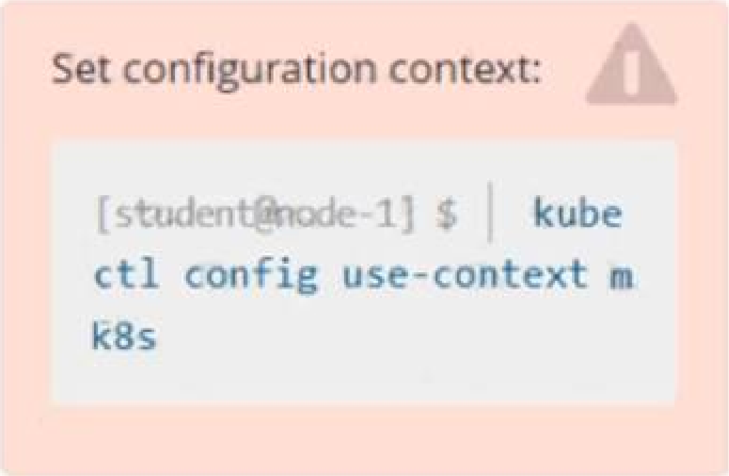
Linux-Foundation

Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

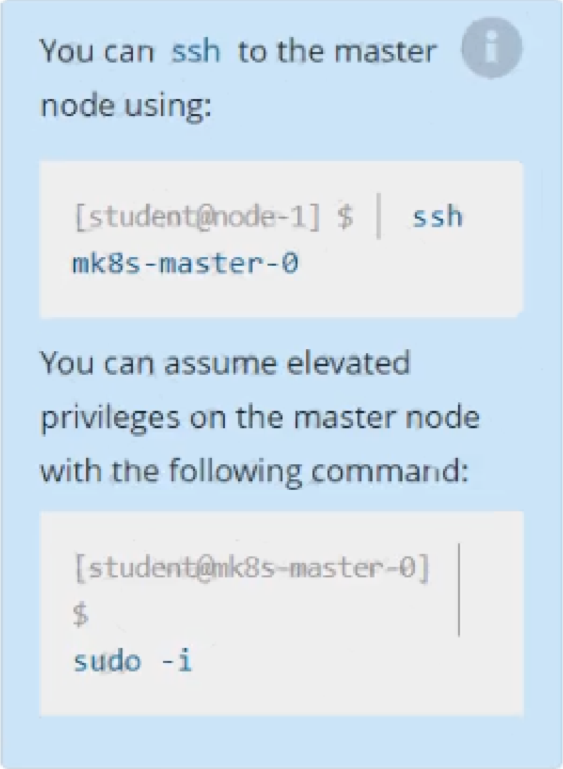
NEW QUESTION 1
Score: 7%



Set configuration context:

```
[student@node-1] $ | kubectl config use-context mk8s
```

Task
Given an existing Kubernetes cluster running version 1.20.0, upgrade all of the Kubernetes control plane and node components on the master node only to version 1.20.1.
Be sure to drain the master node before upgrading it and uncordon it after the upgrade.



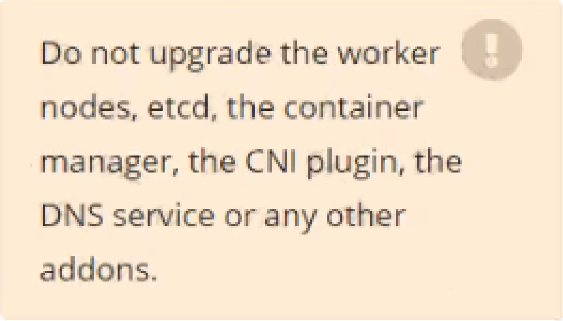
You can ssh to the master node using:

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

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

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

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



Do not upgrade the worker nodes, etcd, the container manager, the CNI plugin, the DNS service or any other addons.

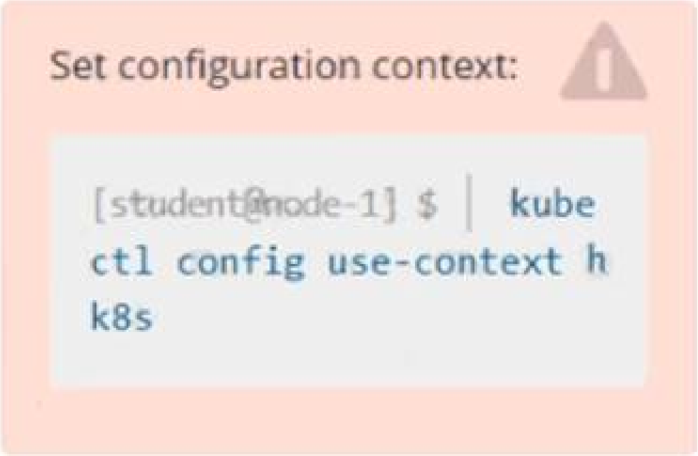
A. Mastered
B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 2
Score: 4%



Set configuration context:

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

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

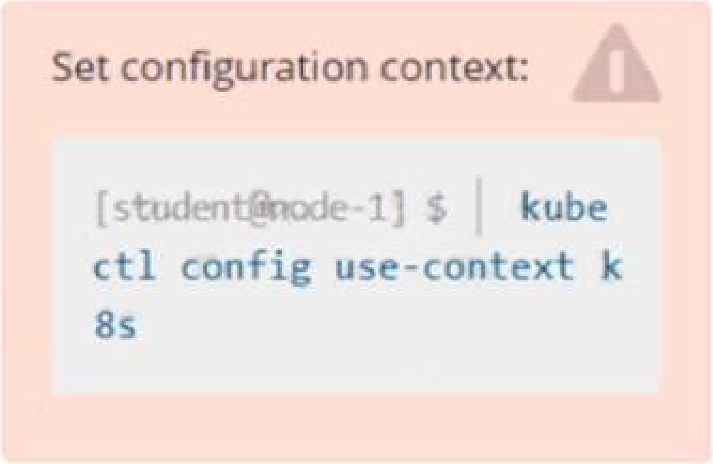
Answer: A

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 3
Score: 5%



Set configuration context:

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

Task
Monitor the logs of pod bar and:
• Extract log lines corresponding to error file-not-found
• Write them to /opt/KUTR00101/bar

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

```
kubectl logs bar | grep 'unable-to-access-website' > /opt/KUTR00101/bar cat /opt/KUTR00101/bar
```

NEW QUESTION 4
Get list of all pods in all namespaces and write it to file “/opt/pods-list.yaml”

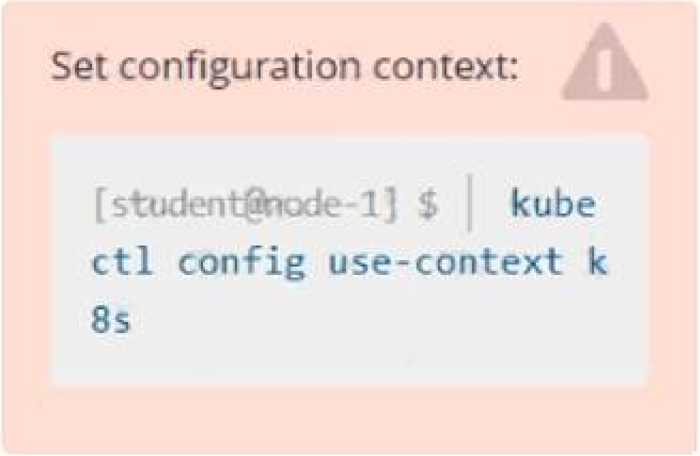
A. Mastered
B. Not Mastered

Answer: A

Explanation:

```
kubectl get po –all-namespaces > /opt/pods-list.yaml
```

NEW QUESTION 5
Score: 7%



Set configuration context:

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

Task
Reconfigure the existing deployment front-end and add a port specification named http exposing port 80/tcp of the existing container nginx.
Create a new service named front-end-svc exposing the container port http.
Configure the new service to also expose the individual Pods via a NodePort on the nodes on which they are scheduled.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
kubectl get deploy front-end
kubectl edit deploy front-end -o yaml
#port specification named http
#service.yaml apiVersion: v1
kind: Service metadata:
name: front-end-svc labels:
app: nginx spec: ports:
- port: 80 protocol: tcp name: http selector: app: nginx
type: NodePort
# kubectl create -f service.yaml
# kubectl get svc
# port specification named http
kubectl expose deployment front-end --name=front-end-svc --port=80 --tarport=80 --type=NodePort

