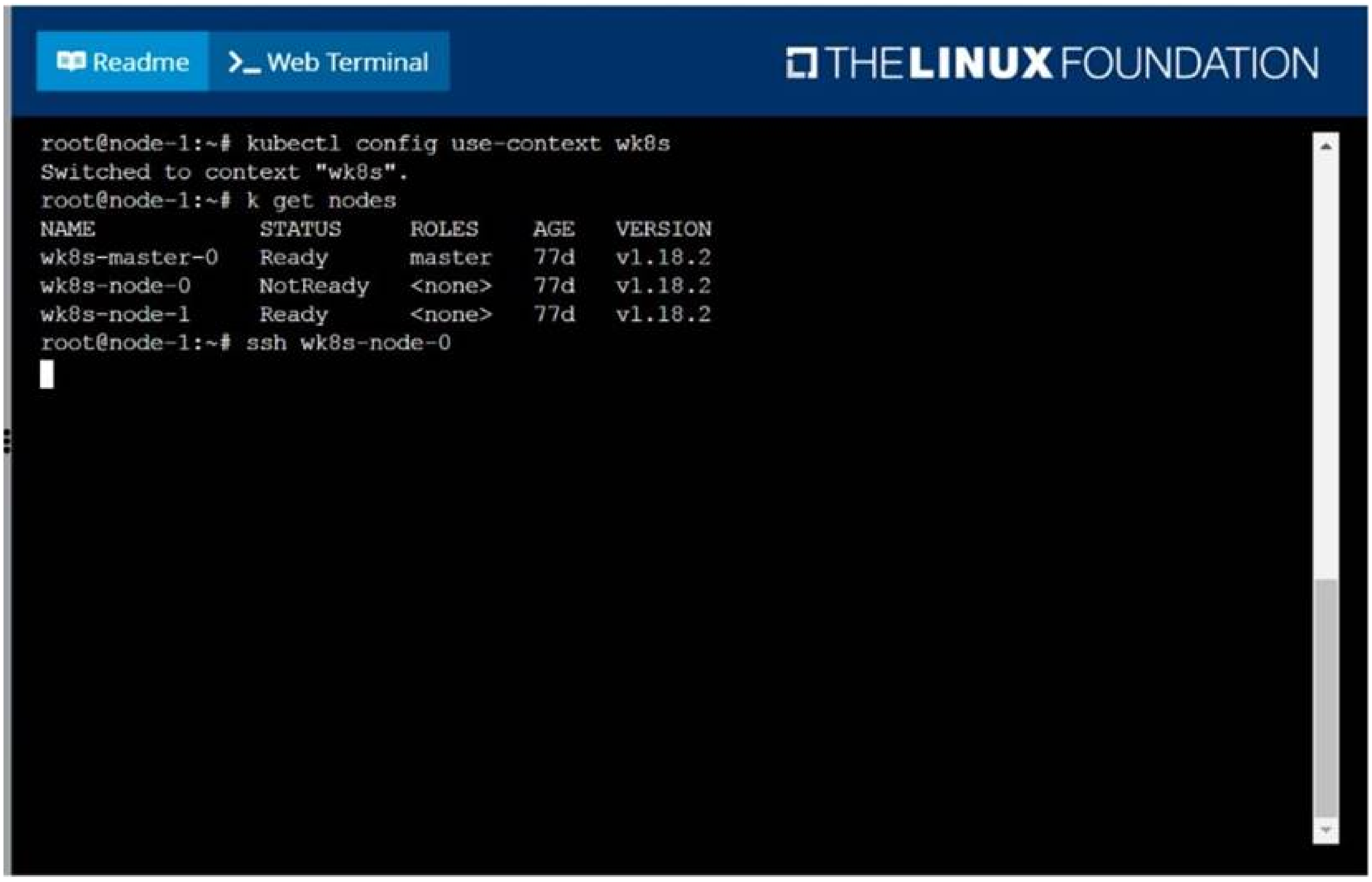
A. Mastered

B. Not Mastered

