# Linux-Foundation

## Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

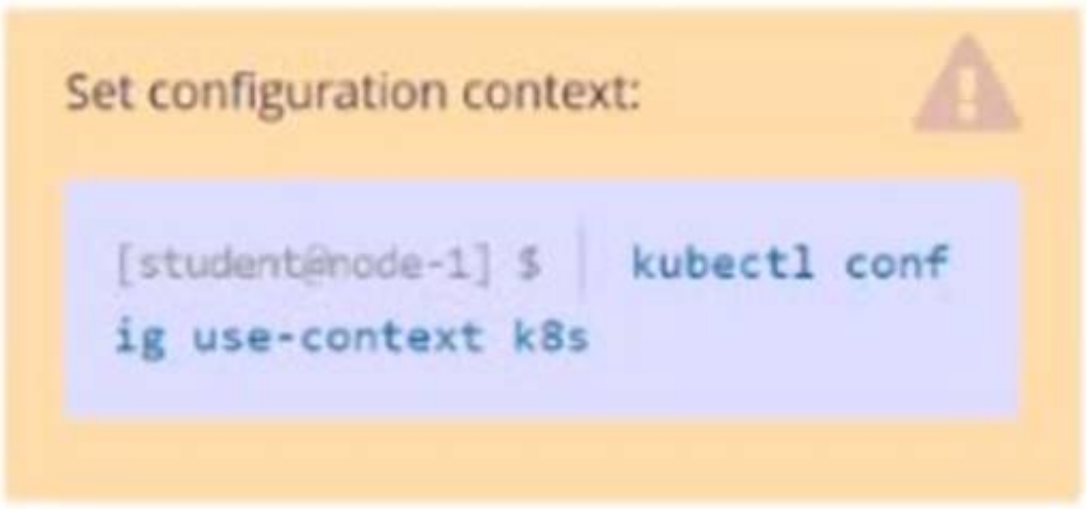
**NEW QUESTION 1**
CORRECT TEXT
Create a pod with image nginx called nginx and allow traffic on port 80

A. Mastered
B. Not Mastered

**Answer:** A

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

**NEW QUESTION 2**
CORRECT TEXT
Task Weight: 4%



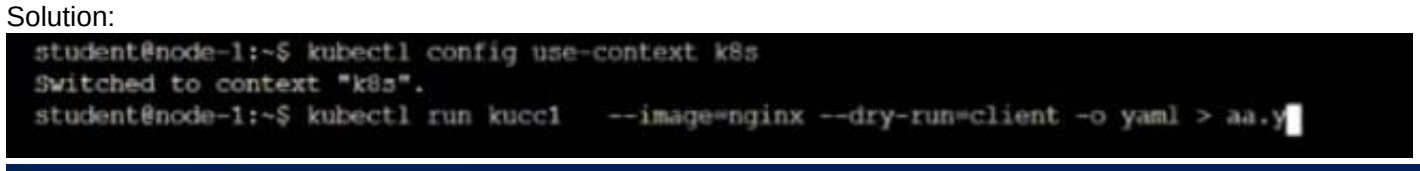

Task
Schedule a Pod as follows:
• Name: kucc1
• App Containers: 2
• Container Name/Images: o nginx
o consul

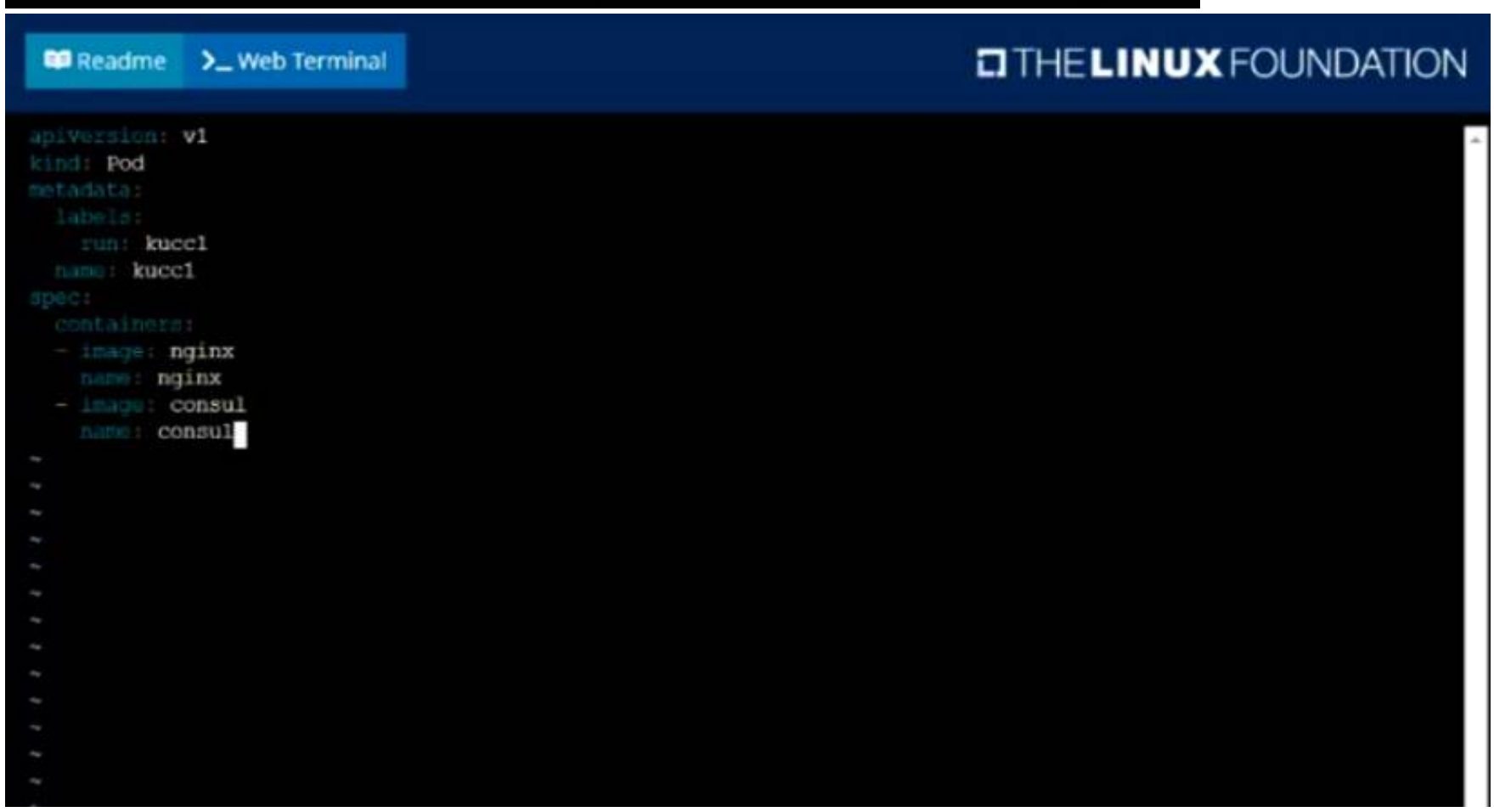
A. Mastered
B. Not Mastered

**Answer:** A

**Explanation:**

Solution:




Graphical user interface, text, application
Description automatically generated

```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl run kucc1   --image=nginx --dry-run=client -o yaml > aa.yaml
student@node-1:~$ vim aa.yaml
student@node-1:~$ kubectl   create  -f aa.yaml
pod/kucc1 created
student@node-1:~$ kubectl get pods
NAME                        READY   STATUS              RESTARTS   AGE
ll-factor-app               1/1     Running             0          6h34m
cpu-loader-98b9se           1/1     Running             0          6h33m
cpu-loader-ab2d3s           1/1     Running             0          6h33m
cpu-loader-kipb9a           1/1     Running             0          6h33m
foobar                      1/1     Running             0          6h34m
front-end-6bc87b9748-24rcm  1/1     Running             0          5m4s
front-end-6bc87b9748-hd5wp  1/1     Running             0          5m2s
kucc1                       0/2     ContainerCreating   0          3s
nginx-kusc00401             1/1     Running             0          2m28s
webserver-84c89dfd75-2dljn  1/1     Running             0          6h38m
webserver-84c89dfd75-8d8x2  1/1     Running             0          6h38m
webserver-84c89dfd75-z5zz4  1/1     Running             0          3m51s
student@node-1:~$
```

Text Description automatically generated

**NEW QUESTION 3**
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 4**
CORRECT TEXT
Score:7%


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 5**
CORRECT TEXT
Score: 4%


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 6**
CORRECT TEXT
Get IP address of the pod – “nginx-dev”

