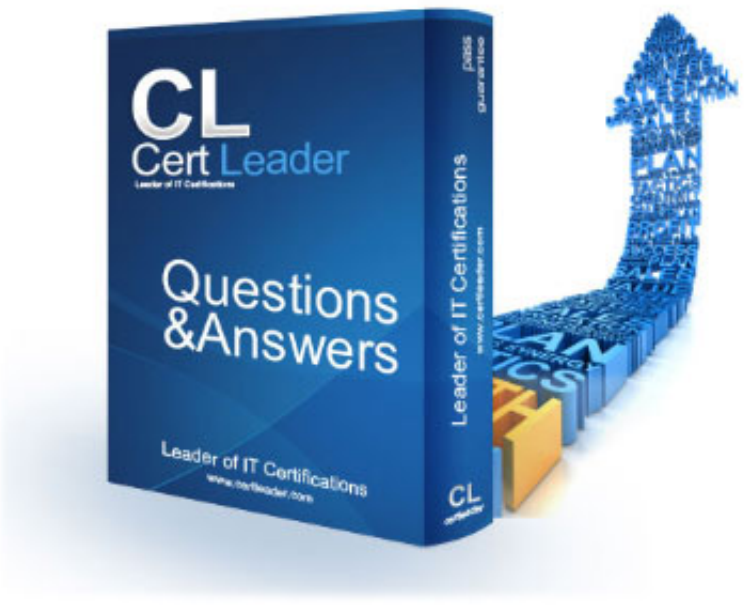
CKA Dumps

Certified Kubernetes Administrator (CKA) Program

https://www.certleader.com/CKA-dumps.html



NEW QUESTION 1
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 2
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

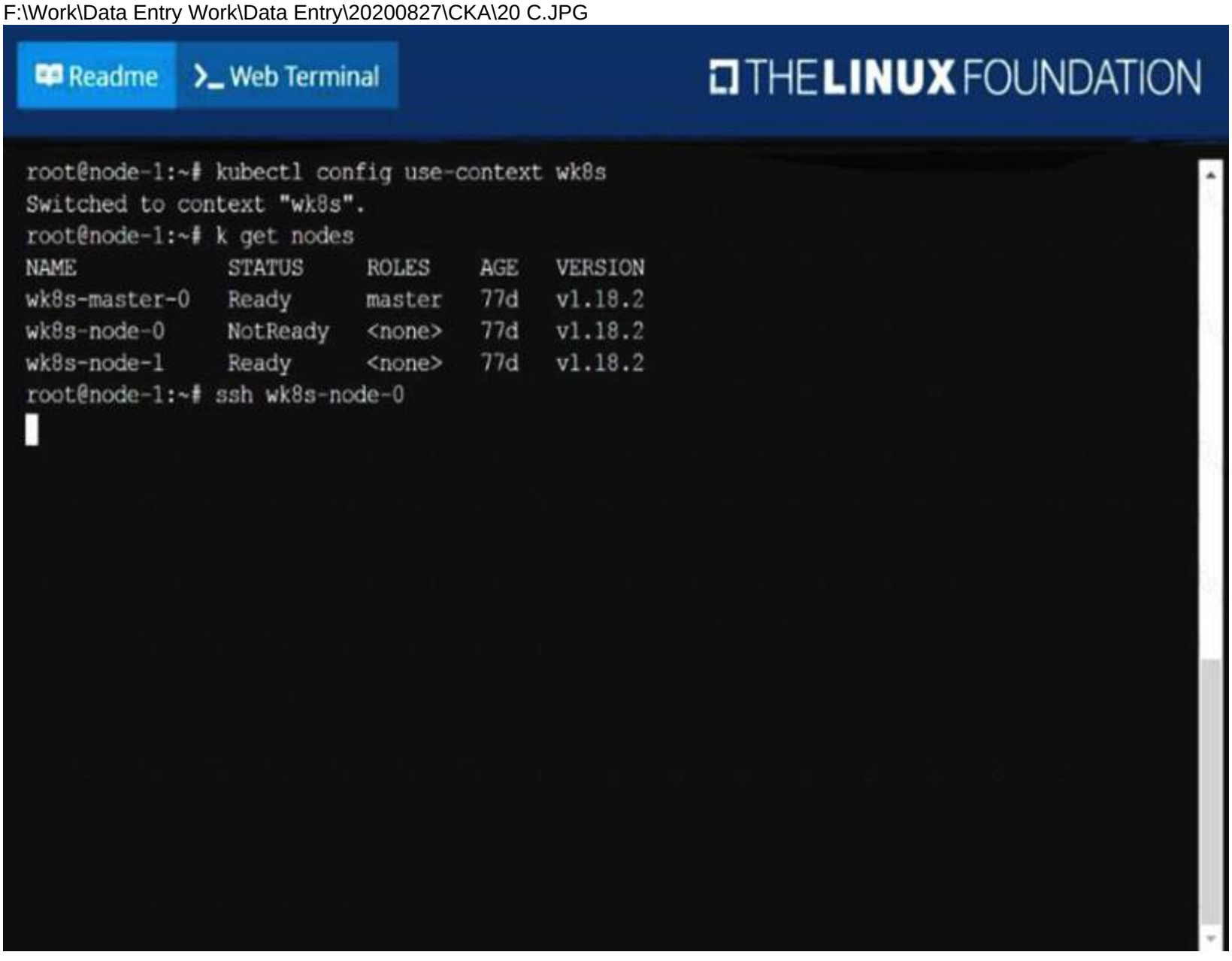
NEW QUESTION 3
A Kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a Ready state, ensuring that any changes are made permanent.
You can ssh to the failed node using:
[student@node-1] $ | ssh Wk8s-node-0
You can assume elevated privileges on the node with the following command:

[student@w8ks-node-0] $ | sudo –i

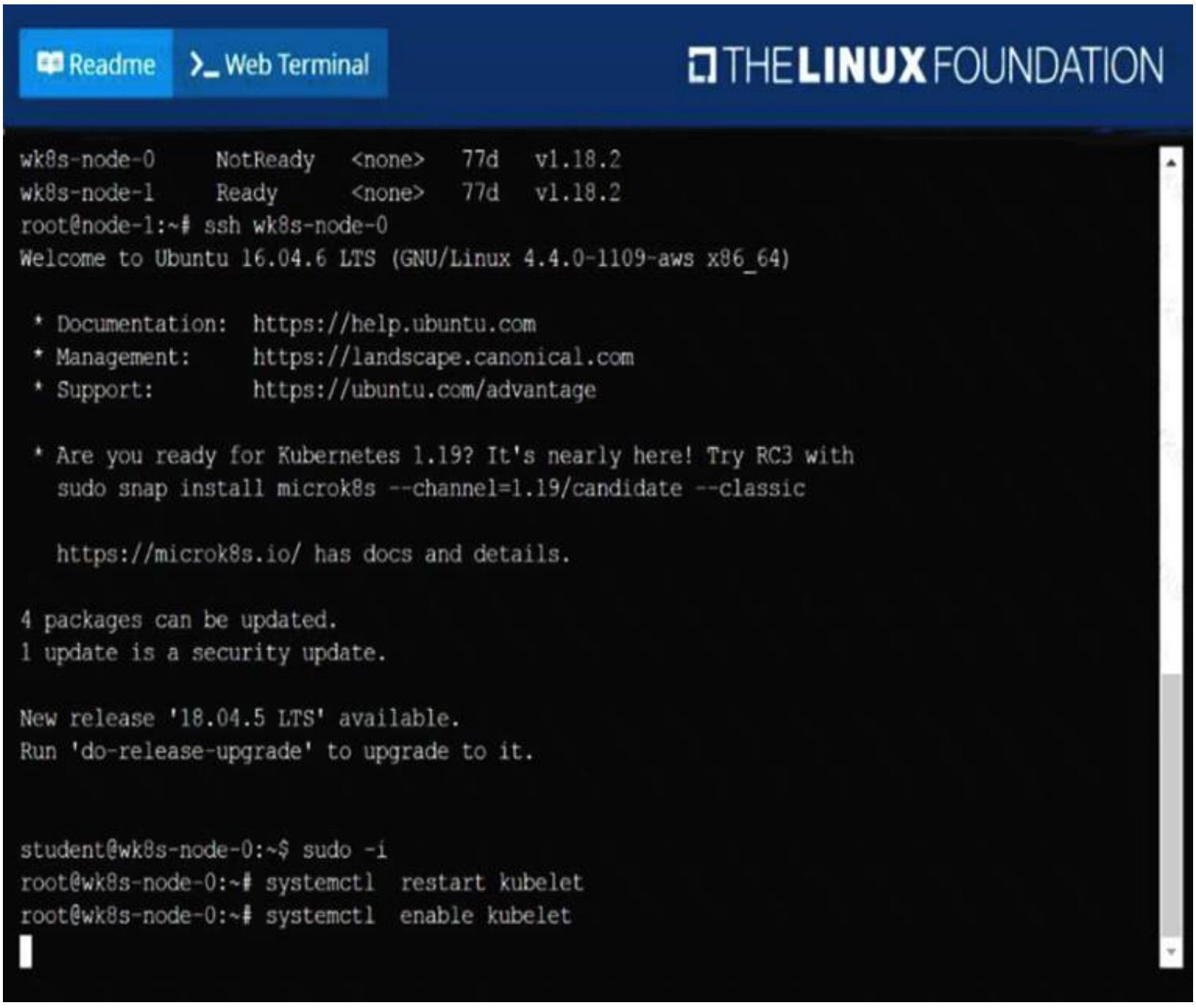
A. Mastered
B. Not Mastered

Answer: A

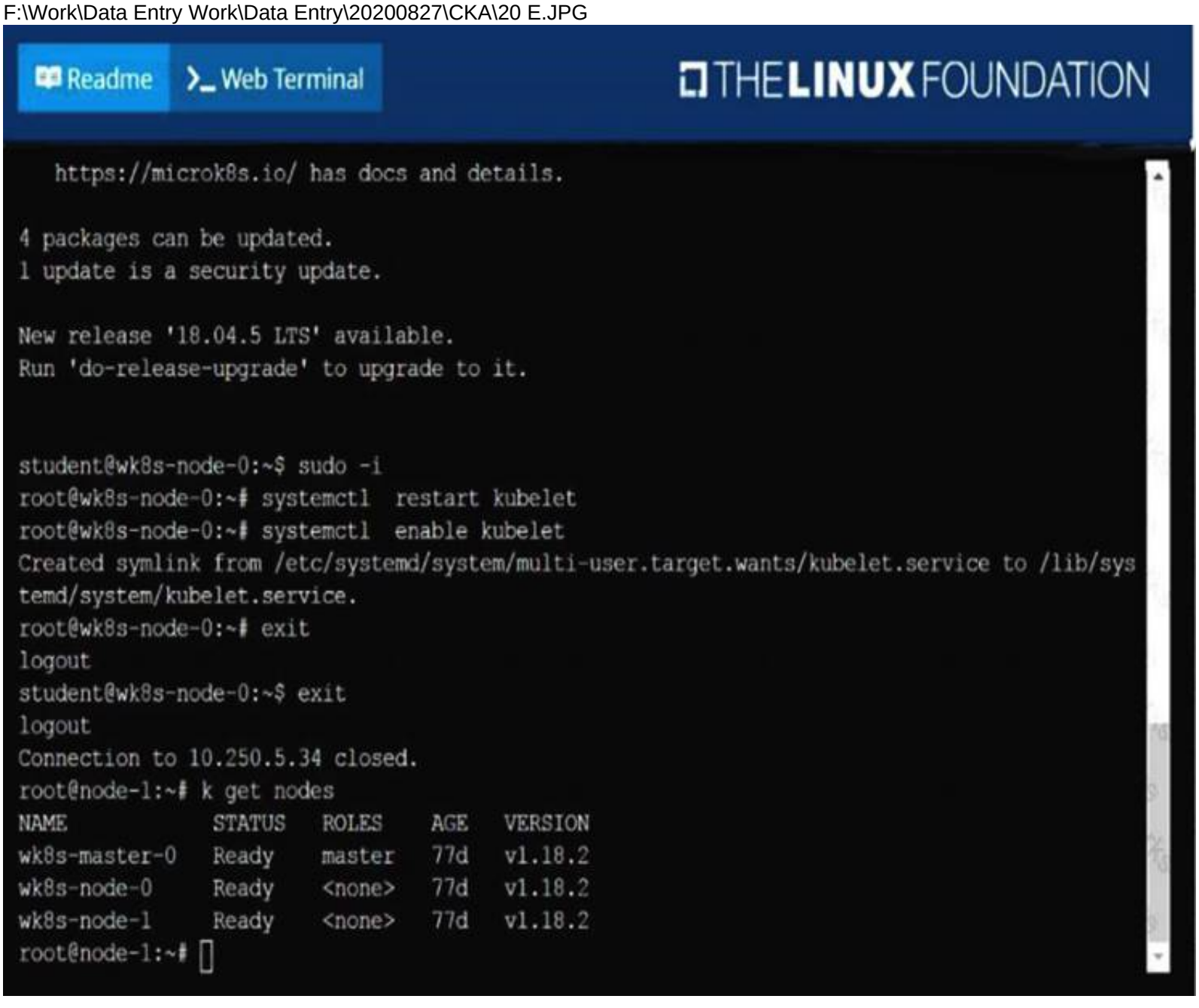
Explanation:
solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 C.JPG