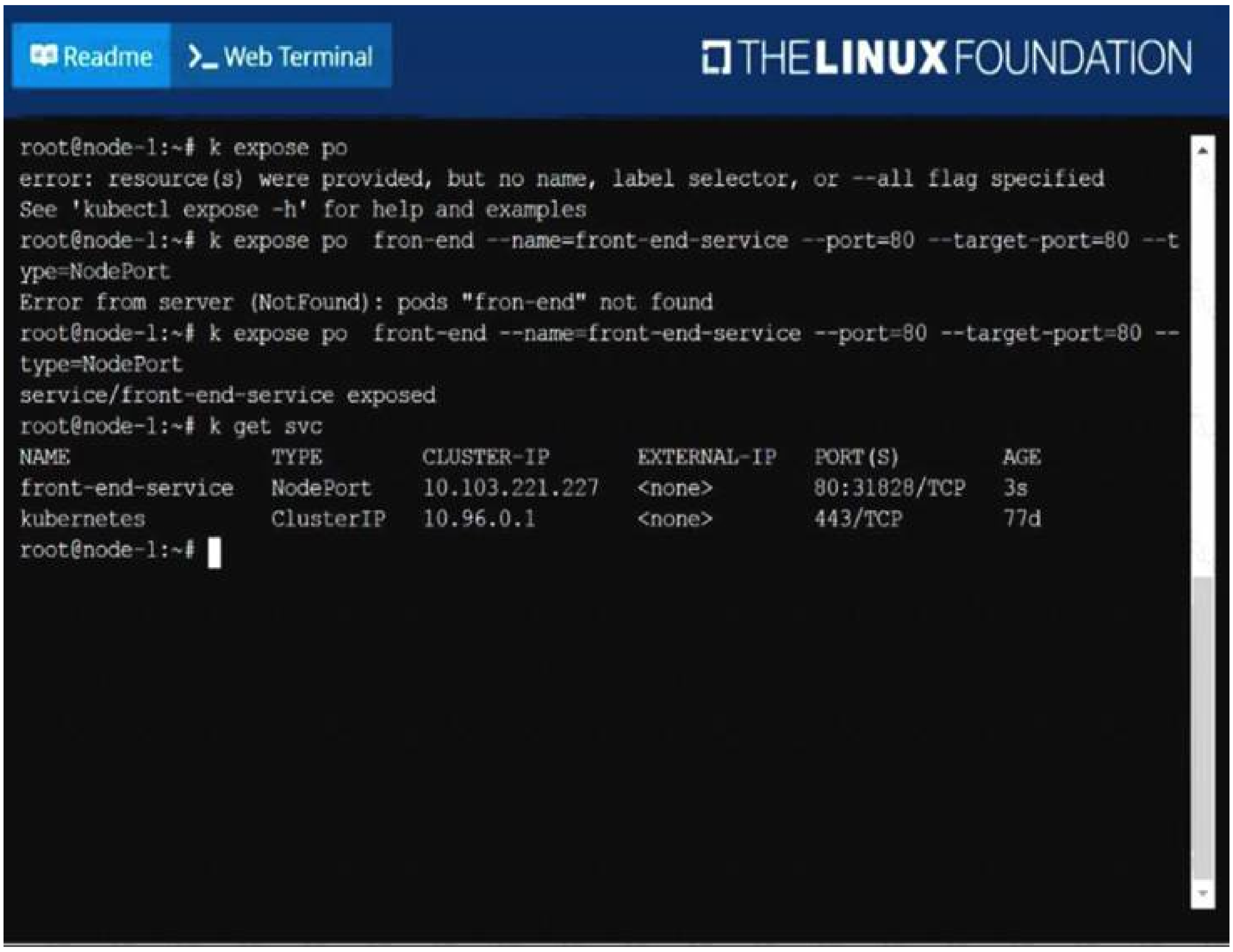
NEW QUESTION 6
Create and configure the service front-end-service so it's accessible through NodePort and routes to the existing pod named front-end.

A. Mastered
B. Not Mastered

Answer: A

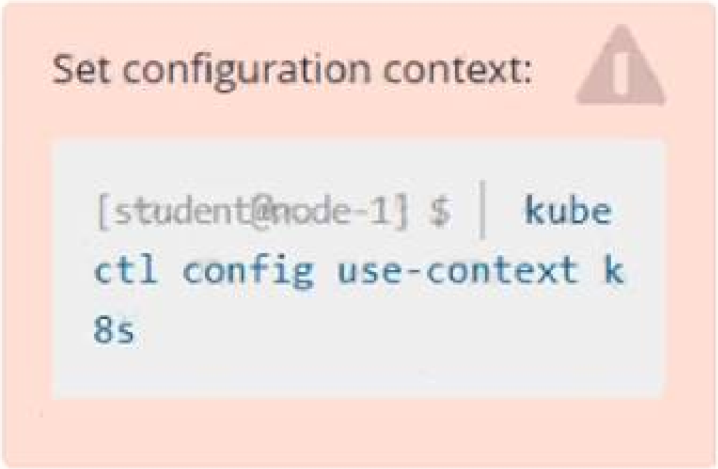
Explanation:

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



```
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:~#
```

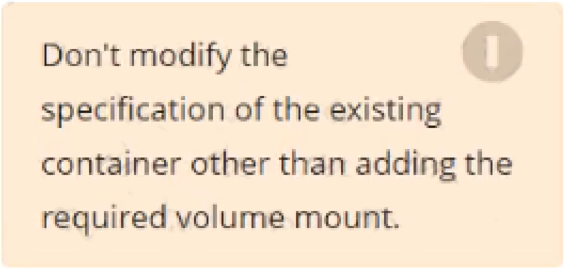
NEW QUESTION 7
Score:7%



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. g. kubectl logs). Adding a streaming sidecar container is a good and common way to accomplish this requirement.
Task
Add a sidecar container named sidecar, using the busybox Image, to the existing Pod big-corp-app. The new sidecar container has to run the following command:
/bin/sh -c tail -n+1 -f /va r/log/big-corp-app.log
Use a Volume, mounted at /var/log, to make the log file big-corp-app.log available to the sidecar container.



Don't modify the specification of the existing container other than adding the required volume mount.

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
```

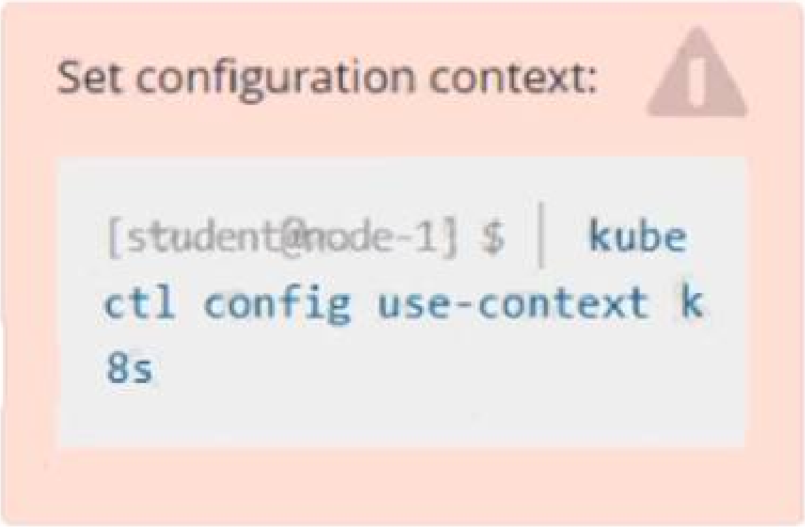
NEW QUESTION 8
List all the pods sorted by name

A. Mastered
B. Not Mastered

Answer: A

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

NEW QUESTION 9
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 10
Create a pod as follows:

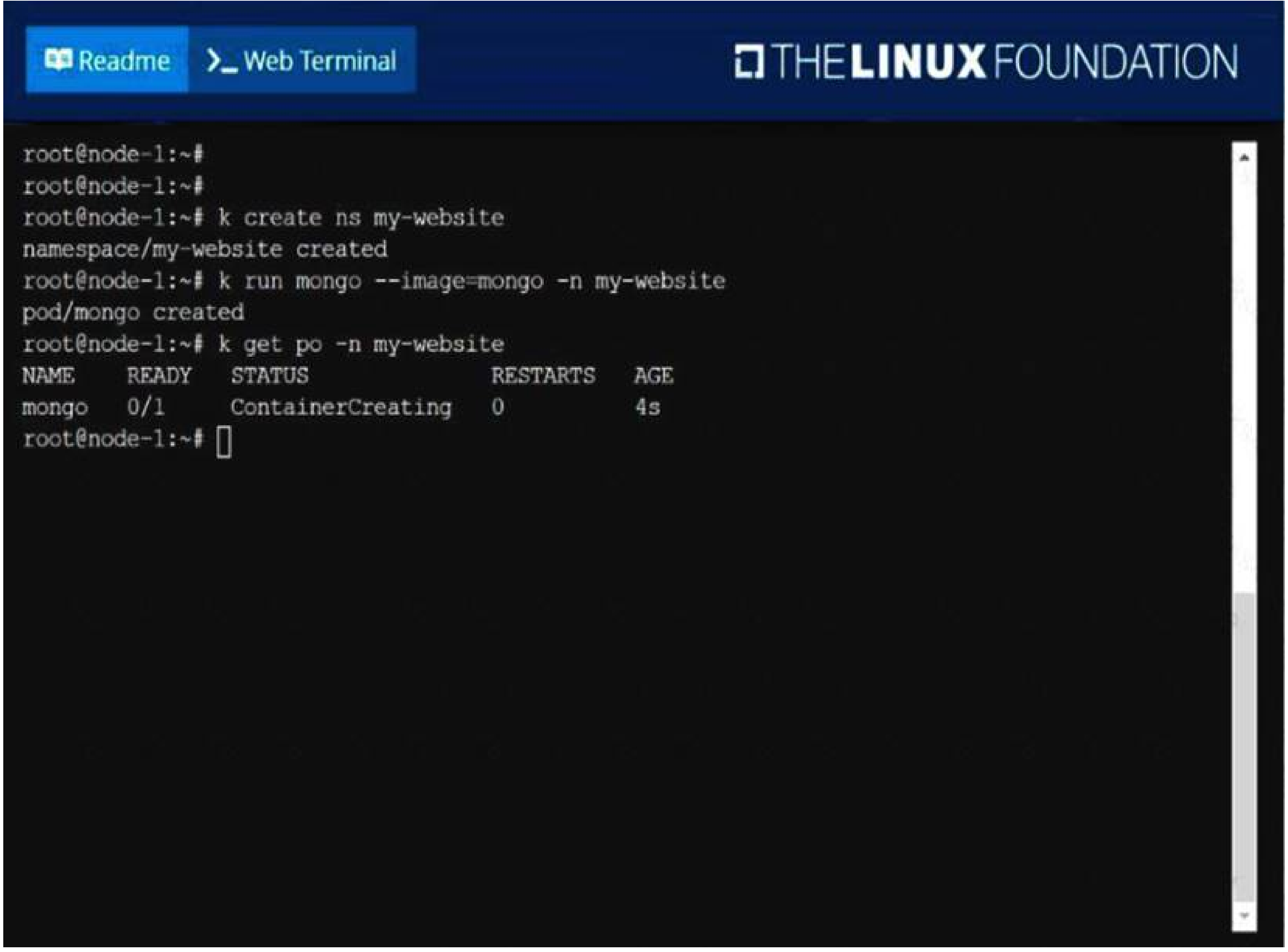
- Name: mongo
- Using Image: mongo
- In a new Kubernetes namespace named: my-website

A. Mastered
B. Not Mastered

Answer: A

Explanation:

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