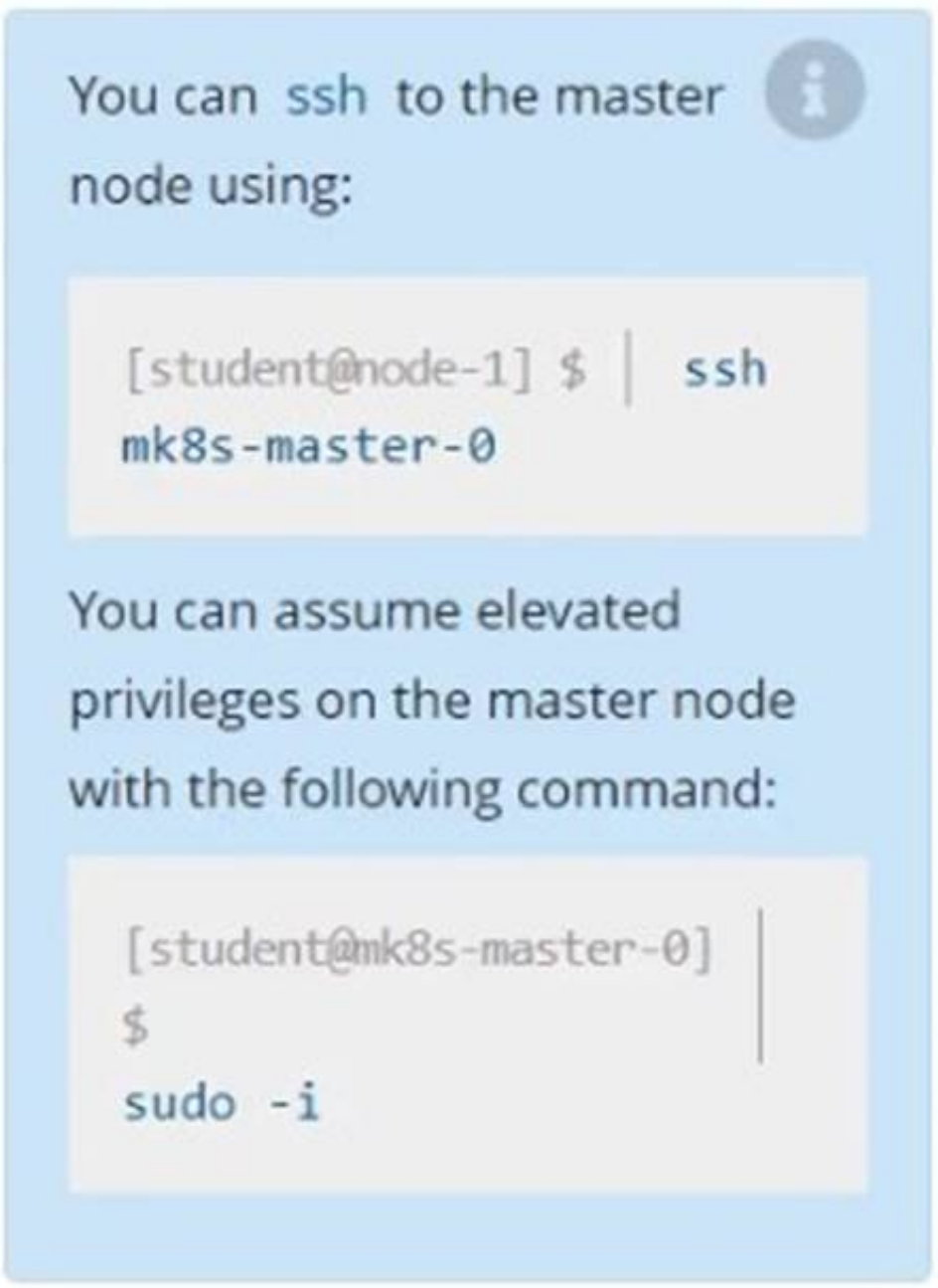
A. Mastered
B. Not Mastered

**Answer:** A

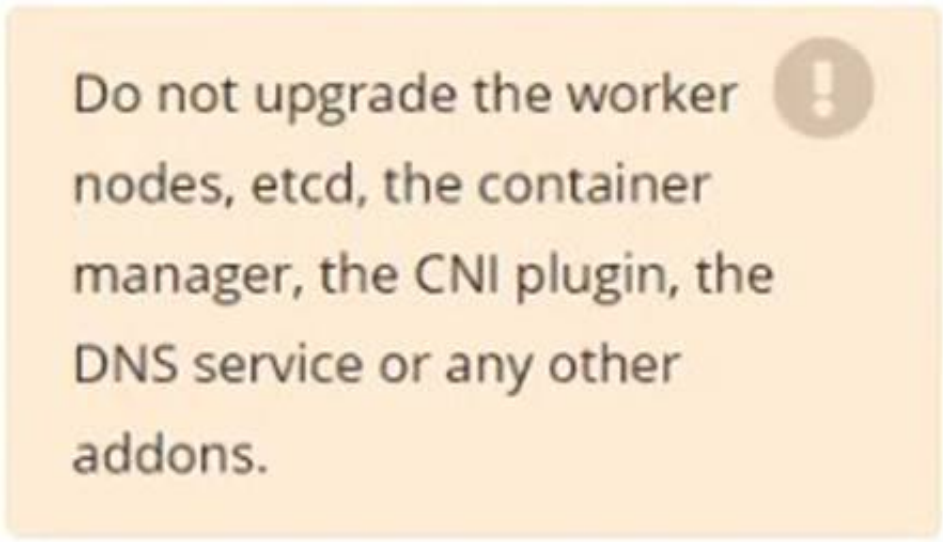
**Explanation:**
Kubect1 get po -o wide
Using JsonPath
kubect1 get pods -o=jsonpath='{range
items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'

**NEW QUESTION 7**
CORRECT TEXT
Score: 7%


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




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 8**
CORRECT TEXT
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 9**
CORRECT TEXT
Score: 5%


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:

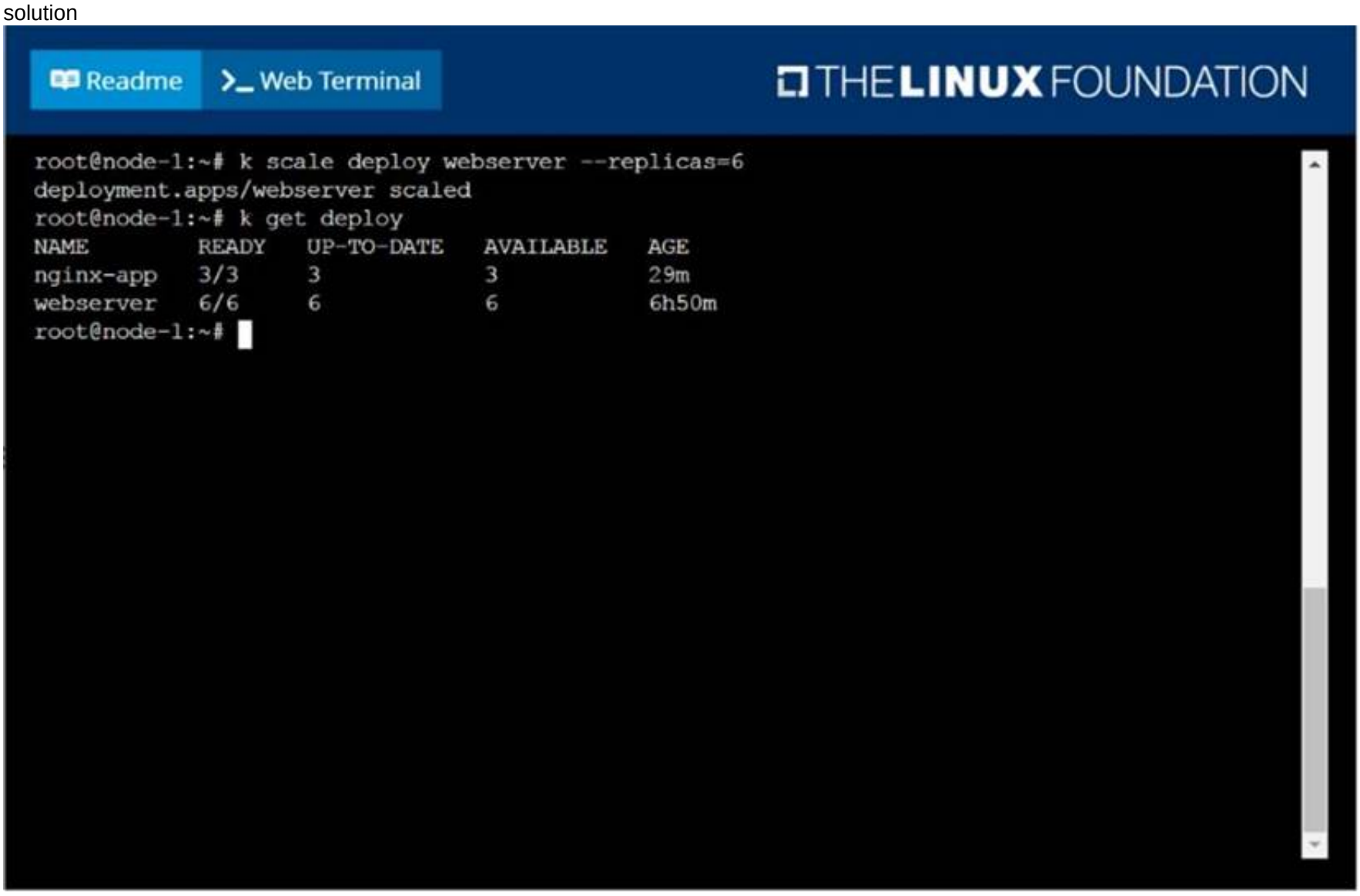
kubectl top -l name=cpu-user -A
echo 'pod name' >> /opt/KUT00401/KUT00401.txt

**NEW QUESTION 10**
CORRECT TEXT
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

**NEW QUESTION 10**
CORRECT TEXT
Score: 7%


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 15**
CORRECT TEXT
Check the Image version of nginx-dev pod using jsonpath