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



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



NEW QUESTION 4
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 5
Create a pod that having 3 containers in it? (Multi-Container)

A. Mastered
B. Not Mastered

Answer: A

Explanation:
image=nginx, image=redis, image=consul Name nginx container as “nginx-container” Name redis container as “redis-container” Name consul container as “consul-container”
Create a pod manifest file for a container and append container section for rest of the images
kubectl run multi-container --generator=run-pod/v1 --image=nginx -- dry-run -o yaml > multi-container.yaml
then
vim multi-container.yaml apiVersion: v1
kind: Pod metadata: labels:
run: multi-container name: multi-container spec:
containers:
- image: nginx
name: nginx-container
- image: redis
name: redis-container
- image: consul
name: consul-container restartPolicy: Always

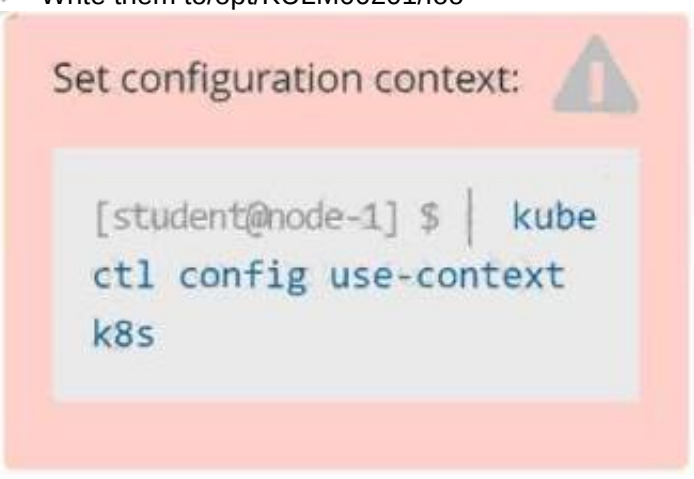
NEW QUESTION 6
List all the pods sorted by name

A. Mastered
B. Not Mastered

Answer: A

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

NEW QUESTION 7
Monitor the logs of pod foo and:
- Extract log lines corresponding to error unable-to-access-website
- Write them to/opt/KULM00201/foo