```
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 10
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 11
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 16
For this item, you will have to ssh to the nodes ik8s-master-0 and ik8s-node-0 and complete all tasks on these nodes. Ensure that you return to the base node (hostname: node-1) when you have completed this item.

Context
As an administrator of a small development team, you have been asked to set up a Kubernetes cluster to test the viability of a new application.
Task
You must use kubeadm to perform this task. Any kubeadm invocations will require the use of the
--ignore-preflight-errors=all option.

- Configure the node ik8s-master-O as a master node. .
- Join the node ik8s-node-o to the cluster.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

solution
You must use the kubeadm configuration file located at /etc/kubeadm.conf when initializingyour cluster.
You may use any CNI plugin to complete this task, but if you don't have your favourite CNI plugin's manifest URL at hand, Calico is one popular option: https://docs.projectcalico.org/v3.14/manifests/calico.yaml
Docker is already installed on both nodes and apt has been configured so that you can install the required tools.

NEW QUESTION 17
List all the pods sorted by name

A. Mastered
B. Not Mastered

Answer: A

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

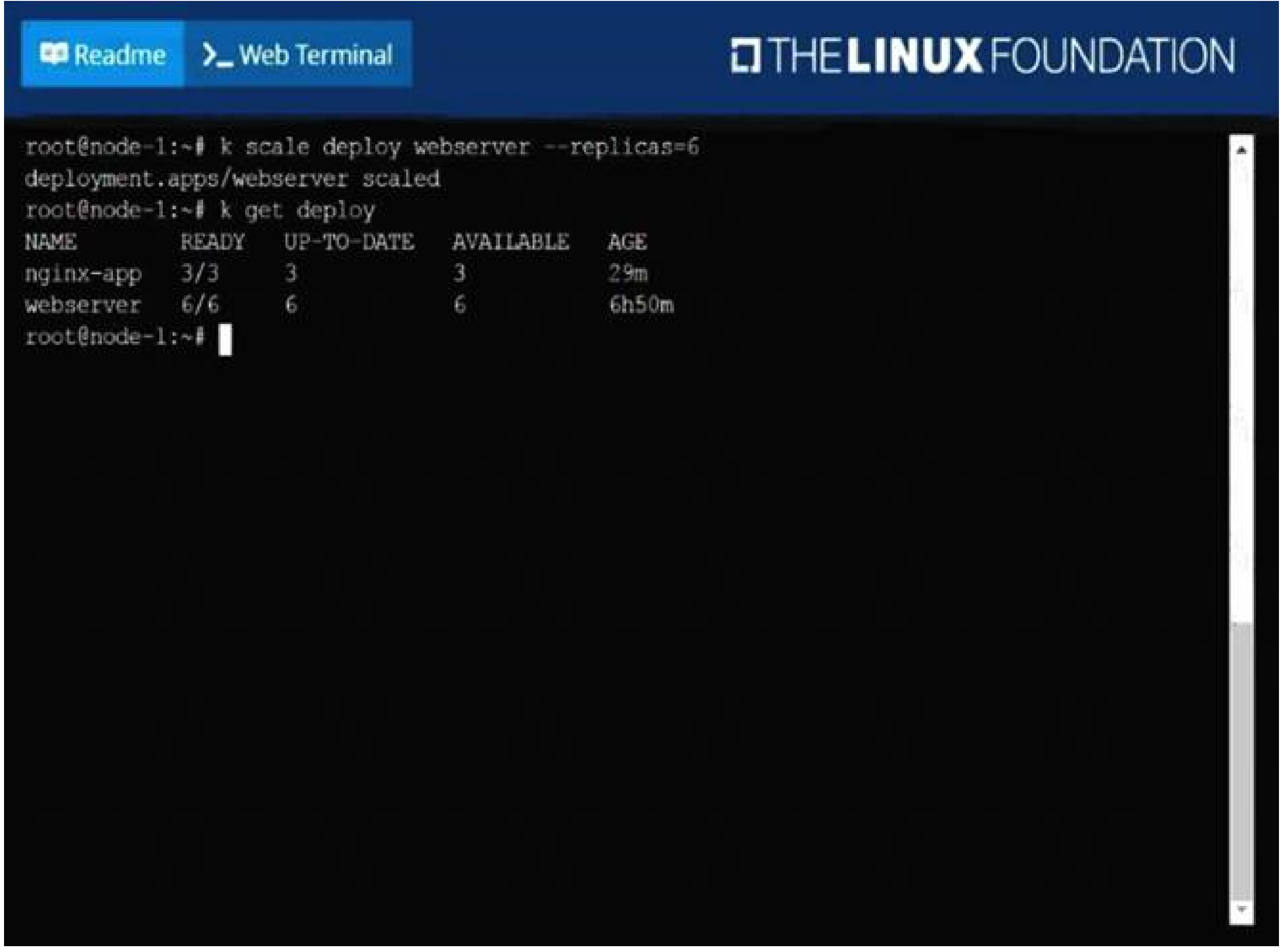
NEW QUESTION 20
Scale the deployment webserver to 6 pods.

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 23
Create a deployment as follows:

-

Name: nginx-random

- Exposed via a service nginx-random
- Ensure that the service & pod are accessible via their respective DNS records
- The container(s) within any pod(s) running as a part of this deployment should use the nginx Image