Answer: A

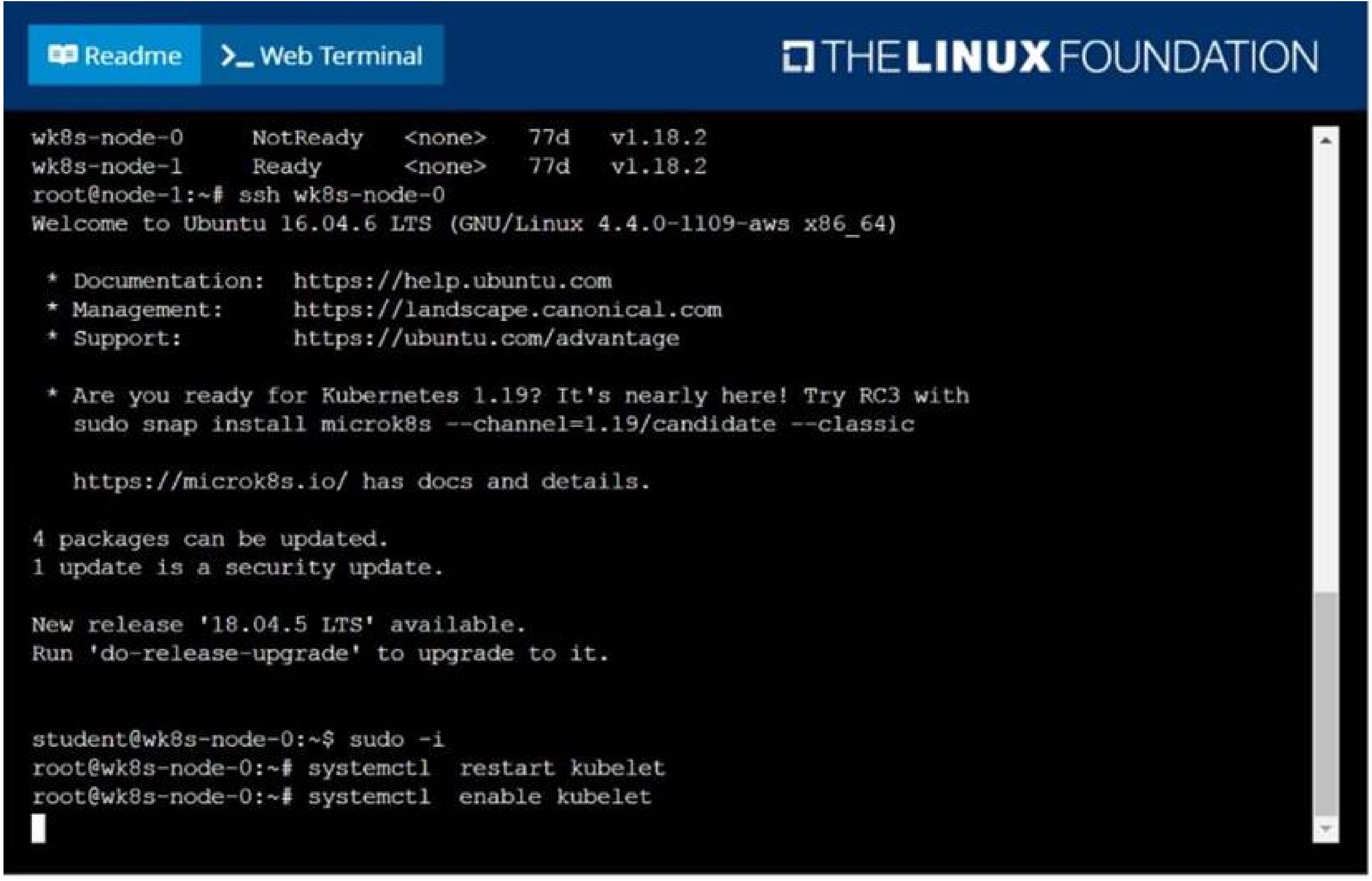
Explanation:

solution



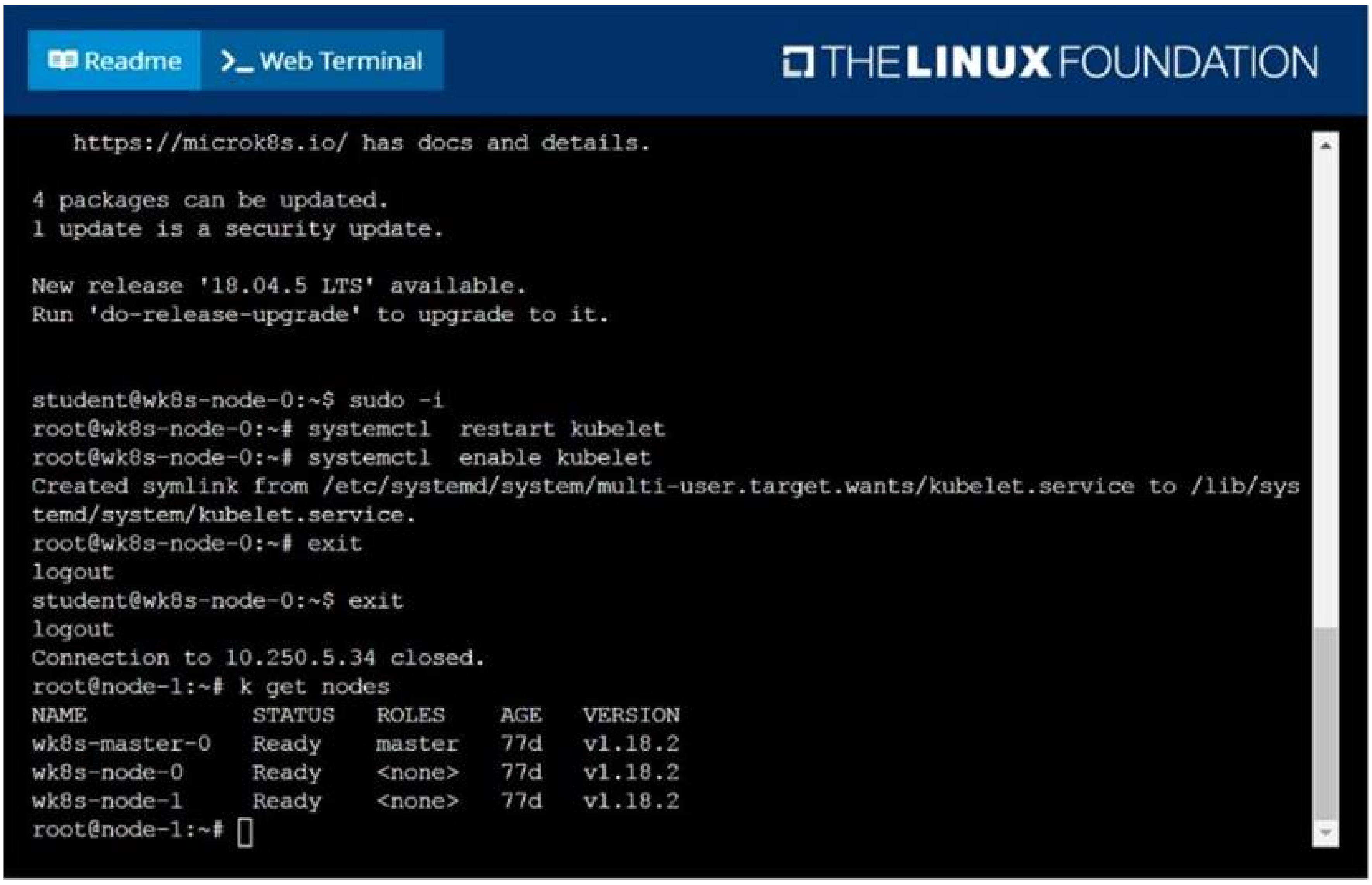
```
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# k get nodes
NAME            STATUS     ROLES    AGE   VERSION
wk8s-master-0   Ready      master   77d   v1.18.2
wk8s-node-0     NotReady   <none>   77d   v1.18.2
wk8s-node-1     Ready      <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 C.JPG