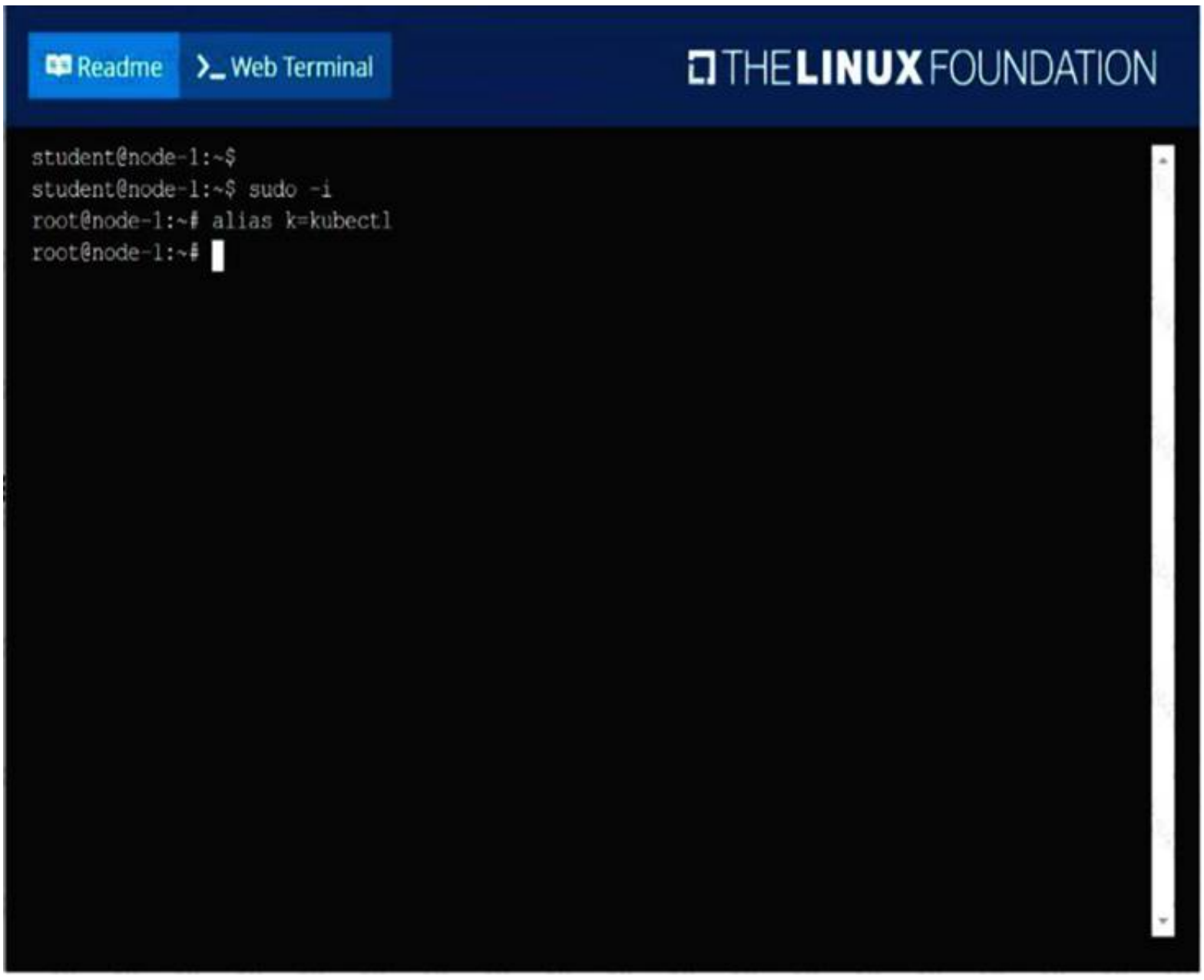


A. Mastered
B. Not Mastered

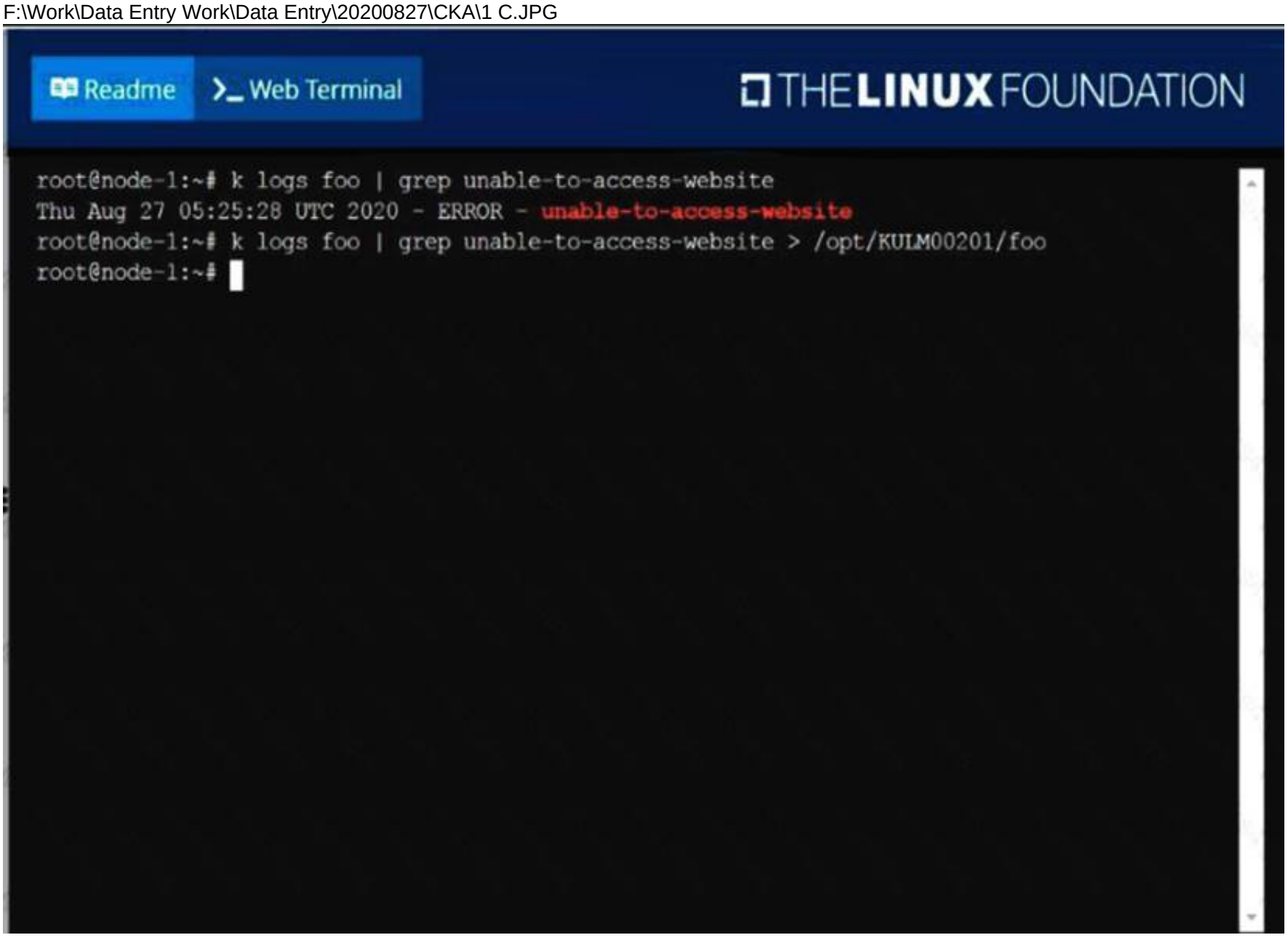
Answer: A

Explanation:

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



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



NEW QUESTION 8
Create a pod as follows:

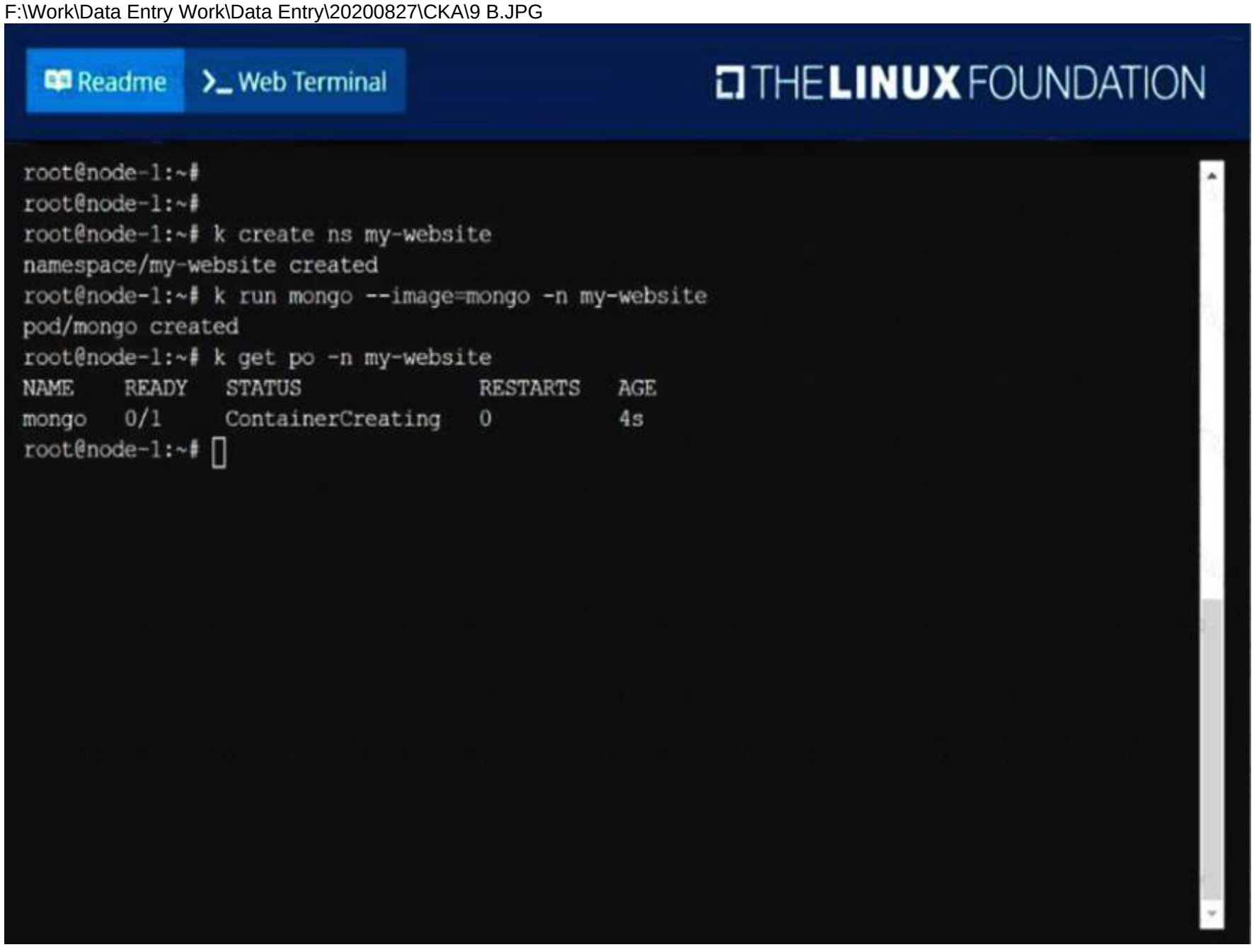
- Name: mongo
- Using Image: mongo
- In a new Kubernetes namespace named: my-website