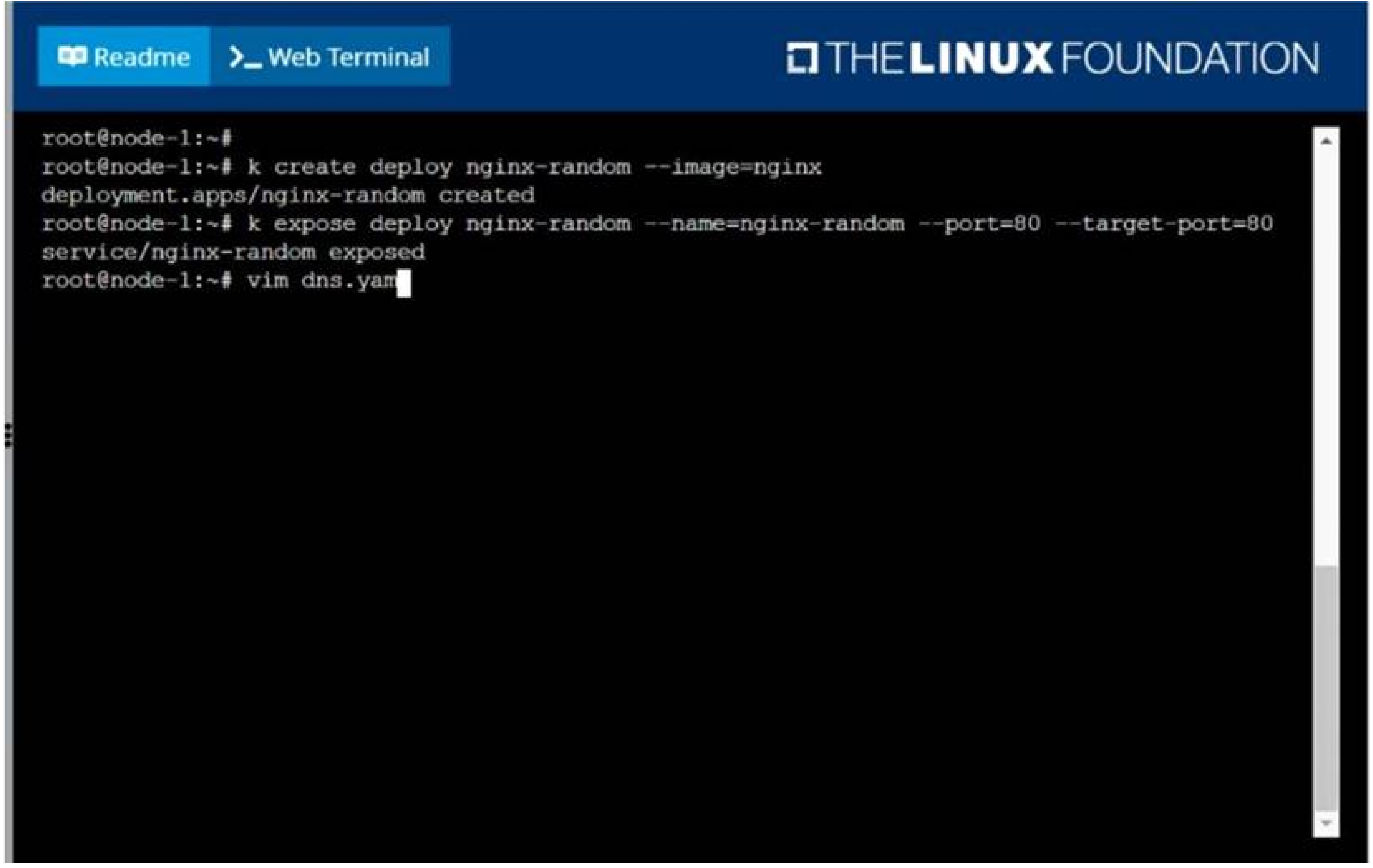
Next, use the utility nslookup to look up the DNS records of the service & pod and write the output to /opt/KUNW00601/service.dns and /opt/KUNW00601/pod.dns respectively.

A. Mastered
B. Not Mastered

Answer: A

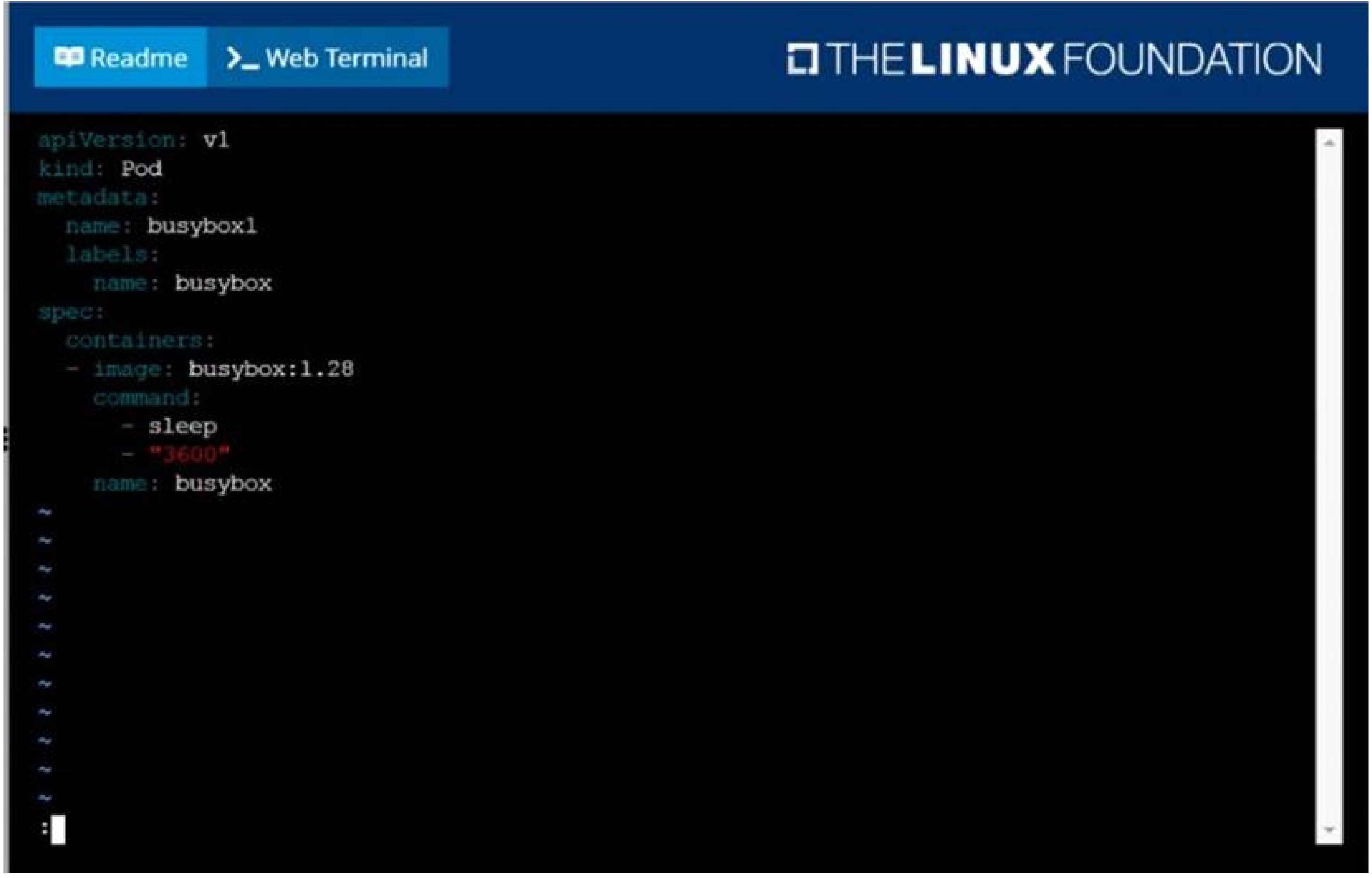
Explanation:

Solution:
F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 C.JPG



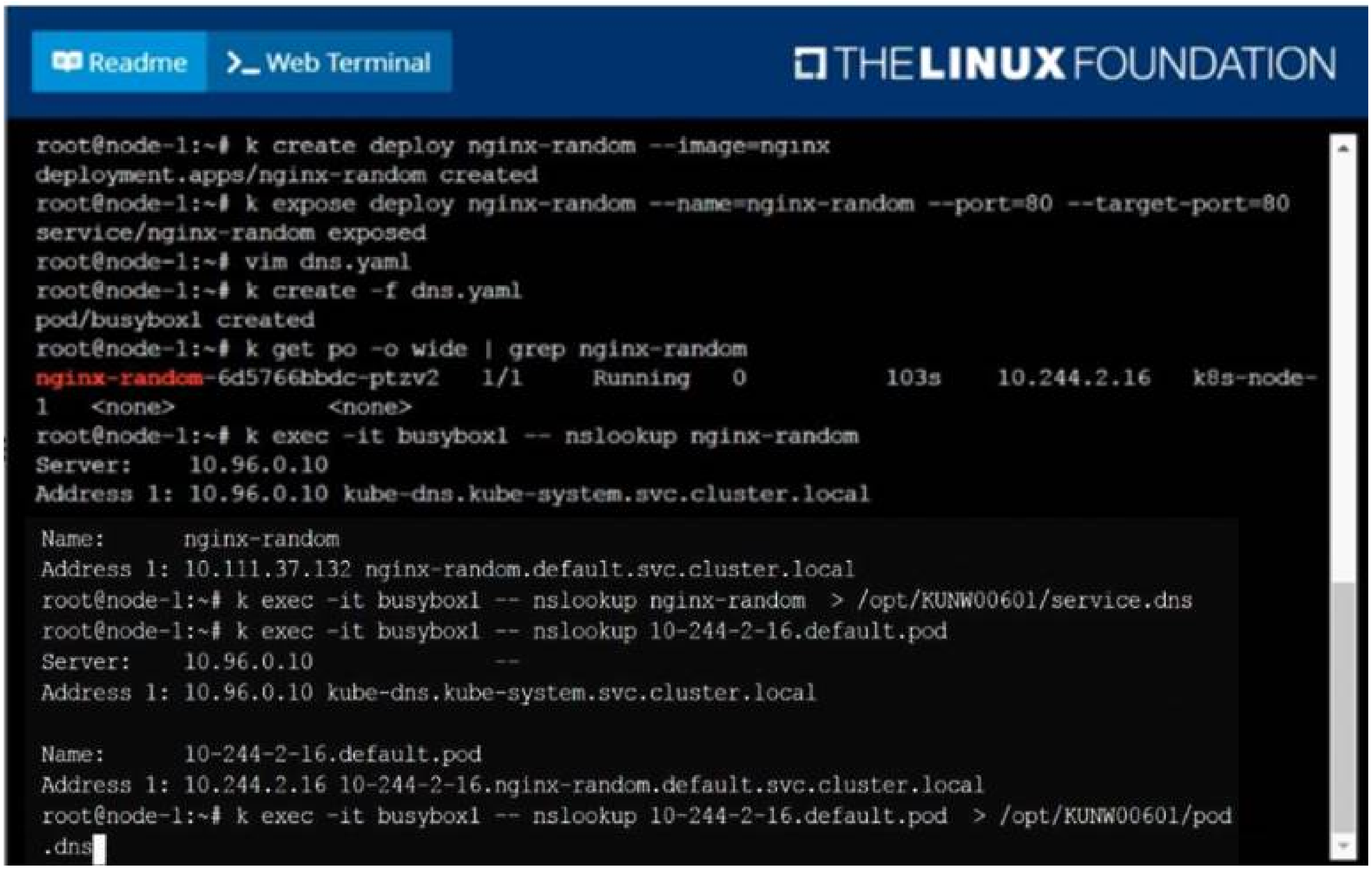
```
root@node-1:~#
root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yam
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 D.JPG



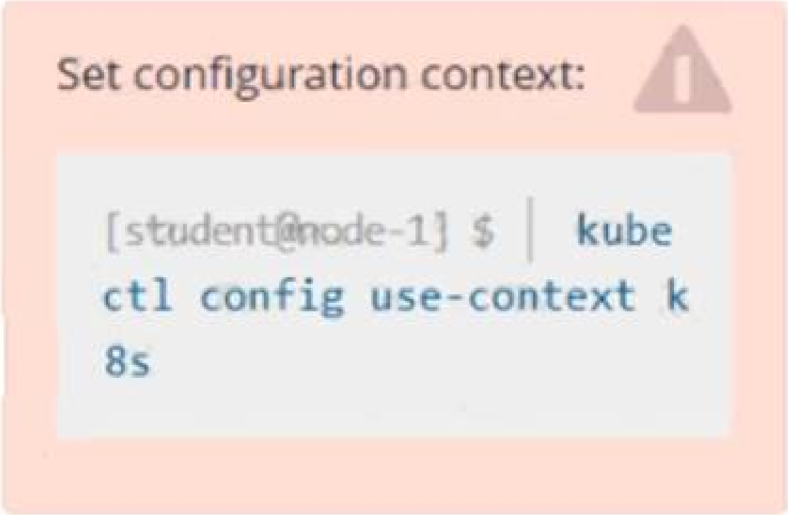
```
apiVersion: v1
kind: Pod
metadata:
  name: busybox1
  labels:
    name: busybox
spec:
  containers:
  - image: busybox:1.28
    command:
      - sleep
      - "3600"
    name: busybox
~
~
~
~
~
~
~
~
~
~
~
:
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 E.JPG



```
root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yaml
root@node-1:~# k create -f dns.yaml
pod/busybox1 created
root@node-1:~# k get po -o wide | grep nginx-random
nginx-random-6d5766bbdc-ptzv2   1/1     Running   0          103s    10.244.2.16    k8s-node-
1   <none>           <none>
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      nginx-random
Address 1: 10.111.37.132 nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random  > /opt/KUNW00601/service.dns
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod
Server:    10.96.0.10          --
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      10-244-2-16.default.pod
Address 1: 10.244.2.16 10-244-2-16.nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod  > /opt/KUNW00601/pod
.dns
```

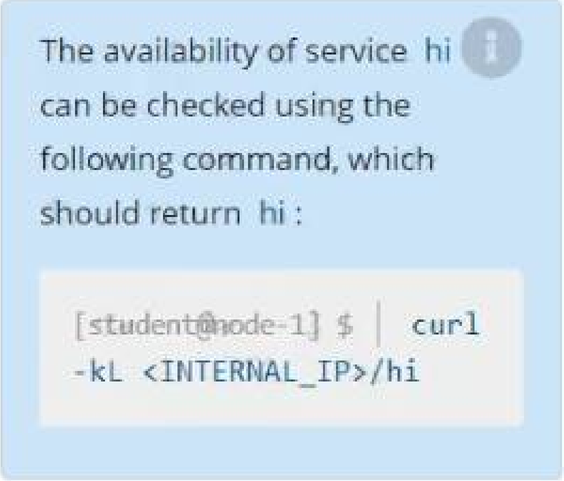
NEW QUESTION 24
Score: 7%



Set configuration context:

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

Task
Create a new nginx Ingress resource as follows:
• Name: ping
• Namespace: ing-internal
• Exposing service hi on path /hi using service port 5678



The availability of service hi can be checked using the following command, which should return hi :

```
[student@node-1] $ | curl -kL <INTERNAL_IP>/hi
```

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
vi ingress.yaml
#

```
apiVersion: networking.k8s.io/v1 kind: Ingress
metadata: name: ping
namespace: ing-internal spec:
rules:
- http:
paths:
- path: /hi pathType: Prefix backend: service:
name: hi port:
number: 5678
#
kubectl create -f ingress.yaml
```

NEW QUESTION 29
Create a busybox pod that runs the command “env” and save the output to “envpod” file

A. Mastered
B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 32
Create a busybox pod and add “sleep 3600” command

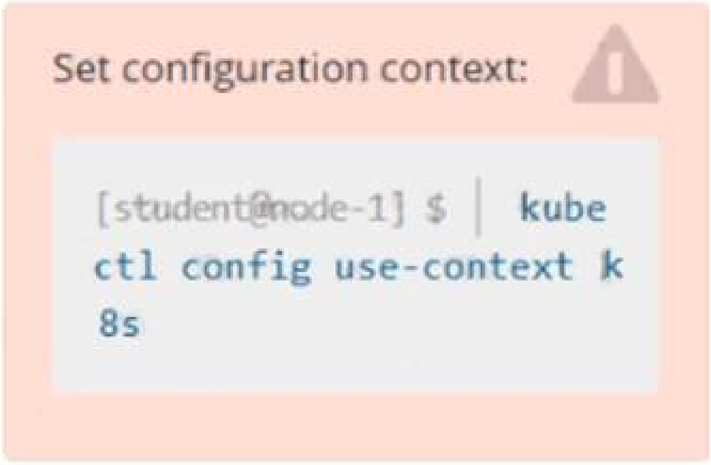
A. Mastered
B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 36
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 38
Print pod name and start time to “/opt/pod-status” file

A. Mastered
B. Not Mastered

Answer: A

Explanation:
kubect1 get pods -o=jsonpath='{range items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'

NEW QUESTION 39
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.