A. Mastered
B. Not Mastered

**Answer:** A

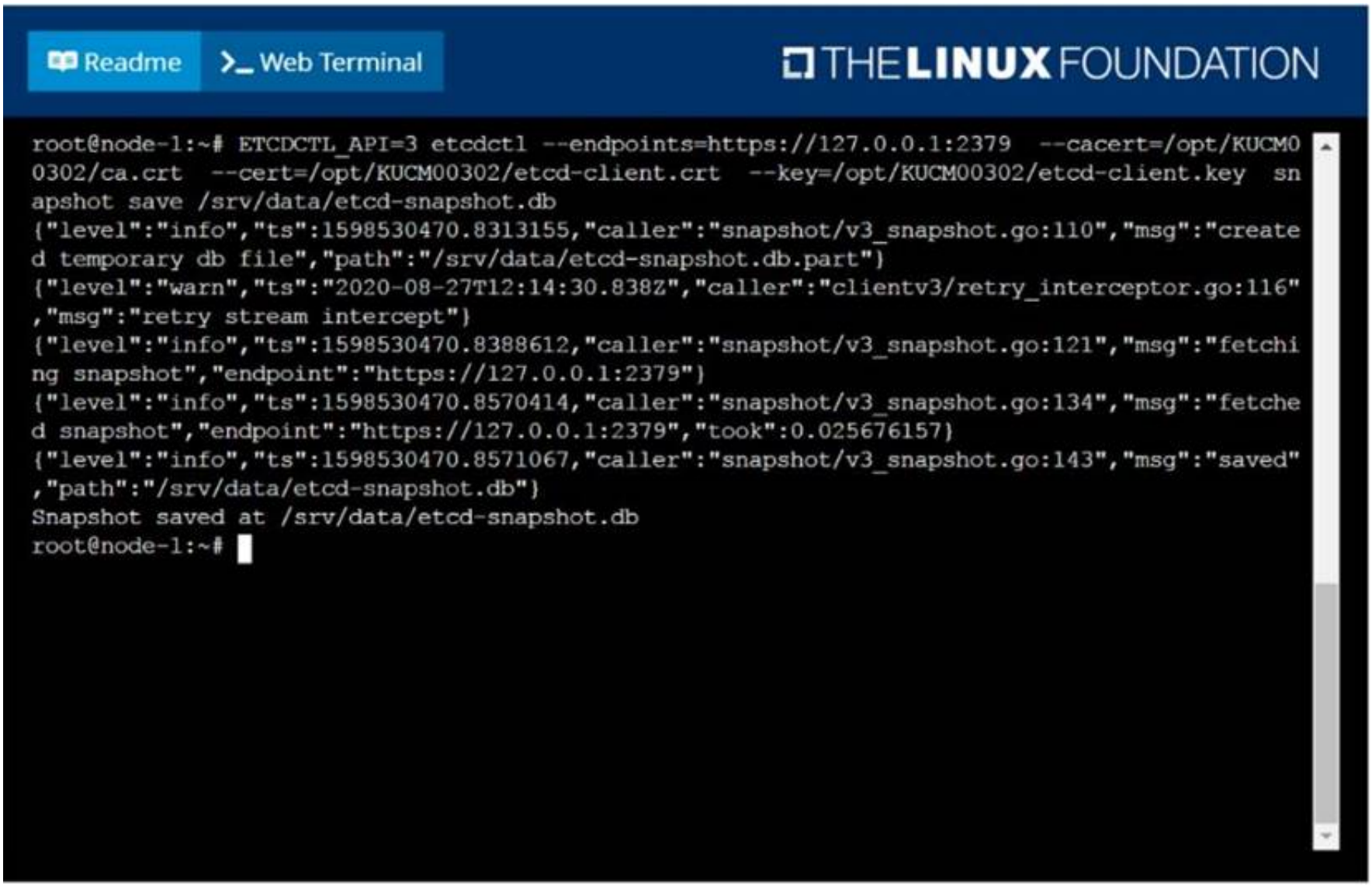
**Explanation:**
kubect1 get po nginx-dev -o
jsonpath='{.spec.containers[].image}{"\n"}'

**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

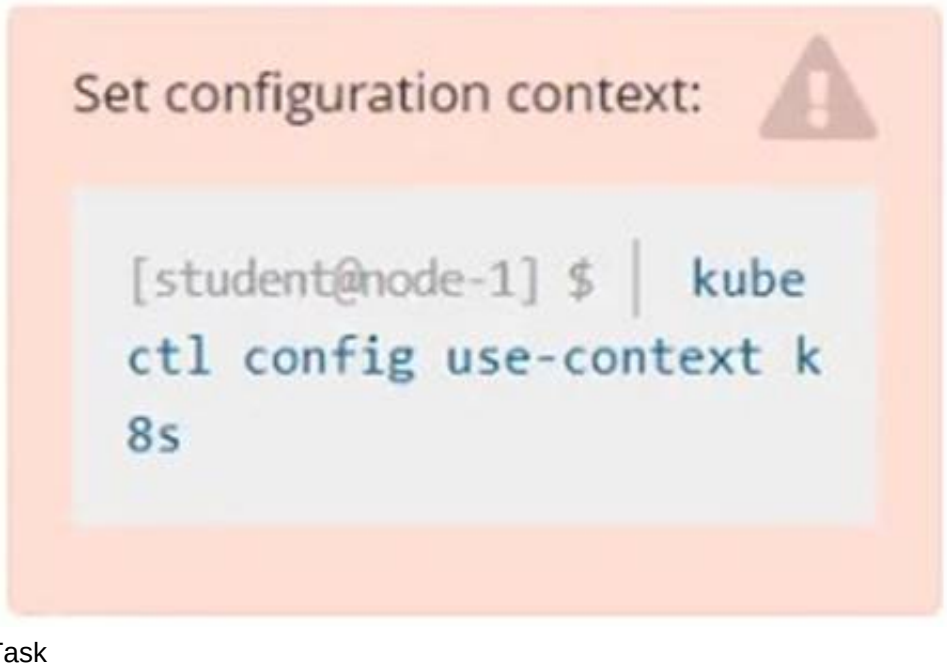
**Answer:** A

**Explanation:**
solution




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

**NEW QUESTION 21**
CORRECT TEXT
Score: 7%




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 23**
CORRECT TEXT
Check the image version in pod without the describe command

A. Mastered
B. Not Mastered

**Answer:** A

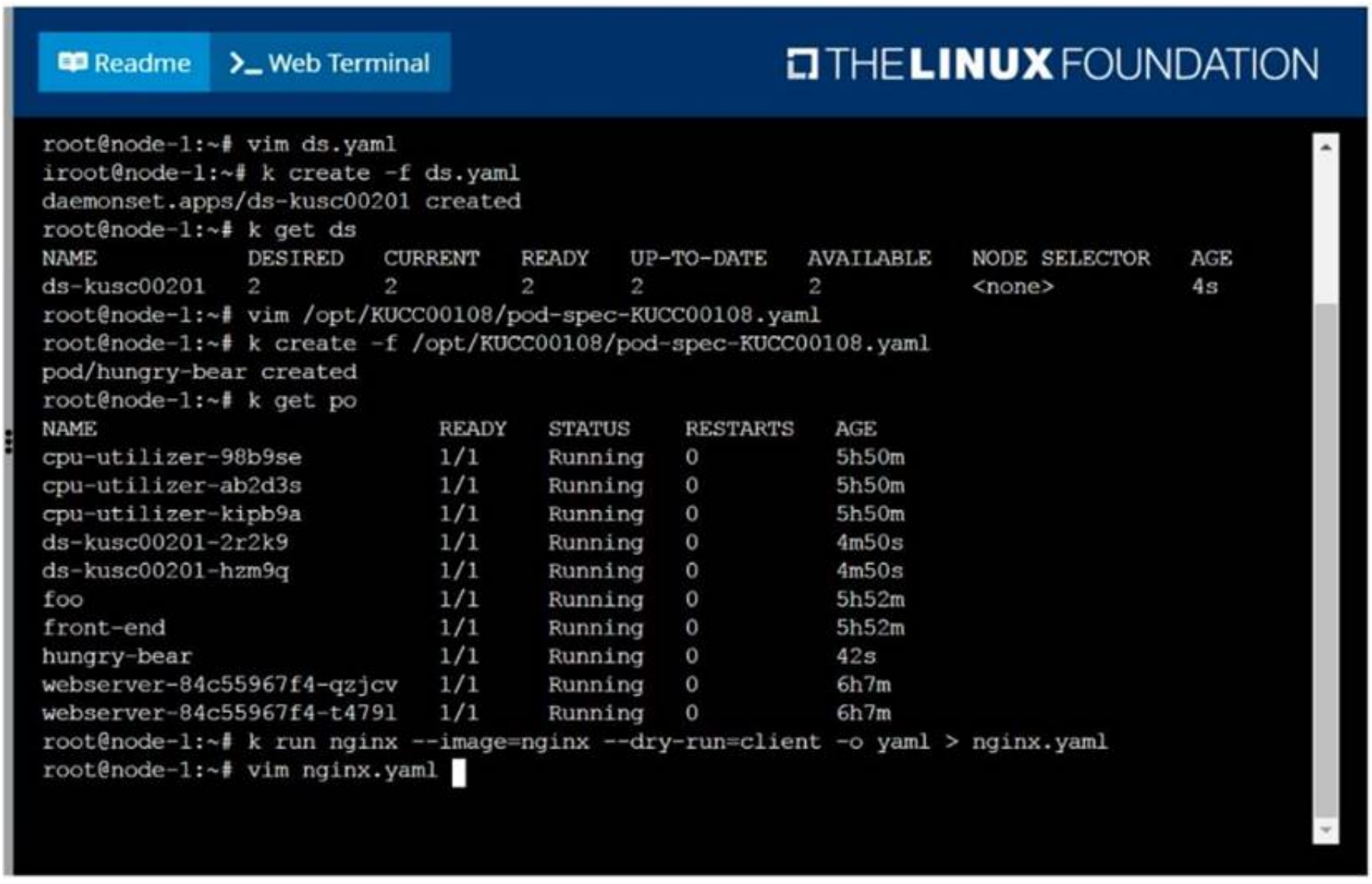
**Explanation:**
kubectl get po nginx -o
jsonpath='{.spec.containers[].image}{"\n"}'