A. Mastered
B. Not Mastered

Answer: A

Explanation:

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



NEW QUESTION 9
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 10
Create an nginx pod and list the pod with different levels of verbosity

A. Mastered
B. Not Mastered

Answer: A

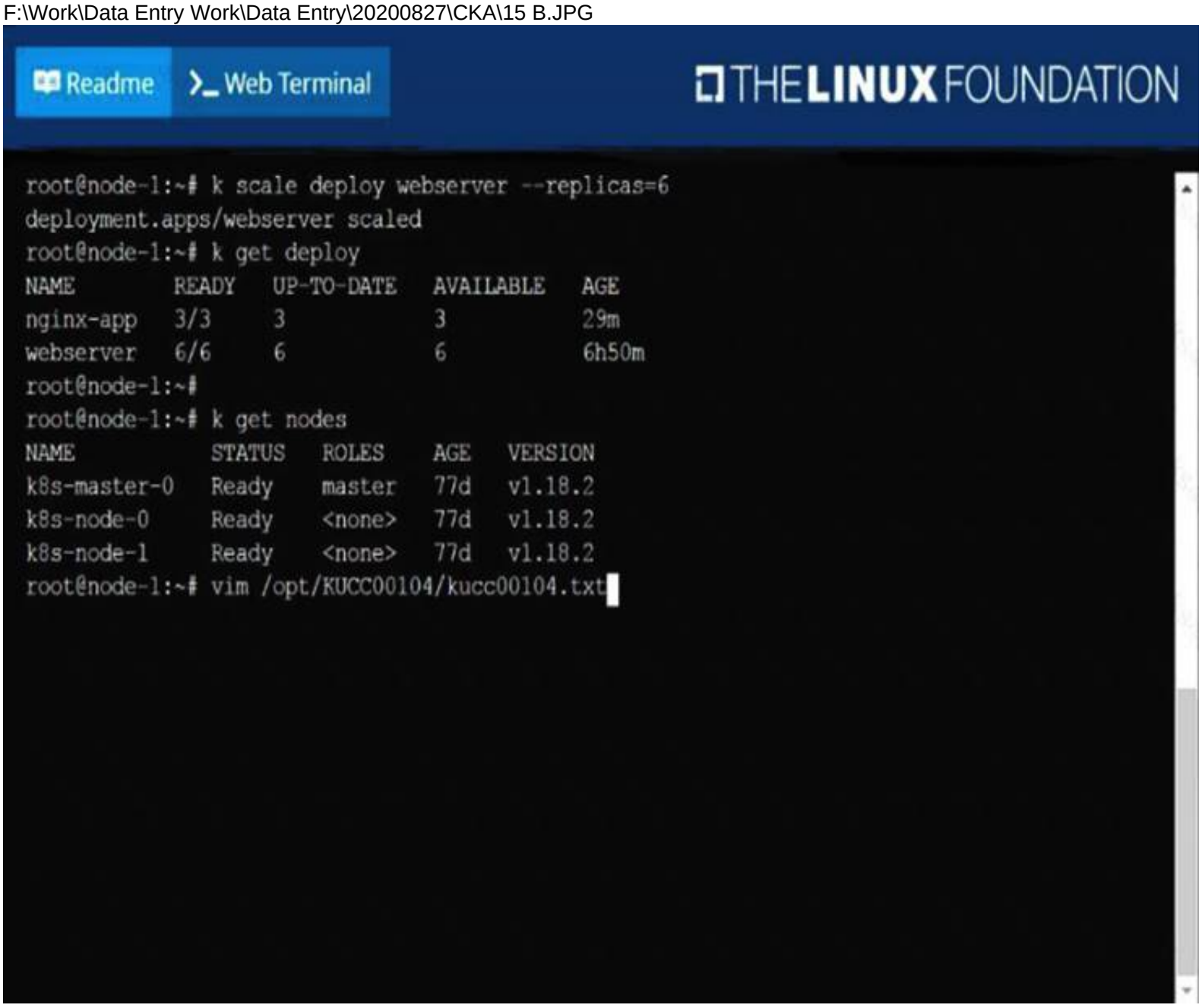
Explanation:
// create a pod
kubectl run nginx --image=nginx --restart=Never --port=80
// List the pod with different verbosity kubectl get po nginx --v=7
kubectl get po nginx --v=8 kubectl get po nginx --v=9

NEW QUESTION 10
Check to see how many worker nodes are ready (not including nodes tainted NoSchedule) and write the number to /opt/KUCC00104/kucc00104.txt.