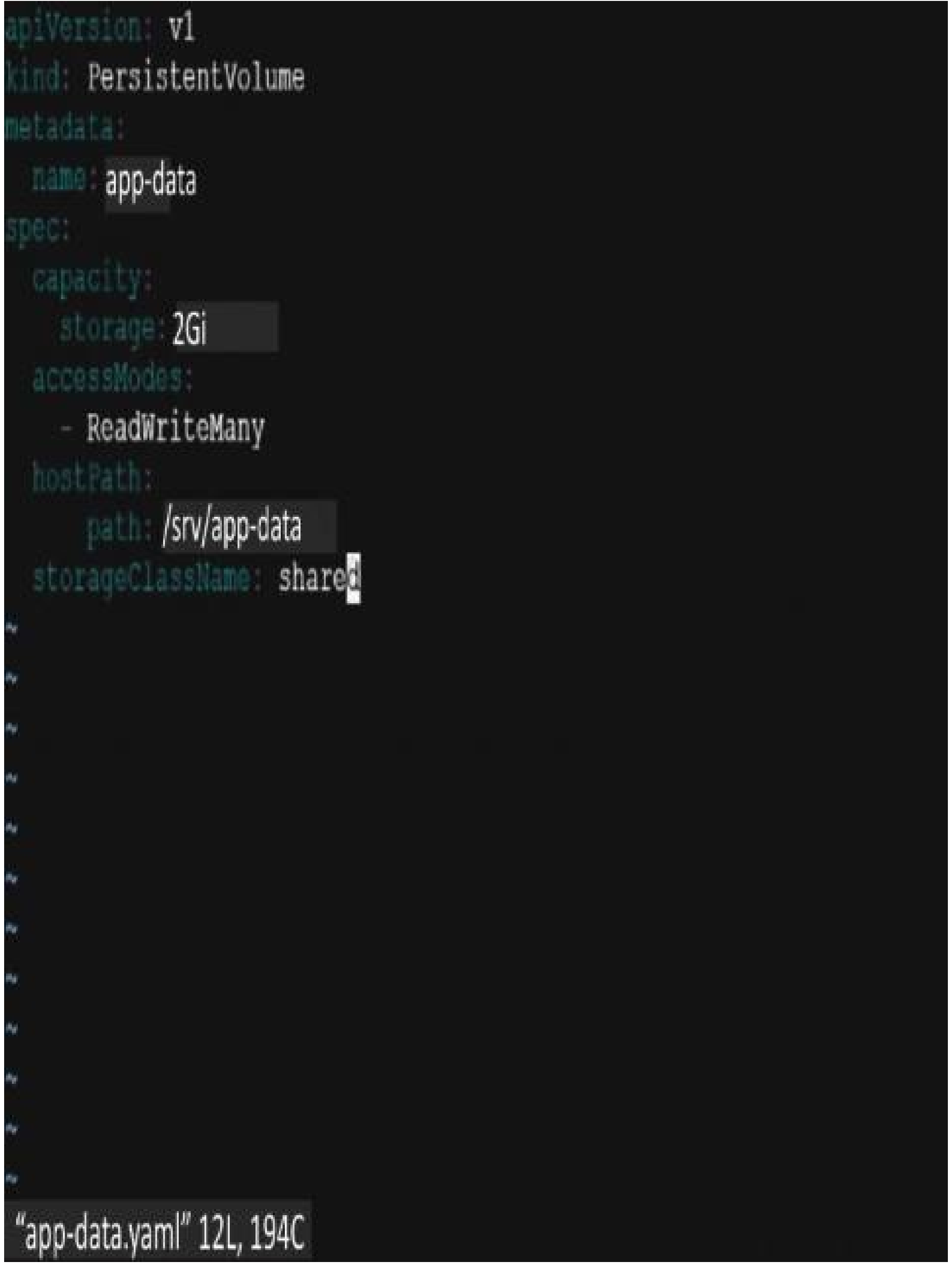
A. Mastered
B. Not Mastered

Answer: A

Explanation:

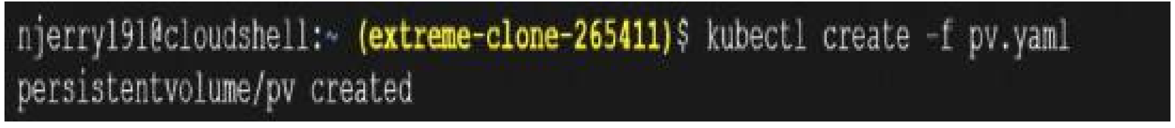
solution
Persistent Volume
A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a cluster-level resource like nodes, which don’t belong to any namespace. It is provisioned by the administrator and has a particular file size. This way, a developer deploying their app 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.



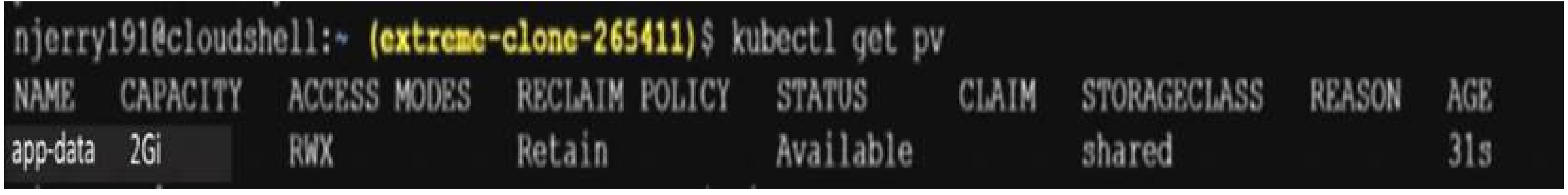
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /srv/app-data
  storageClassName: shared
~
~
~
~
~
~
~
~
~
~
~
~
“app-data.yaml” 12L, 194C
```

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



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

* 3. View the persistent volume.

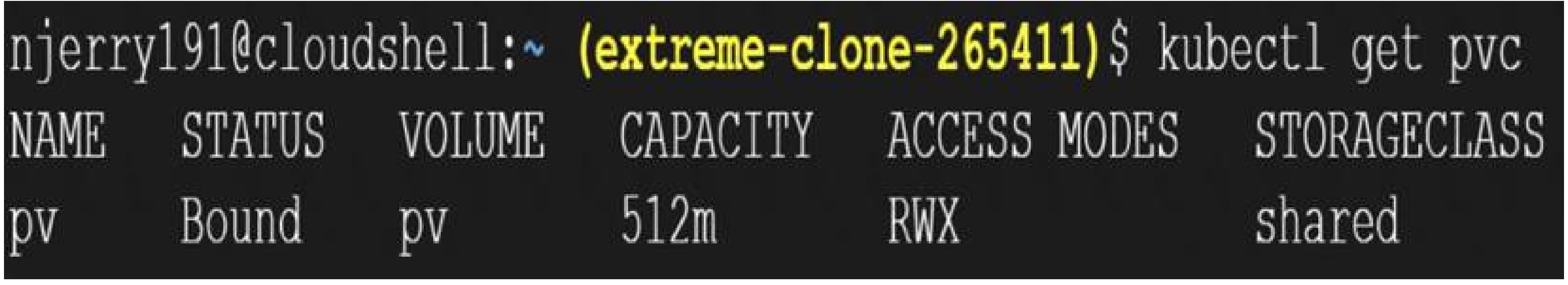