**NEW QUESTION 25**
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.

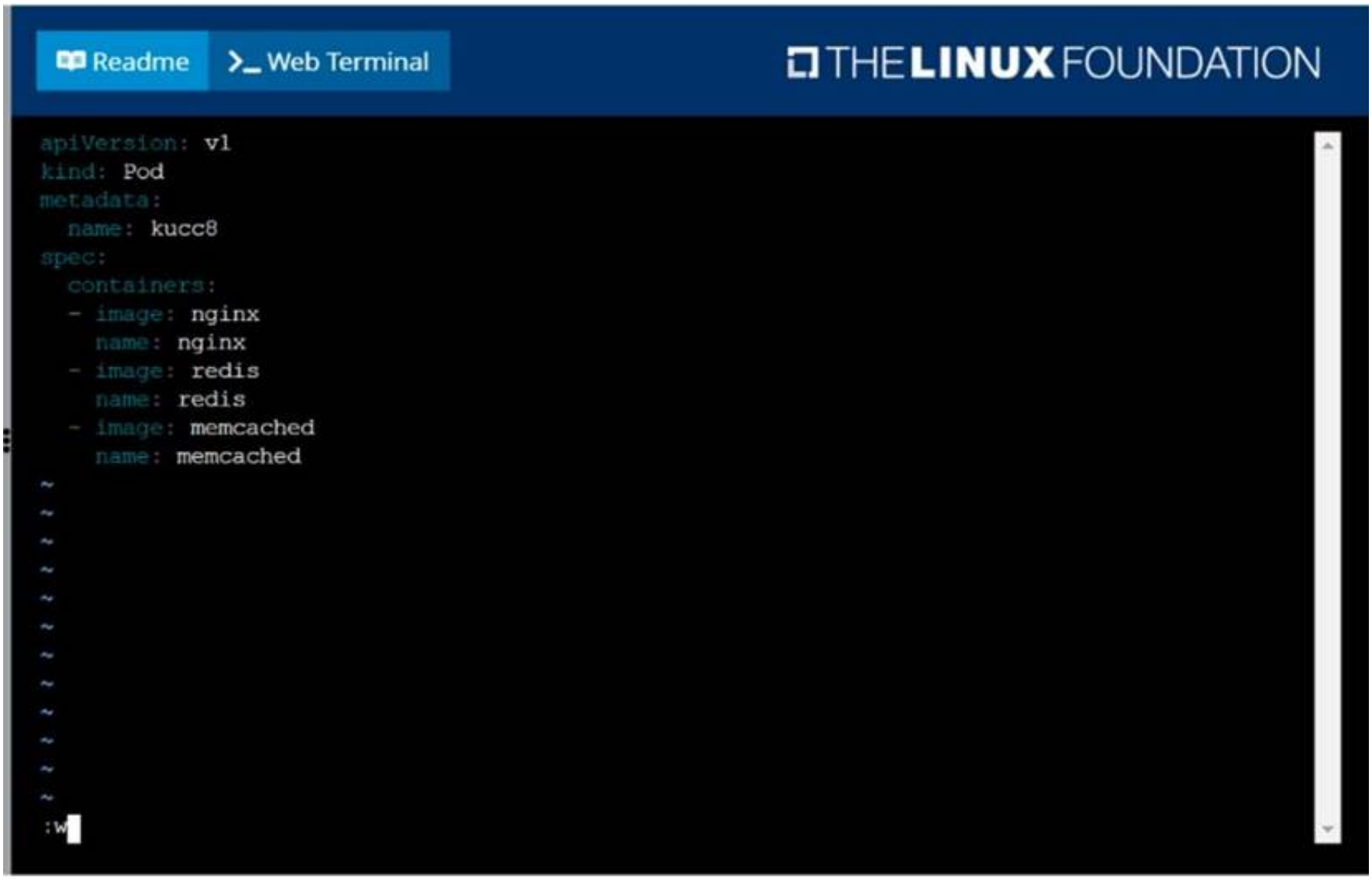
A. Mastered
B. Not Mastered

**Answer:** A

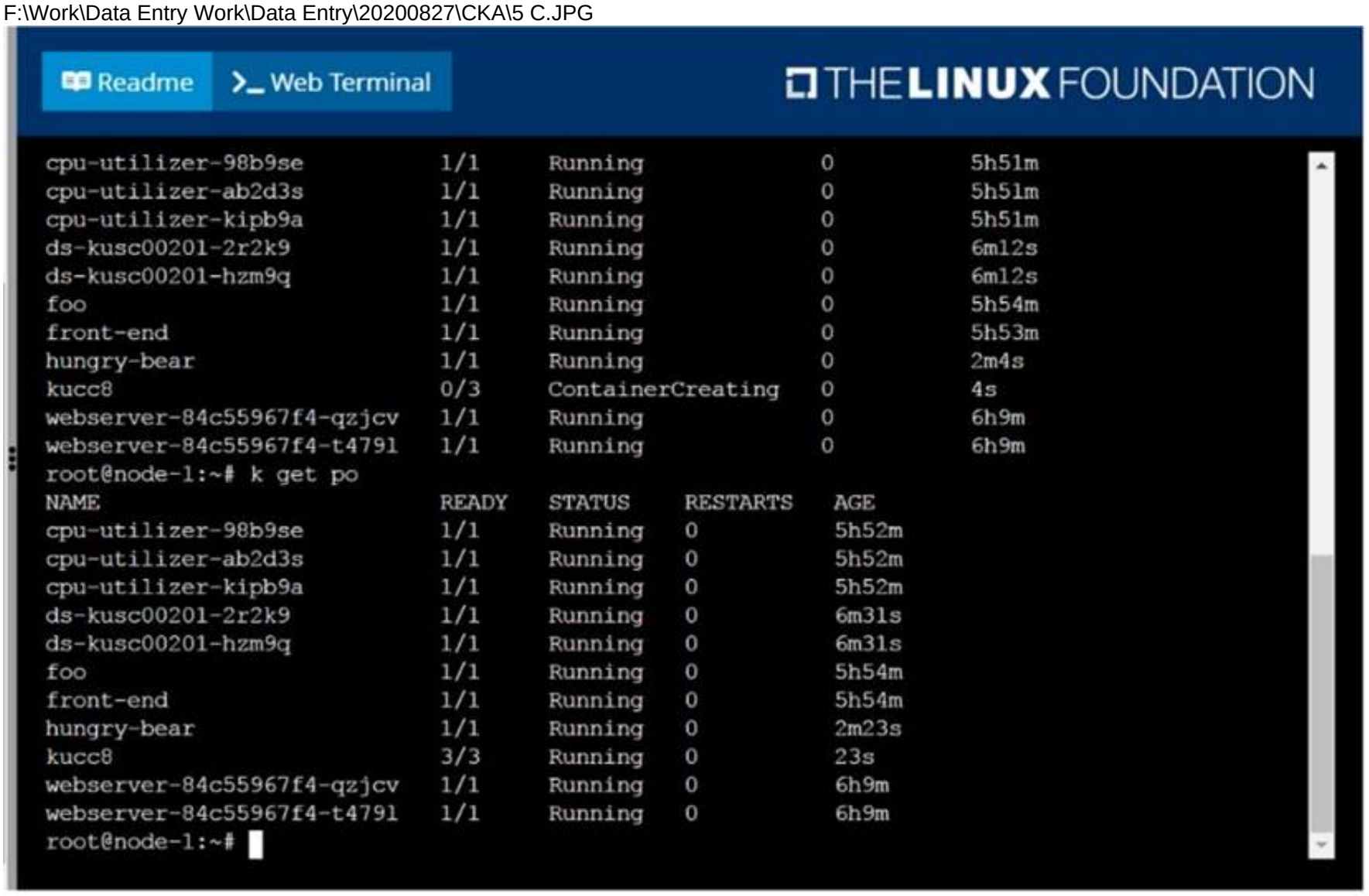
**Explanation:**
solution



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

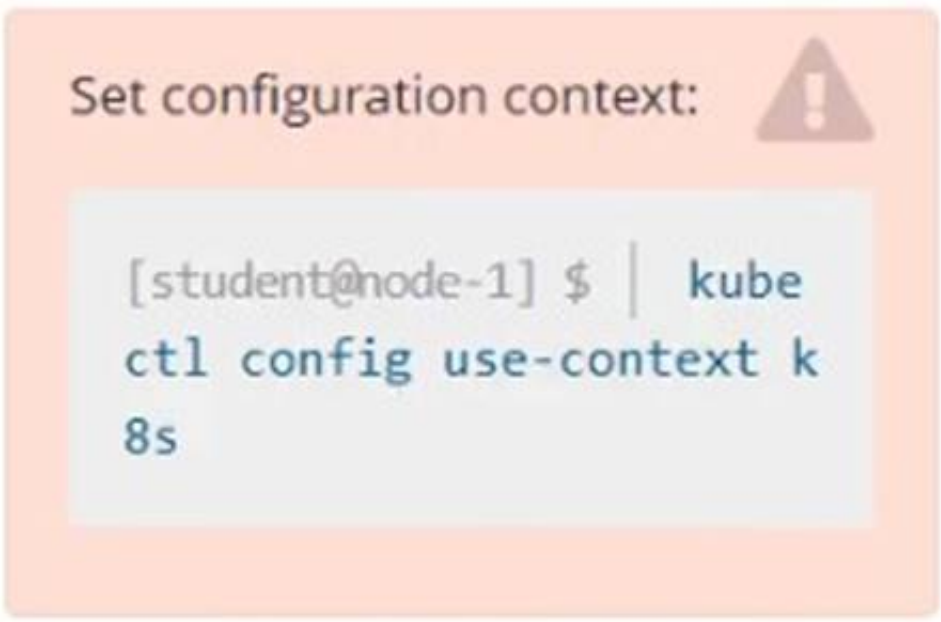



F:\Work\Data Entry Work\Data Entry\20200827\CKA\5 C.JPG




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

**NEW QUESTION 29**
CORRECT TEXT
Score:7%




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

A. Mastered
B. Not Mastered

**Answer:** A

**Explanation:**

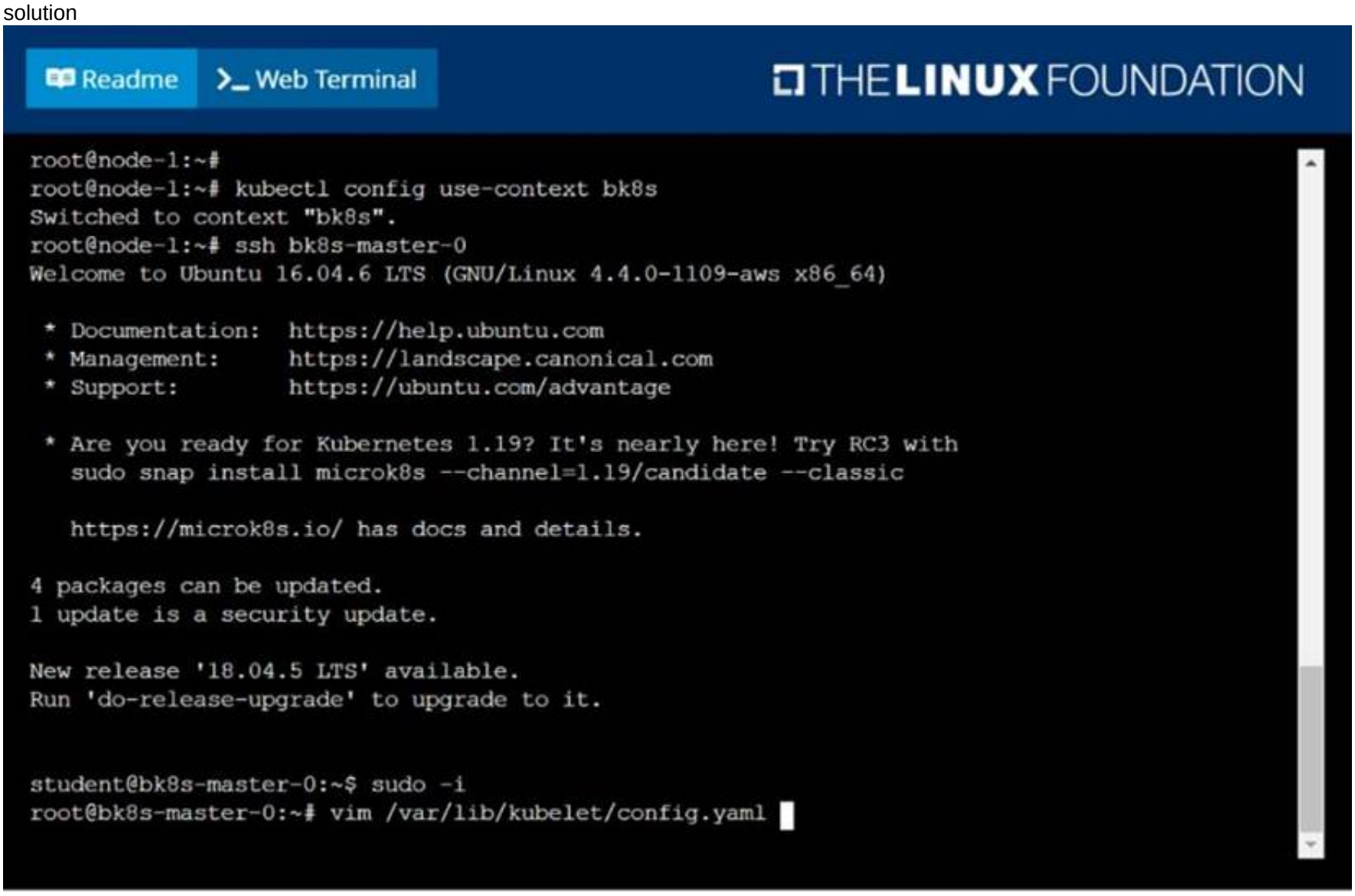
Solution:
#
kubectl get pod big-corp-app -o yaml
#
apiVersion: v1
kind: Pod
metadata:
name: big-corp-app
spec:
containers:
- name: big-corp-app
image: busybox
args:
- /bin/sh
- -c
- > i=0;
while true;
do
echo "$(date) INFO $i" >> /var/log/big-corp-app.log;
i=$((i+1));
sleep 1;
done
volumeMounts:
- name: 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 33**
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