```
wk8s-node-0     NotReady   <none>   77d   v1.18.2
wk8s-node-1     Ready      <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
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-0:~$ sudo -i
root@wk8s-node-0:~# systemctl  restart kubelet
root@wk8s-node-0:~# systemctl  enable kubelet
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 D.JPG



```
   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-0:~$ sudo -i
root@wk8s-node-0:~# systemctl  restart kubelet
root@wk8s-node-0:~# systemctl  enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/sys
temd/system/kubelet.service.
root@wk8s-node-0:~# exit
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.5.34 closed.
root@node-1:~# k get nodes
NAME            STATUS    ROLES     AGE    VERSION
wk8s-master-0   Ready     master    77d    v1.18.2
wk8s-node-0     Ready     <none>    77d    v1.18.2
wk8s-node-1     Ready     <none>    77d    v1.18.2
root@node-1:~# 
```

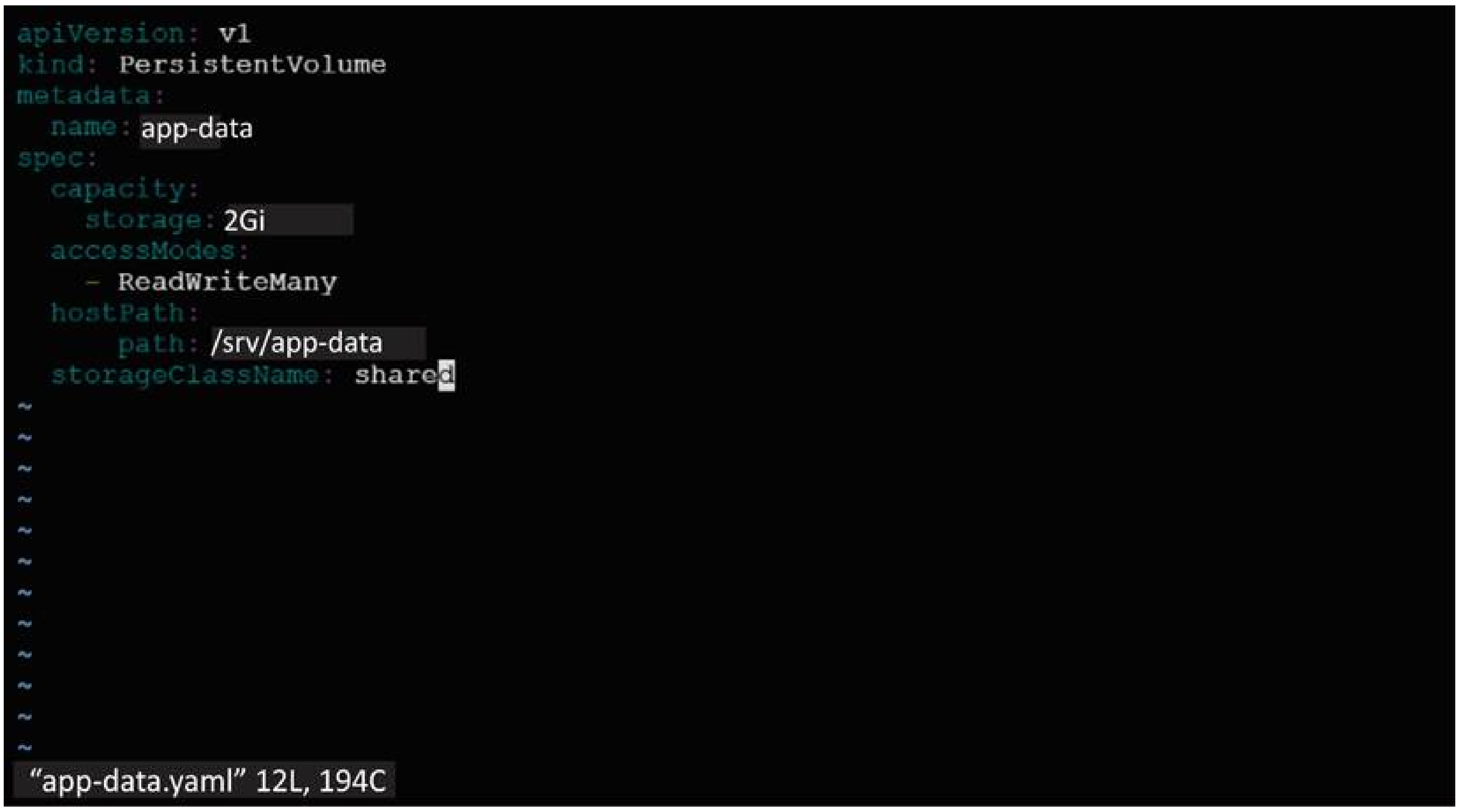
F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 E.JPG