```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
app-data  2Gi        RWX            Retain           Available           shared                  31s
```

- 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

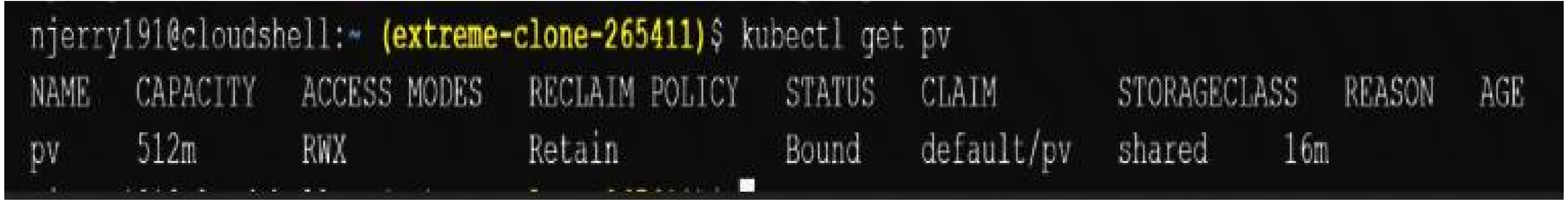
to a persistentVolumeClaim.
PersistentVolumeClaim
In a real ecosystem, a system admin will create the PersistentVolume then a developer will create a PersistentVolumeClaim which will be referenced in a pod. A PersistentVolumeClaim is created by specifying the minimum size and the access mode they require 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: - 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 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: 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 44
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 49
Create a pod with image nginx called nginx and allow traffic on port 80

A. Mastered
B. Not Mastered

Answer: A

Explanation:
kubectl run nginx --image=nginx --restart=Never --port=80

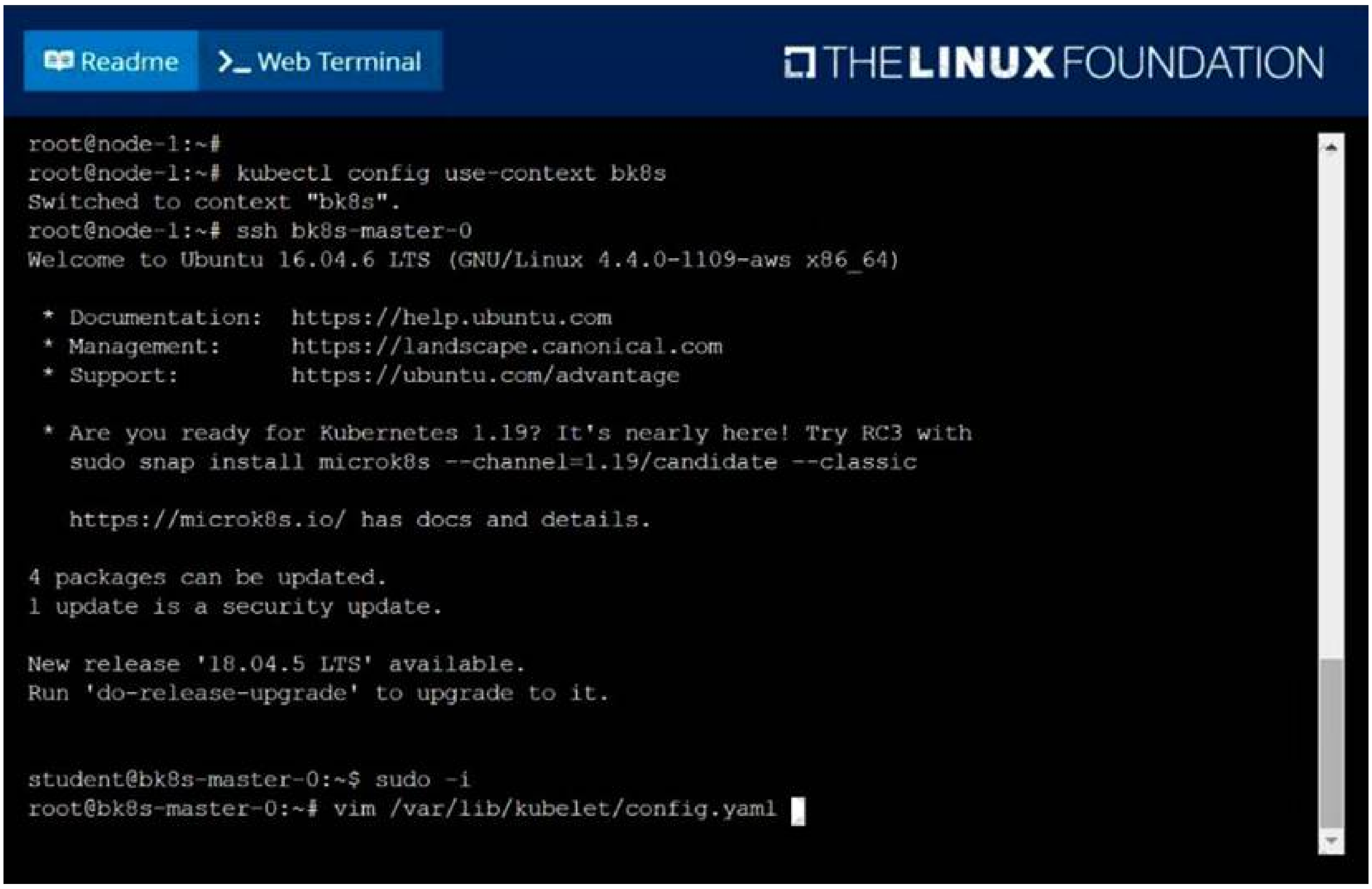
NEW QUESTION 54
Given a partially-functioning Kubernetes cluster, identify symptoms of failure on the cluster.
Determine the node, the failing service, and take actions to bring up the failed service and restore the health of the cluster. Ensure that any changes are made permanently.
You can ssh to the 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

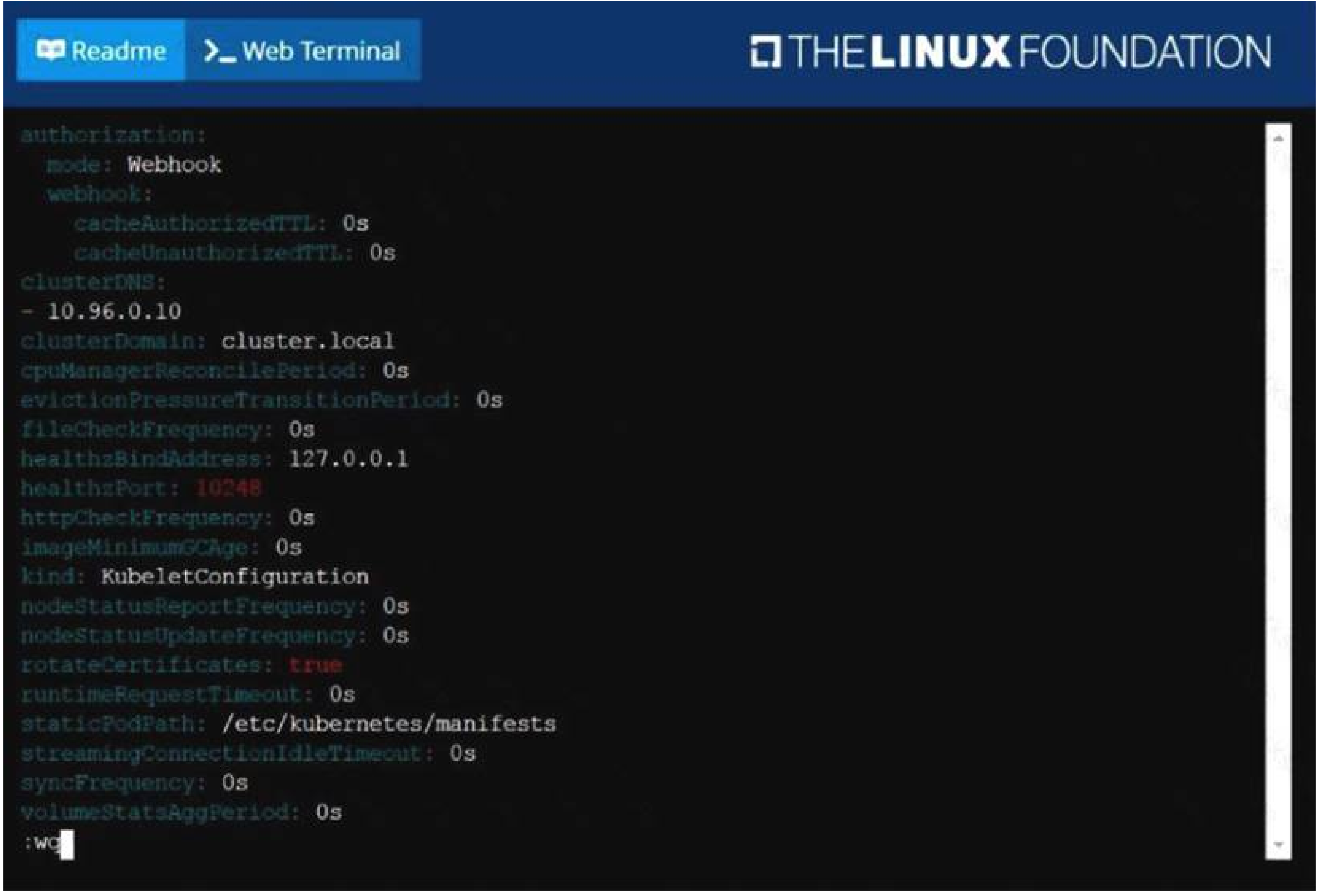
Explanation:

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



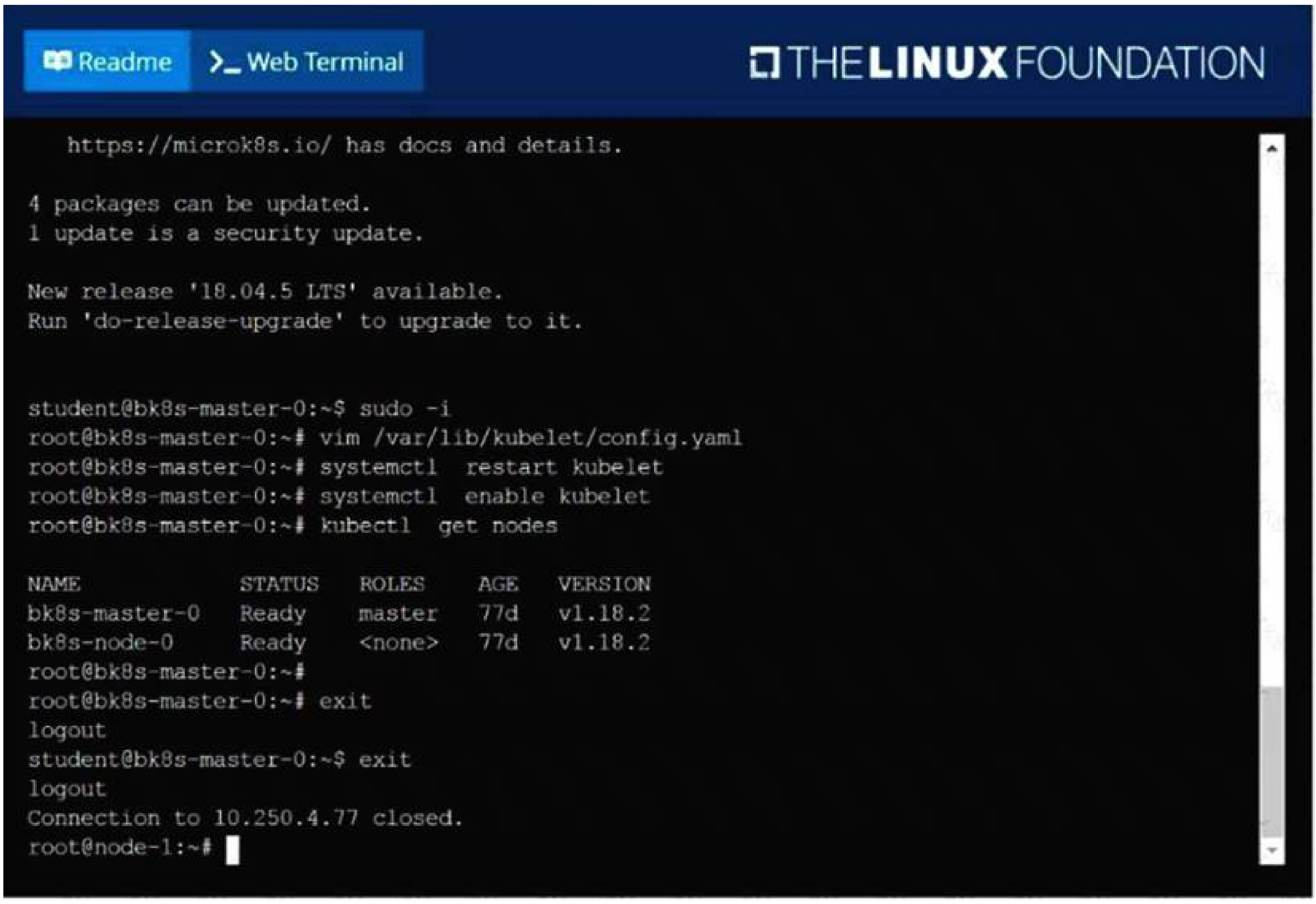
```
root@node-1:~#
root@node-1:~# kubectl config use-context bk8s
Switched to context "bk8s".
root@node-1:~# ssh bk8s-master-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