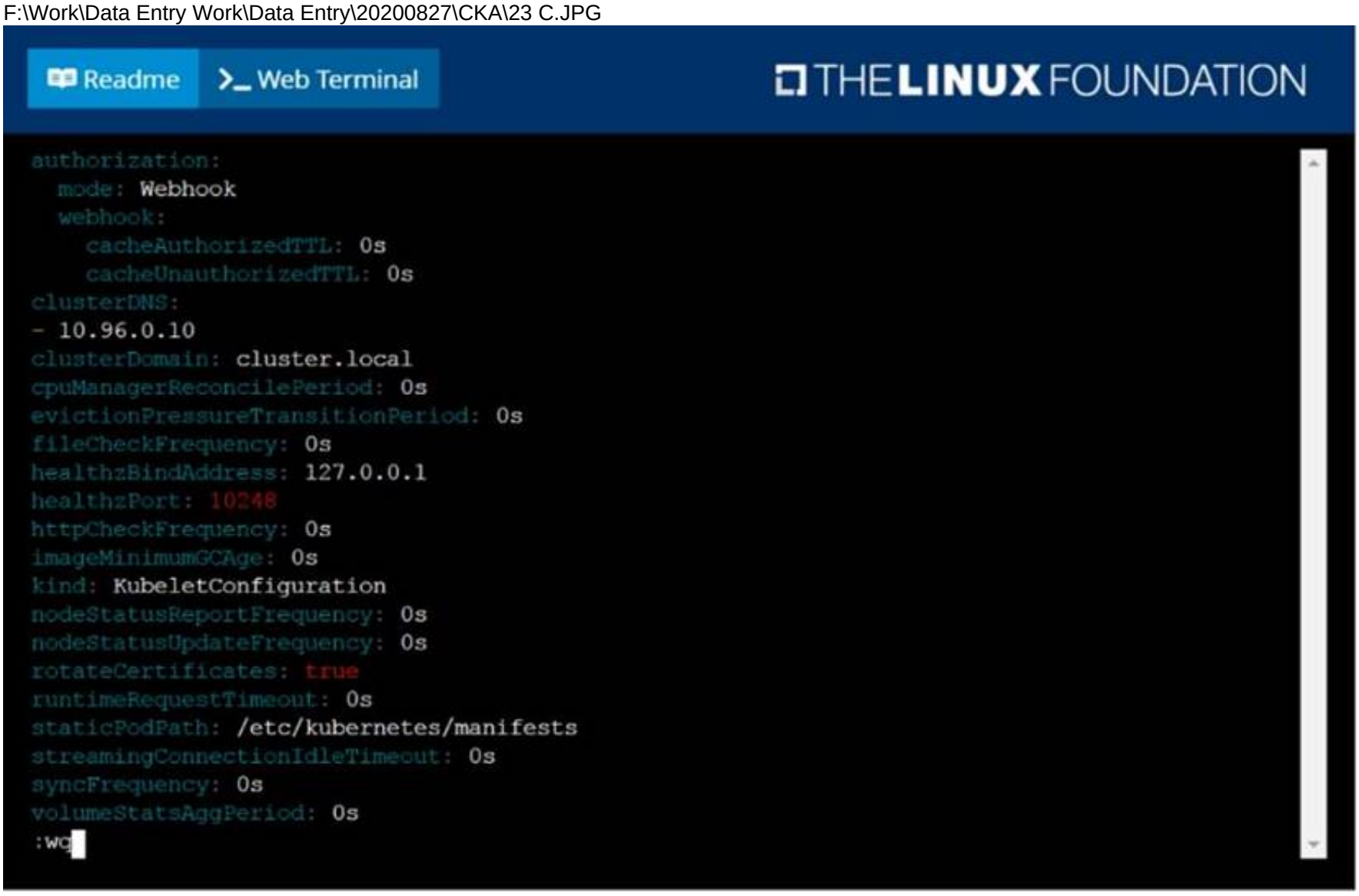
A. Mastered
B. Not Mastered

**Answer:** A

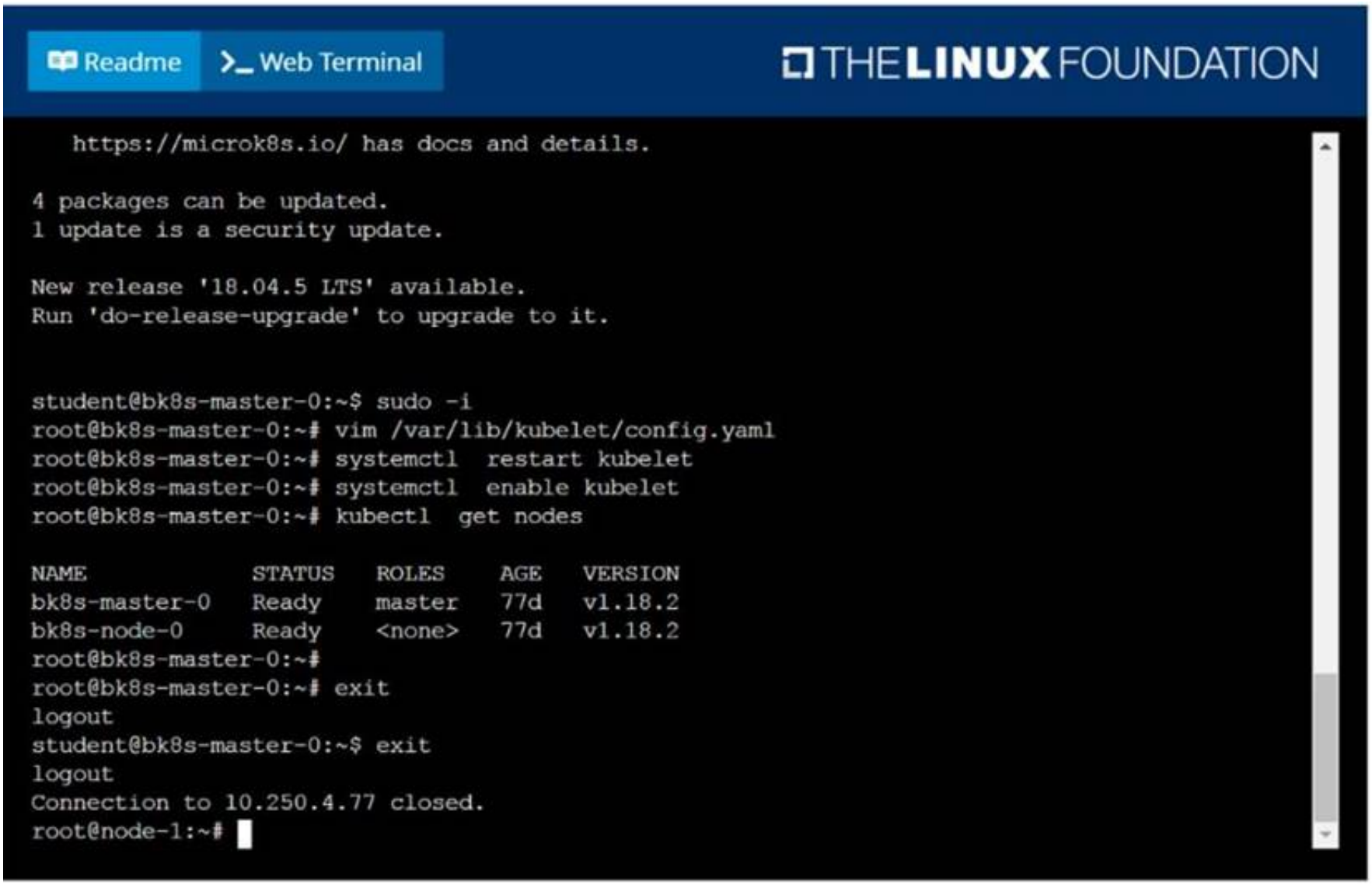
**Explanation:**
solution




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




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




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

**NEW QUESTION 35**
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

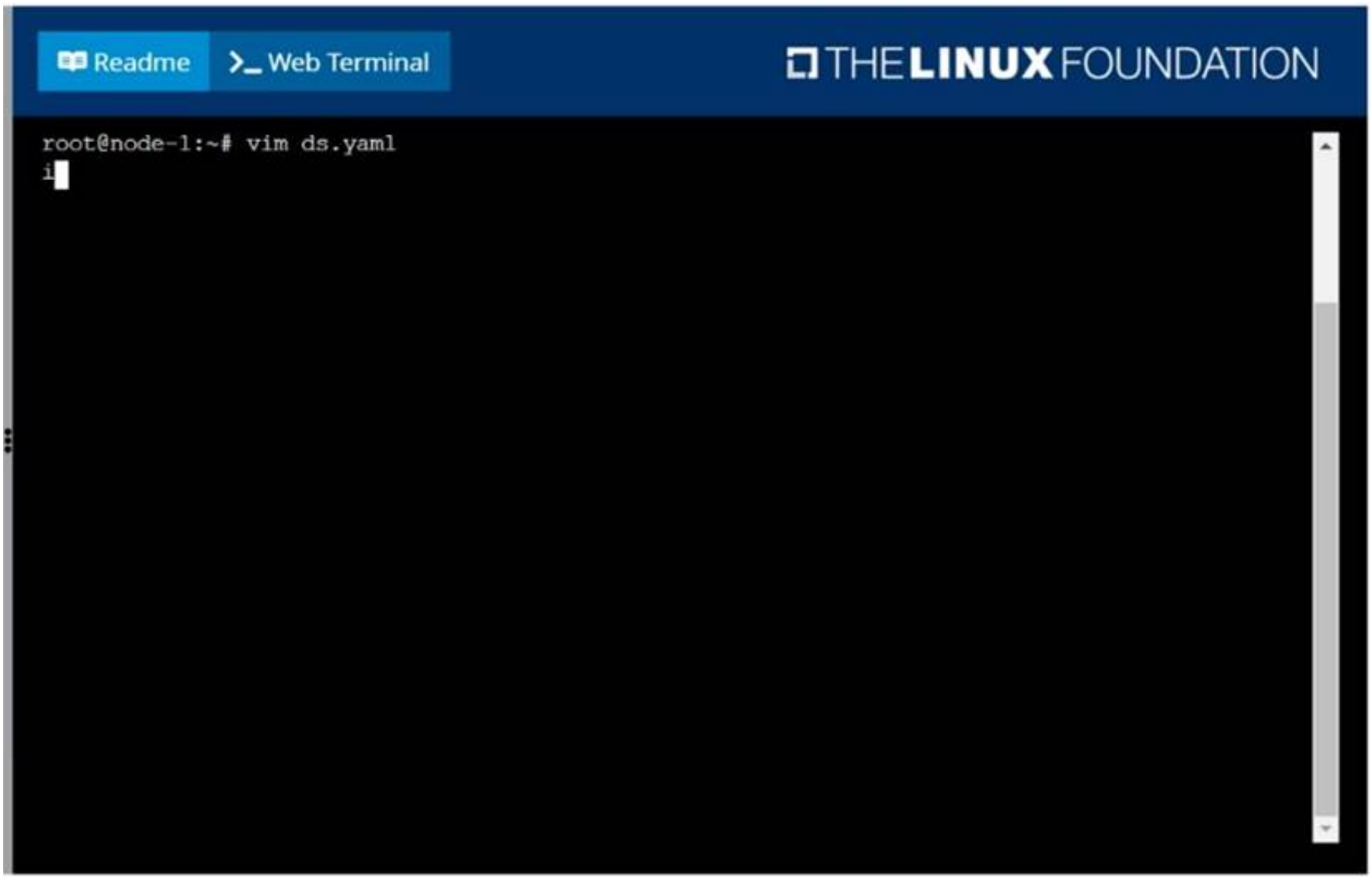
**Explanation:**
kubectl run busybox --image=busybox -it --rm --restart=Never --
/bin/sh -c 'echo hello world'
kubectl get po # You shouldn't see pod with the name "busybox"