NEW QUESTION 61
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

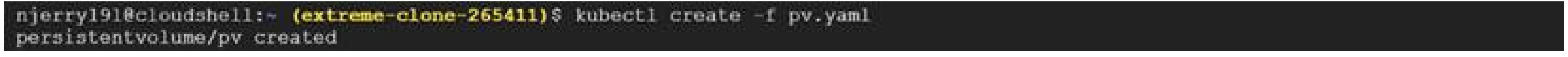
Answer: A

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 tying 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.



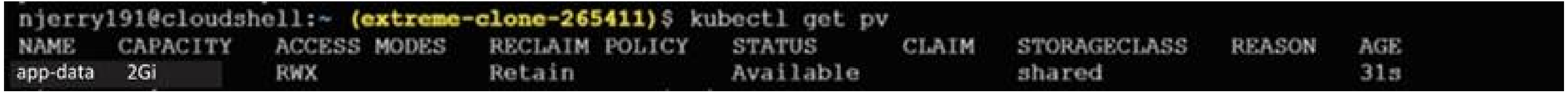
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
      path: /srv/app-data
  storageClassName: shared
~
~
~
~
~
~
~
~
~
~
~
~
 "app-data.yaml" 12L, 194C
```

* 2. Save the file and create the persistent volume.



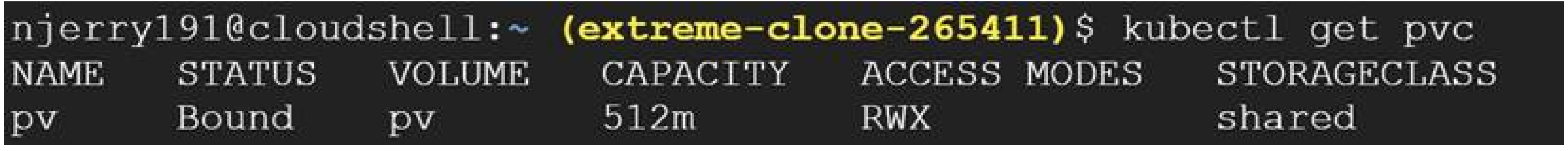
```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl create -f pv.yaml
persistentvolume/pv created
```

Image for post
* 3. View the persistent volume.



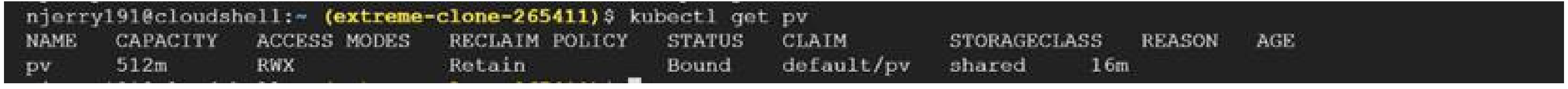
```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
NAME      CAPACITY    ACCESS MODES    RECLAIM POLICY    STATUS       CLAIM    STORAGECLASS    REASON    AGE
app-data  2Gi         RWX             Retain            Available             shared                    31s
```

? 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: - ReadWriteMany resources:
requests: storage: 2Gi
storageClassName: shared
* 2. Save and create the pvc
njerry191@cloudshell:~ (extreme-clone-2654111)$ kubect1 create -f app-data.yaml persistentvolumeclaim/app-data created
* 3. View the pvc