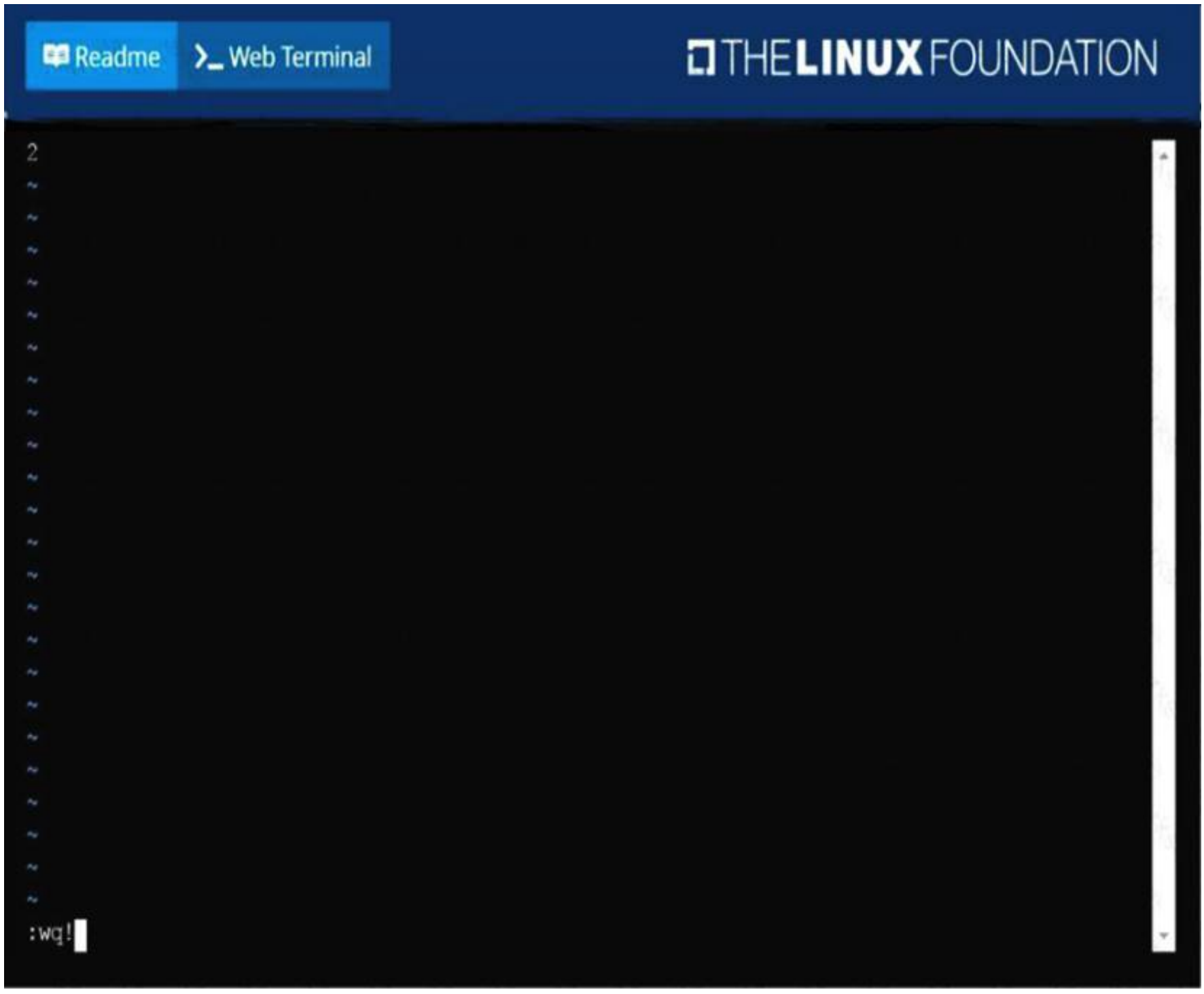
A. Mastered
B. Not Mastered

Answer: A

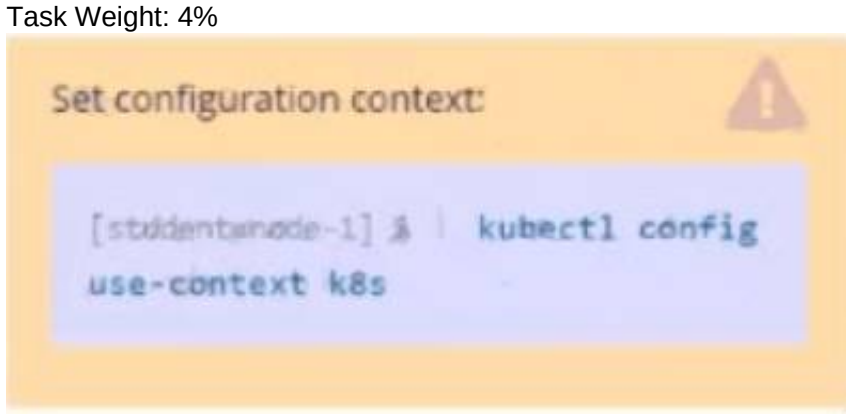
Explanation:
solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\15 B.JPG



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



NEW QUESTION 13
Task Weight: 4%

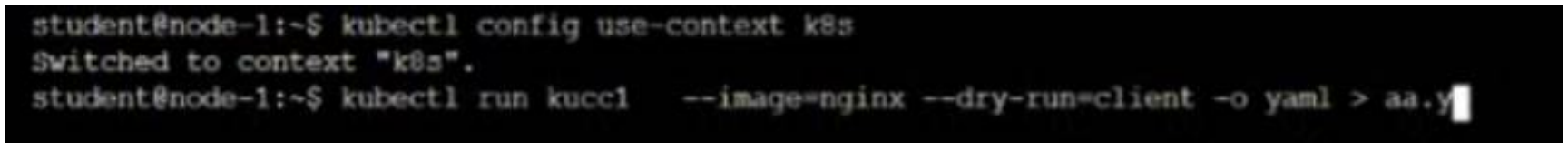


Task
Schedule a Pod as follows:
• Name: kucc1
• App Containers: 2
• Container Name/Images: o nginx
o consul