**NEW QUESTION 40**
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.

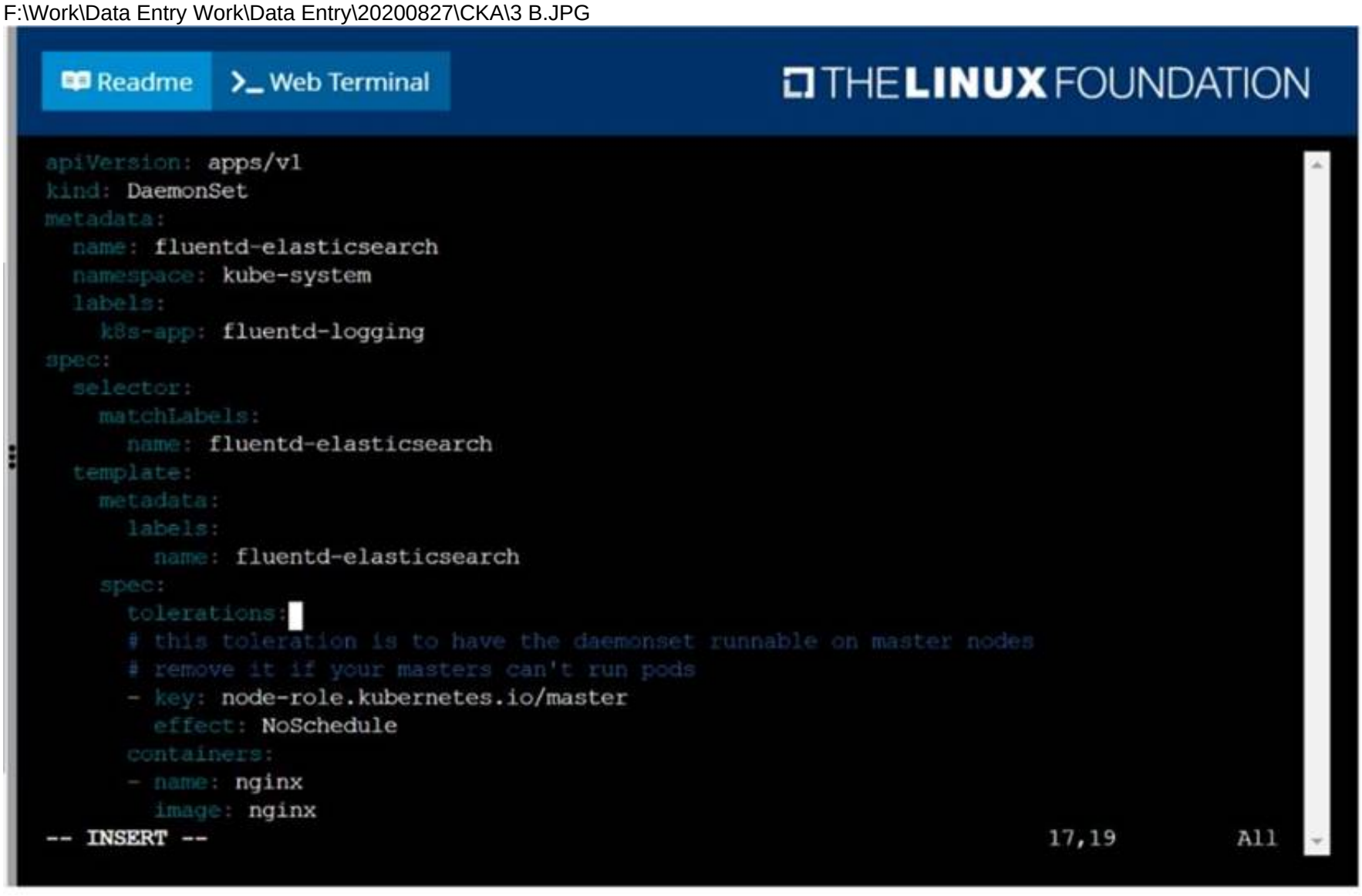
A. Mastered
B. Not Mastered

**Answer:** A

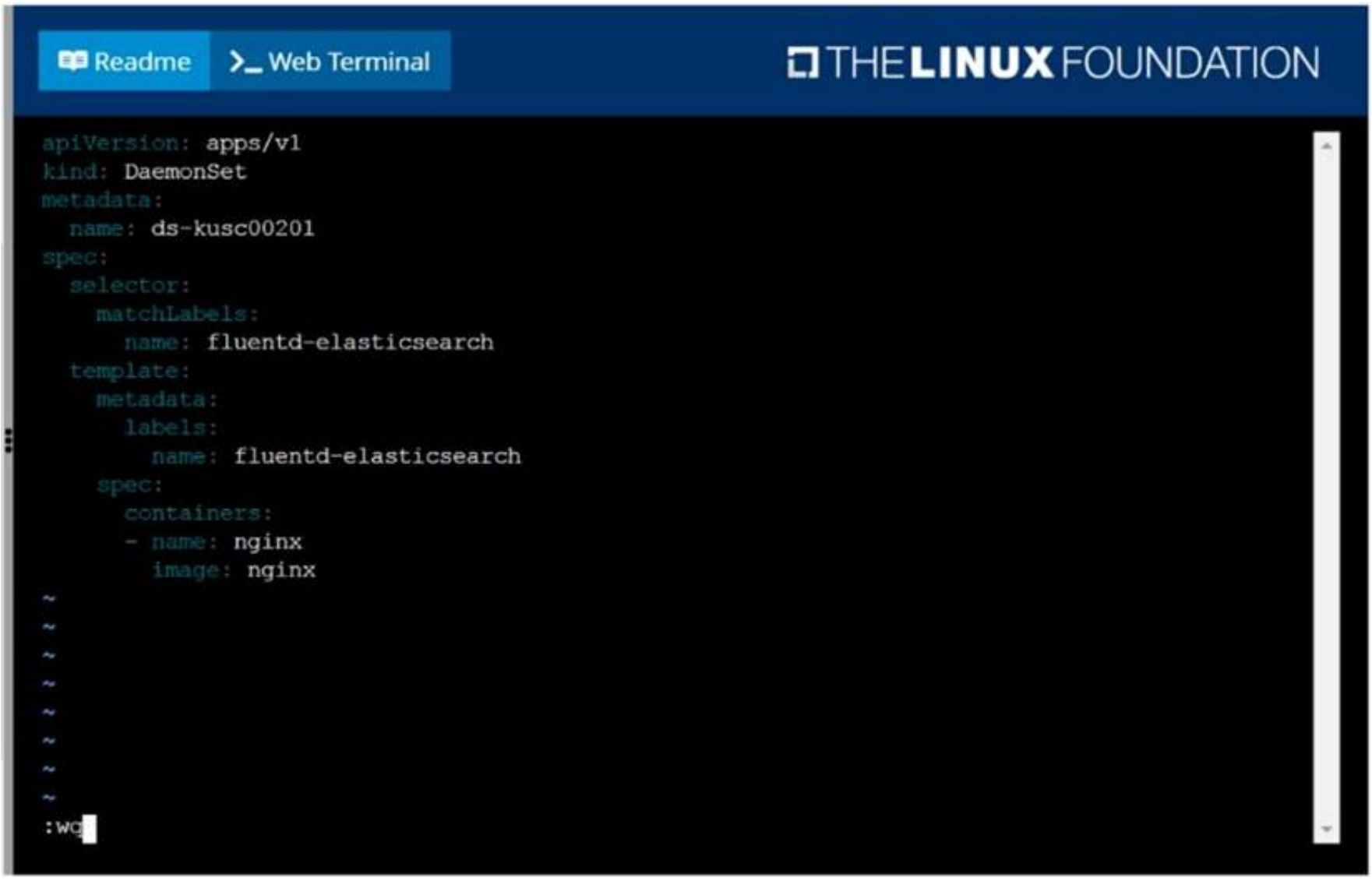
**Explanation:**
solution



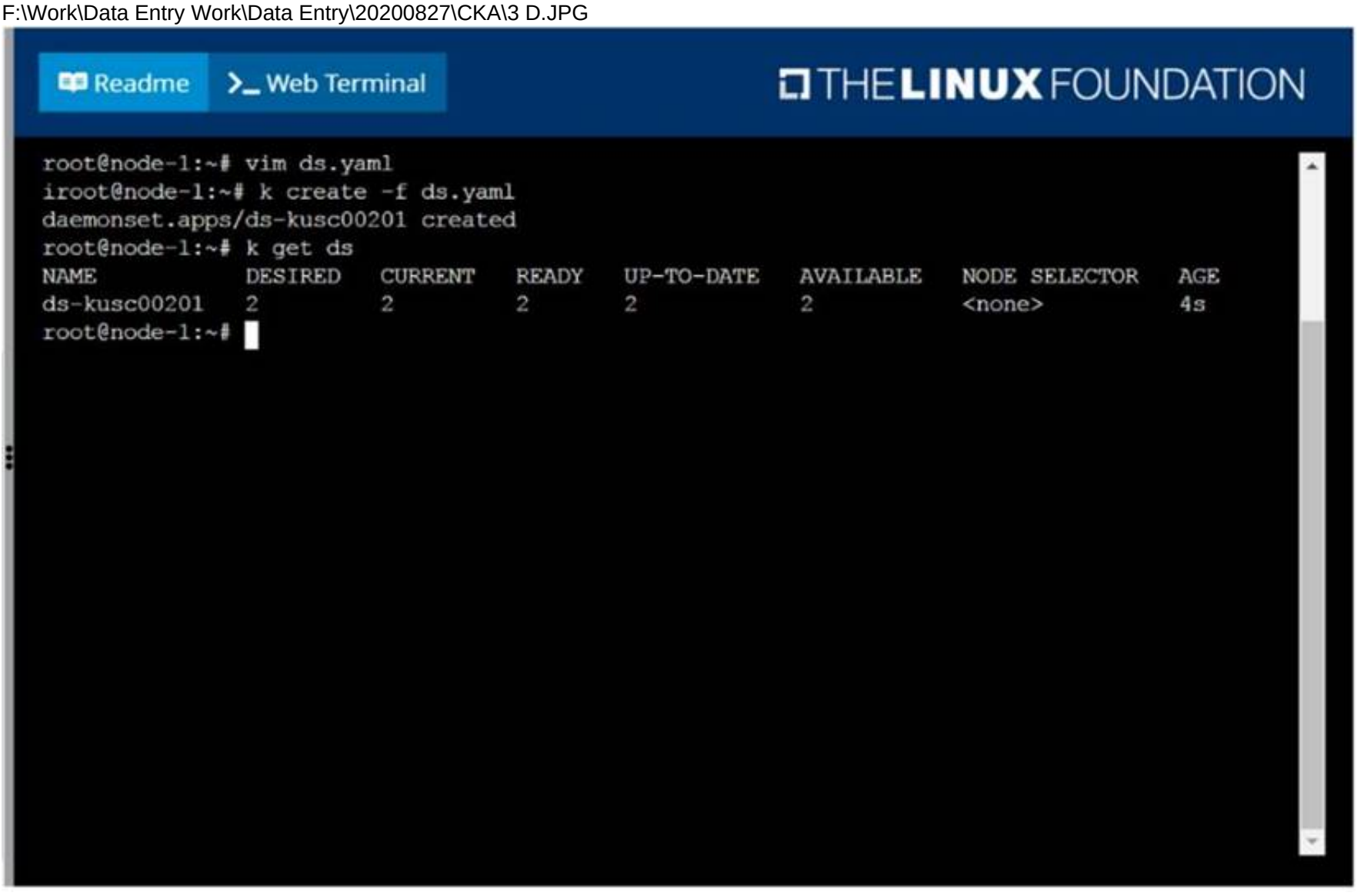

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 B.JPG