```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pvc
NAME    STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS
pv      Bound     pv        512m        RWX             shared
```

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



```
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
```

Image for post
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: null name: app-dataspec: volumes: - name:congigpvc persistenVolumeClaim: claimName: app-data containers: - image: nginx name: app volumeMounts: - mountPath: "/srv/app-data " name: configpvc

NEW QUESTION 64
CORRECT TEXT
List all the pods sorted by created timestamp

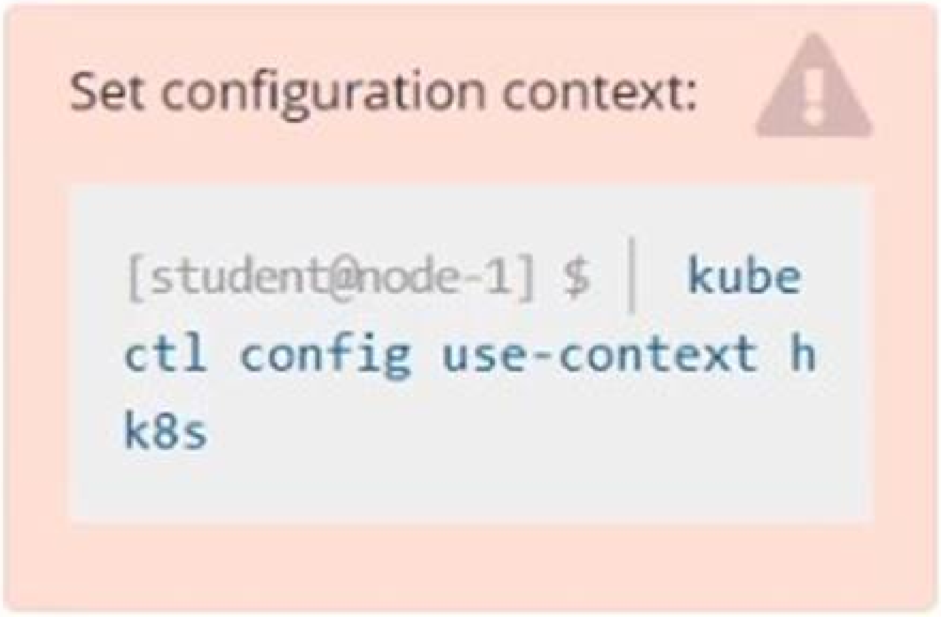
A. Mastered
B. Not Mastered

Answer: A

Explanation:
kubect1 get pods--sort-by=.metadata.creationTimestamp

NEW QUESTION 69

CORRECT TEXT
Score: 4%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context hk8s
```

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 .

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:

```
#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
```

NEW QUESTION 70
CORRECT TEXT
List “nginx-dev” and “nginx-prod” pod and delete those pods

A. Mastered
B. Not Mastered

Answer: A

Explanation:

```
kubect1 get pods -o wide
kubectl delete po “nginx-dev”kubectl delete po “nginx-prod”
```

NEW QUESTION 71
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-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
```

NEW QUESTION 76

......

Relate Links

Contact us