 * 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 D.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 E.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:~# █
```

NEW QUESTION 55

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

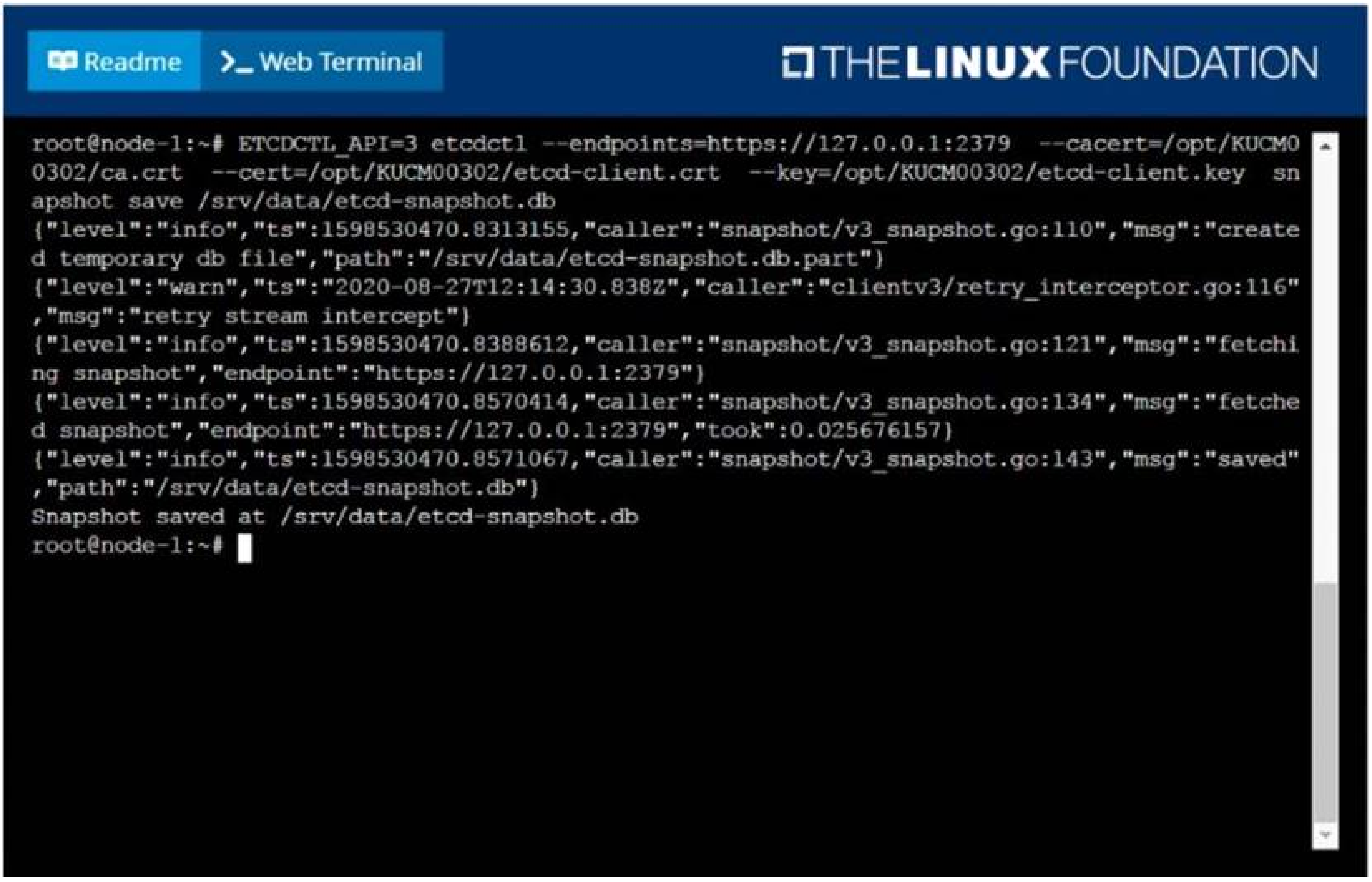
B. Not Mastered

Answer: A

Explanation:

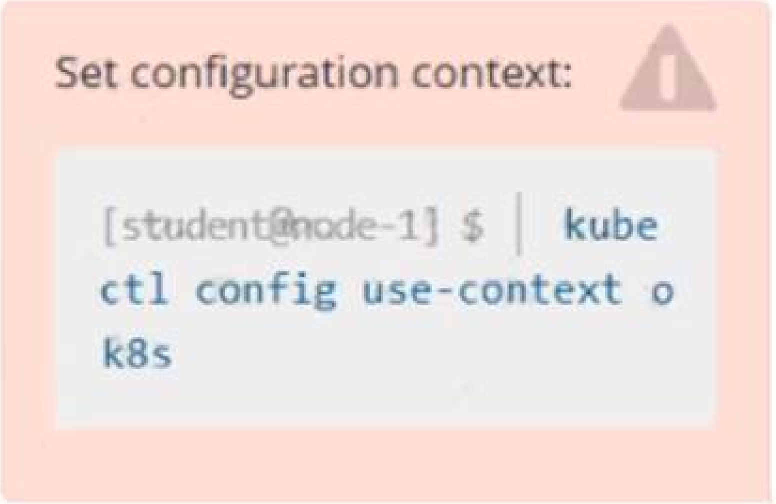
solution

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



```
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:~#
```

NEW QUESTION 56
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.

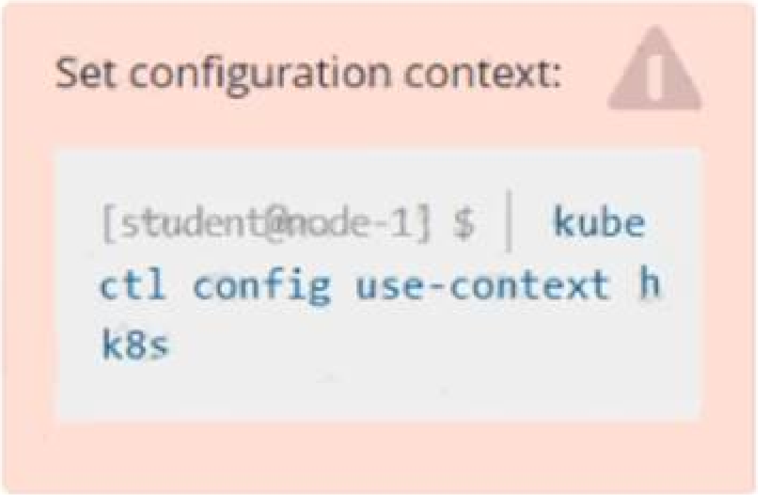
A. Mastered
B. Not Mastered

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 58
Score: 7%



Set configuration context:

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

Task
Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace echo. Ensure that the new NetworkPolicy allows Pods in namespace my-app to connect to port 9000 of Pods in namespace echo.
Further ensure that the new NetworkPolicy:
• does not allow access to Pods, which don't listen on port 9000
• does not allow access from Pods, which are not in namespace my-app

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:

```
#network.yaml
apiVersion: networking.k8s.io/v1 kind: NetworkPolicy
metadata:
name: allow-port-from-namespace namespace: internal
spec: podSelector: matchLabels: {
}
policyTypes:
- Ingress ingress:
- from:
- podSelector: {
}
ports:
- protocol: TCP port: 8080
#spec.podSelector namespace pod kubectl create -f network.yaml
```

NEW QUESTION 61
......

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

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

100% Actual & Verified — Instant Download, Please Click
Order The CKA Practice Test Here