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 45**
CORRECT TEXT
Get list of all the pods showing name and namespace with a jsonpath expression.

A. Mastered
B. Not Mastered

**Answer:** A

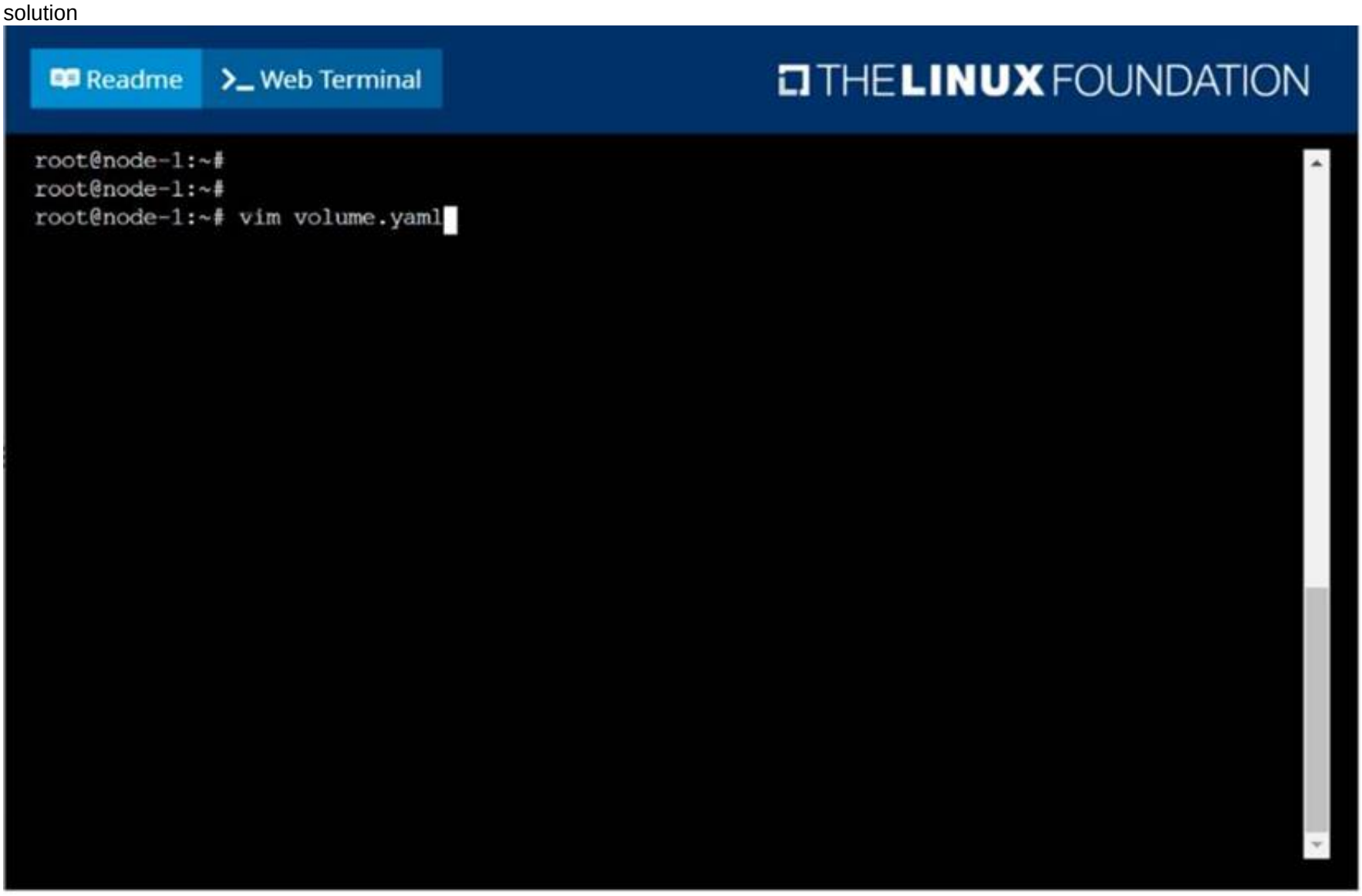
**Explanation:**
kubectl get pods -o=jsonpath="{.items[*]['metadata.name'
, 'metadata.namespace']}"

**NEW QUESTION 46**
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.

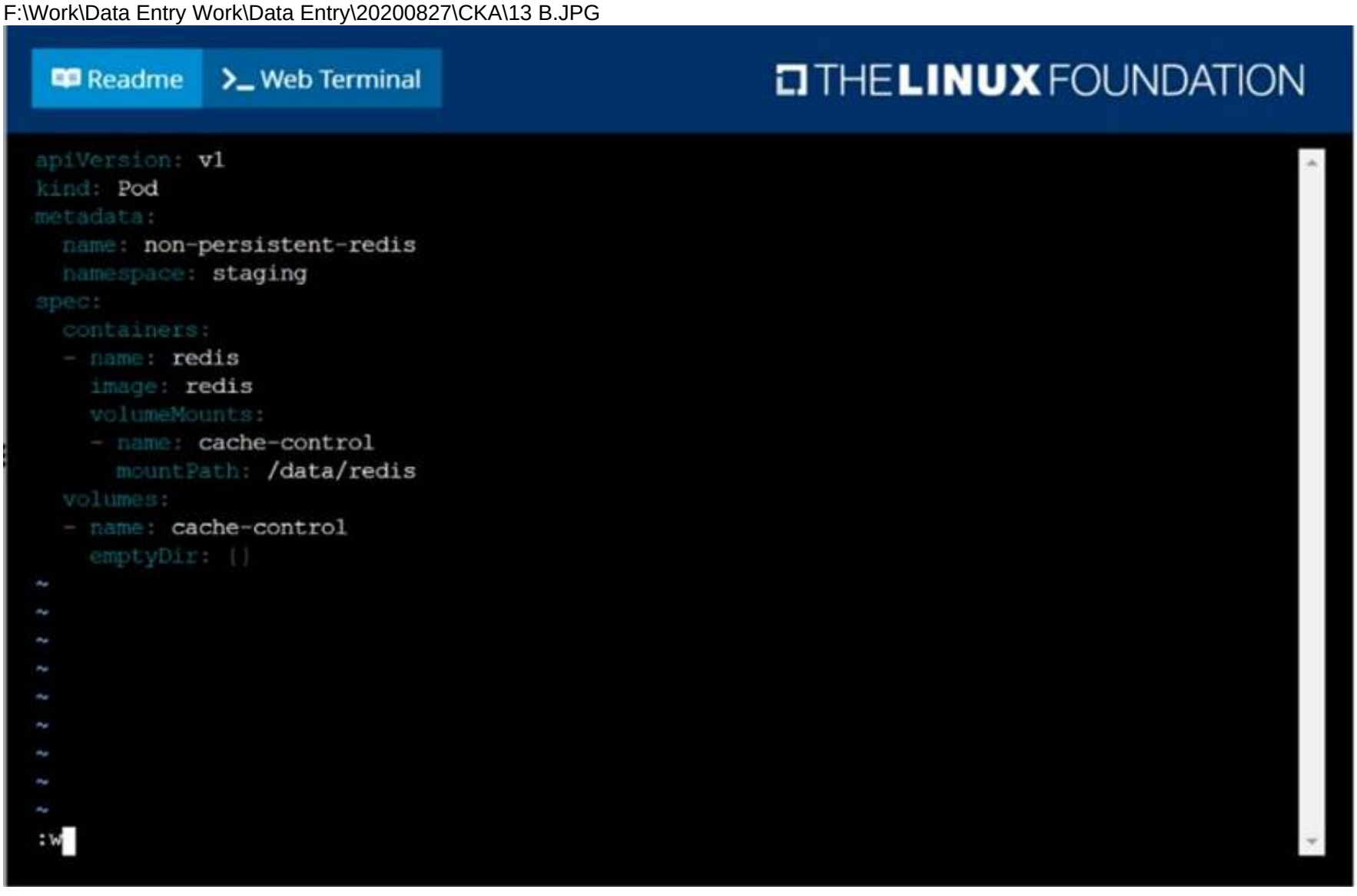
A. Mastered
B. Not Mastered

**Answer:** A

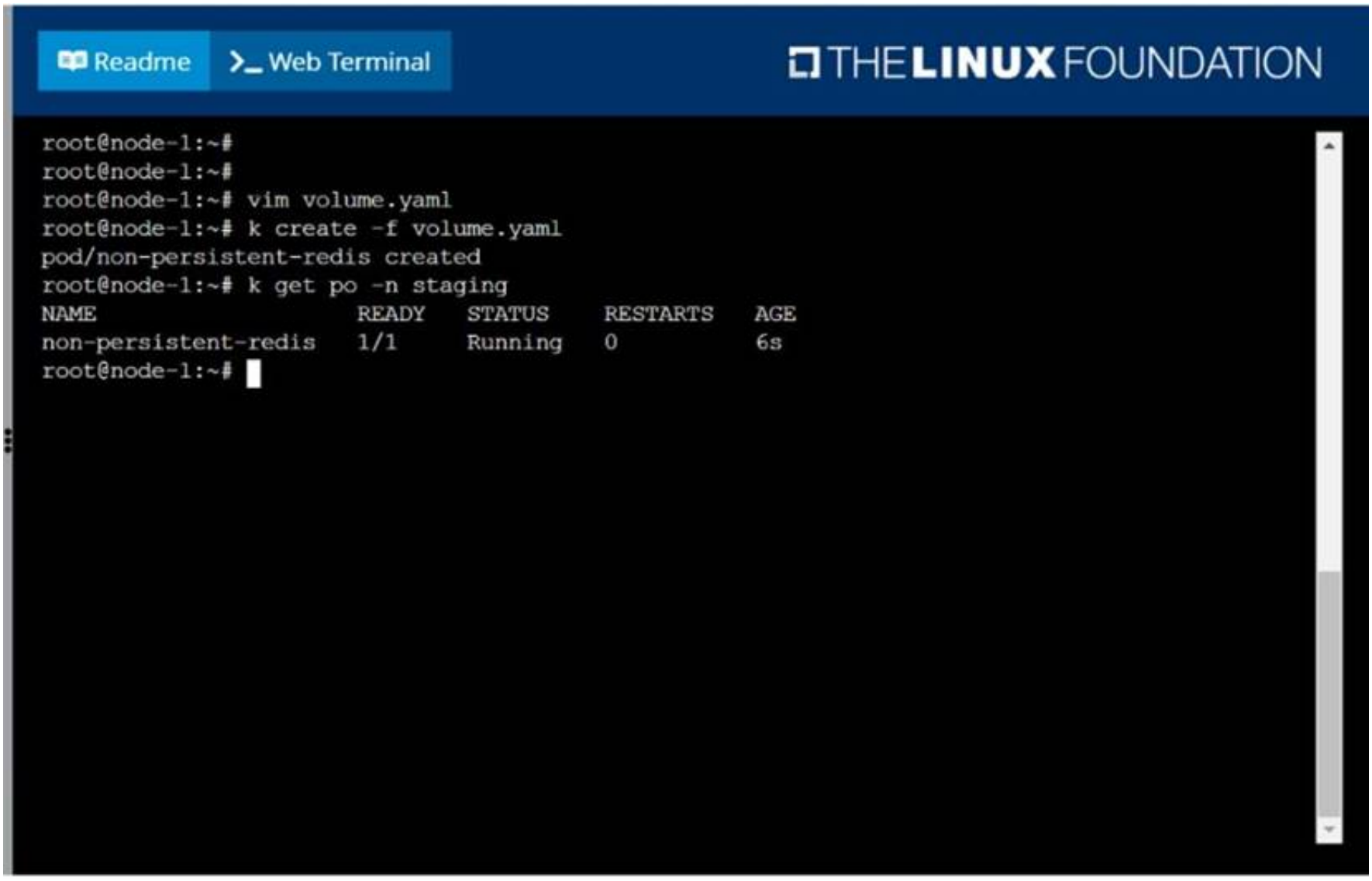
**Explanation:**
solution




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




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




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

**NEW QUESTION 48**
CORRECT TEXT
Score: 4%


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

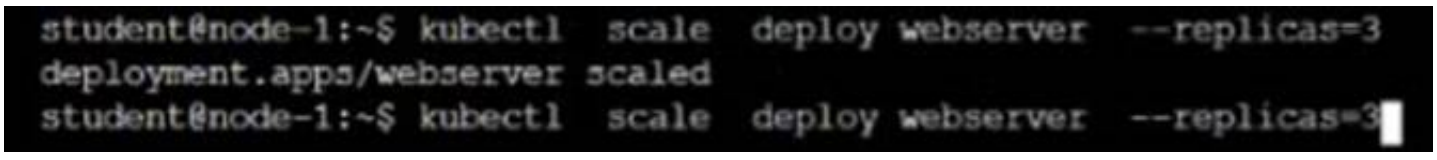
**NEW QUESTION 53**
CORRECT TEXT
Task Weight: 4%


Task
Scale the deployment webserver to 3 pods.

A. Mastered
B. Not Mastered

**Answer:** A

**Explanation:**
Solution:




**NEW QUESTION 58**
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 62**
CORRECT TEXT
List all the pods showing name and namespace with a json path expression

A. Mastered
B. Not Mastered

**Answer:** A

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

**NEW QUESTION 66**
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 70**

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"

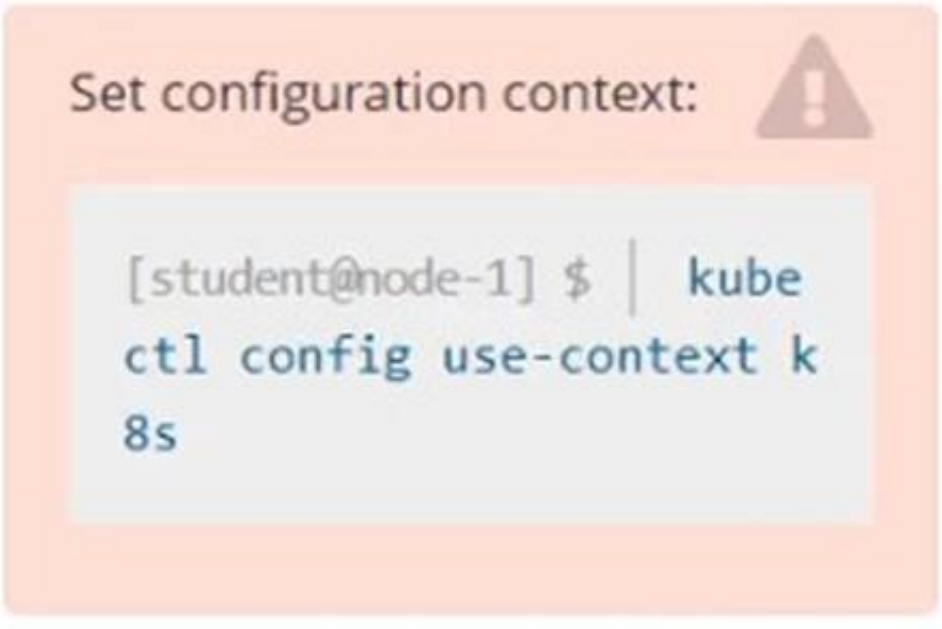
**NEW QUESTION 73**
CORRECT TEXT
Create a busybox pod that runs the command “env” and save the output to “envpod” file

A. Mastered
B. Not Mastered

**Answer:** A

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

**NEW QUESTION 75**
CORRECT TEXT
Score: 5%




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 76**
CORRECT TEXT
Score: 4%


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 81**
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 83**
.....

# 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