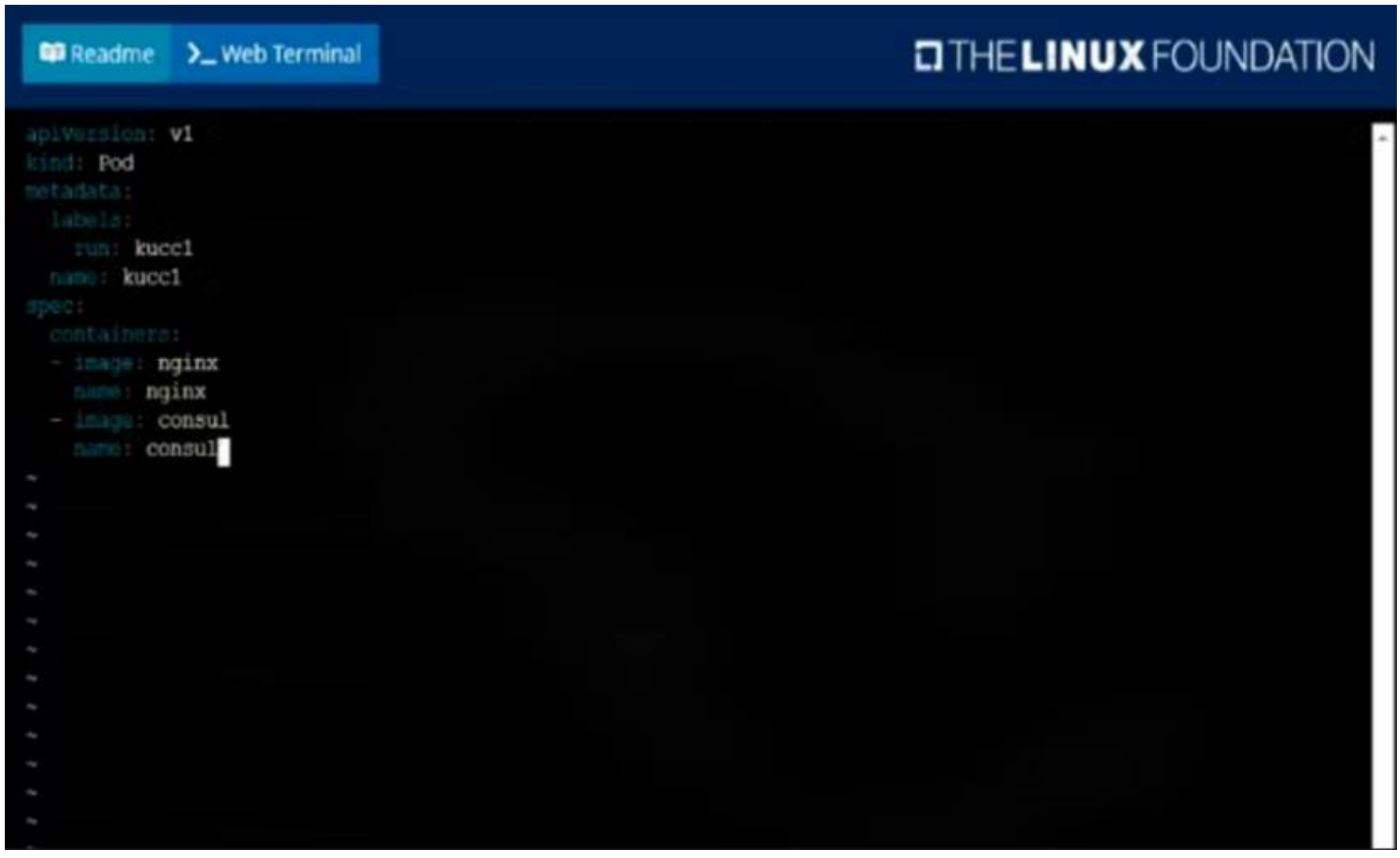
A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:



Graphical user interface, text, application Description automatically generated



Text 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
11-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:~$
```

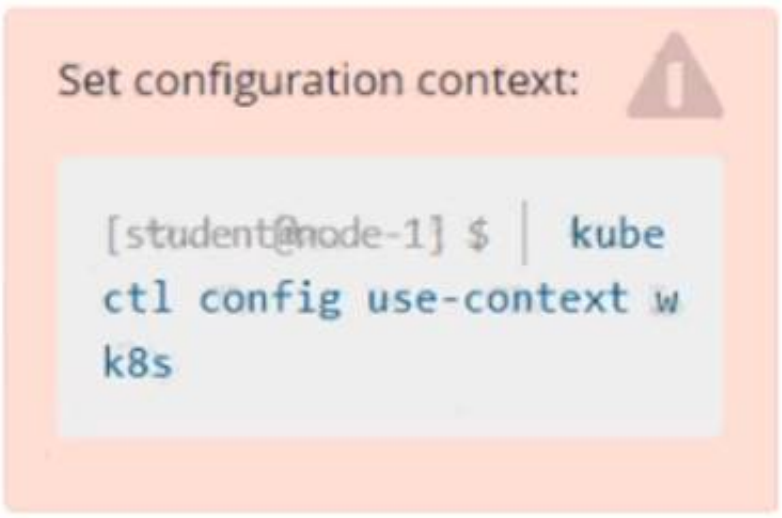
NEW QUESTION 17
Get list of all the pods showing name and namespace with a jsonpath expression.

A. Mastered
B. Not Mastered

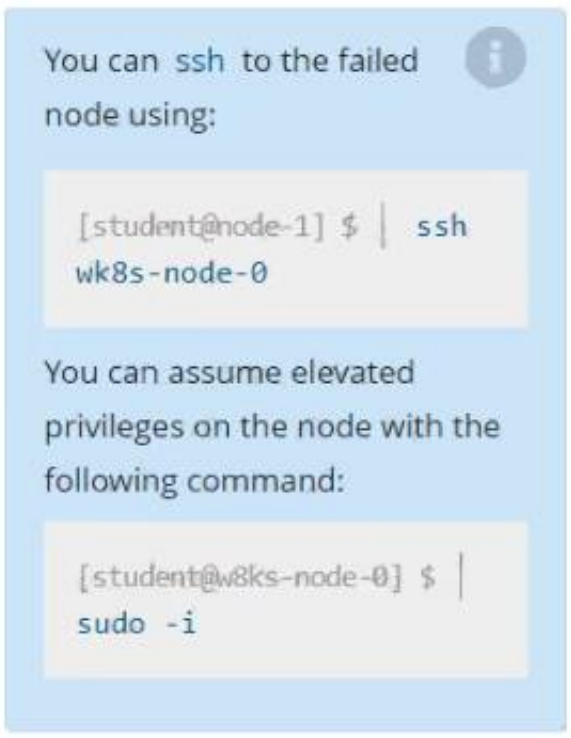
Answer: A

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

NEW QUESTION 20
Score: 13%



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



A. Mastered
B. Not Mastered

Answer: A

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

NEW QUESTION 24
List all the pods sorted by name

A. Mastered
B. Not Mastered

Answer: A

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

NEW QUESTION 26
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 27
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

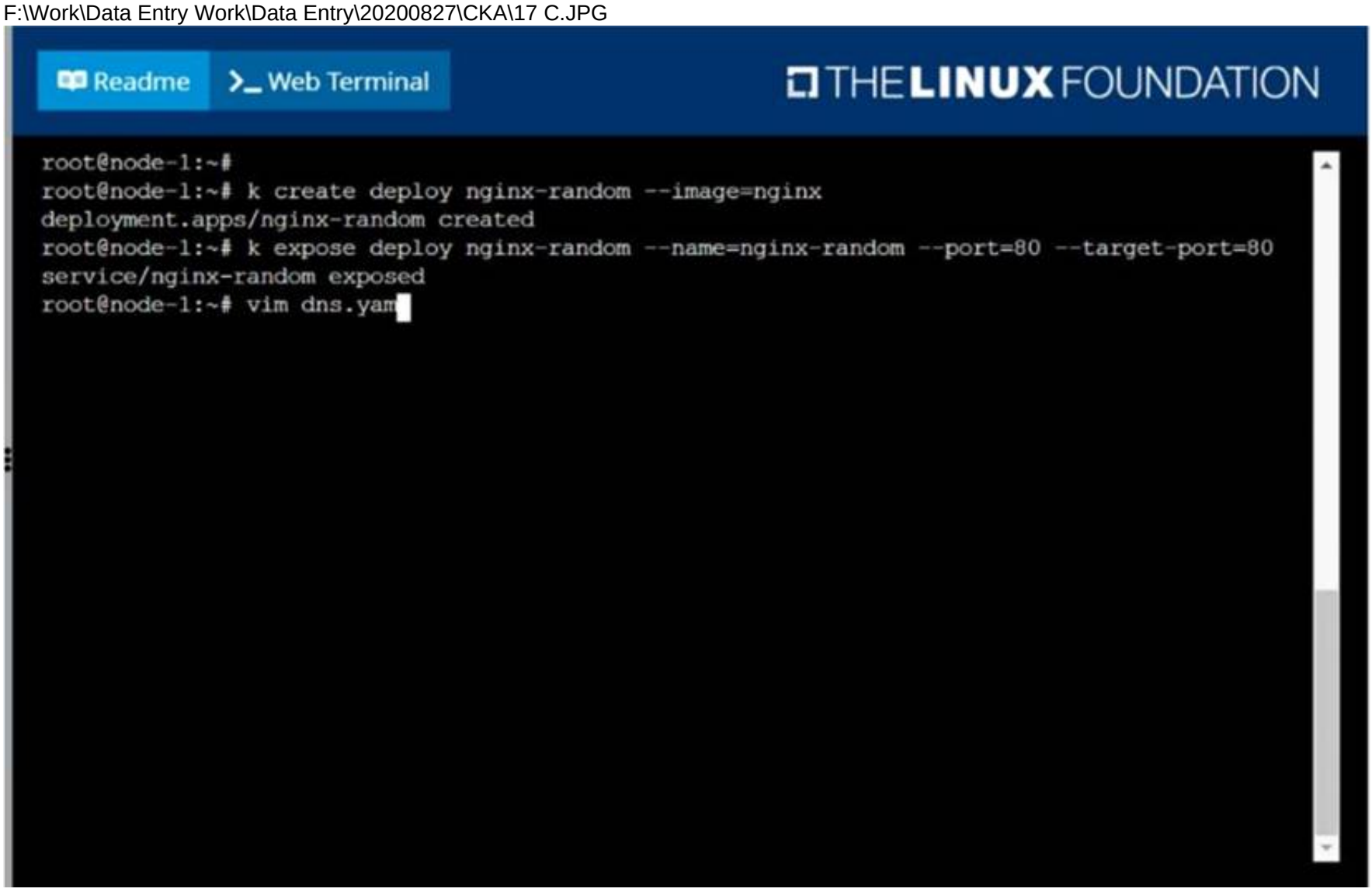
The container(s) within any pod(s) running as a part of this deployment should use the nginx Image
Next, use the utility nslookup to look up the DNS records of the service & pod and write the output to /opt/KUNW00601/service.dns and /opt/KUNW00601/pod.dns respectively.

A. Mastered
B. Not Mastered

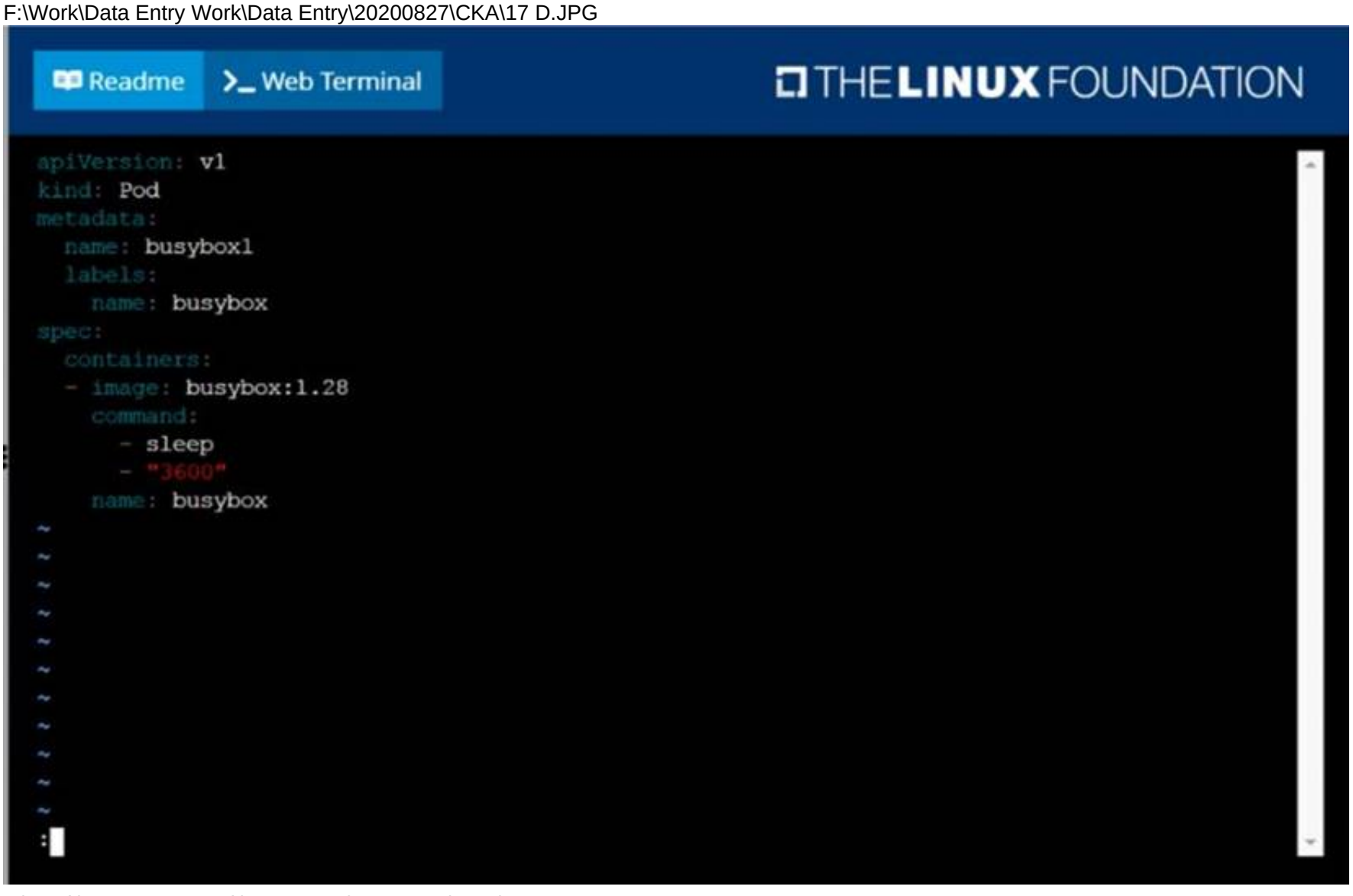
Answer: A

Explanation:

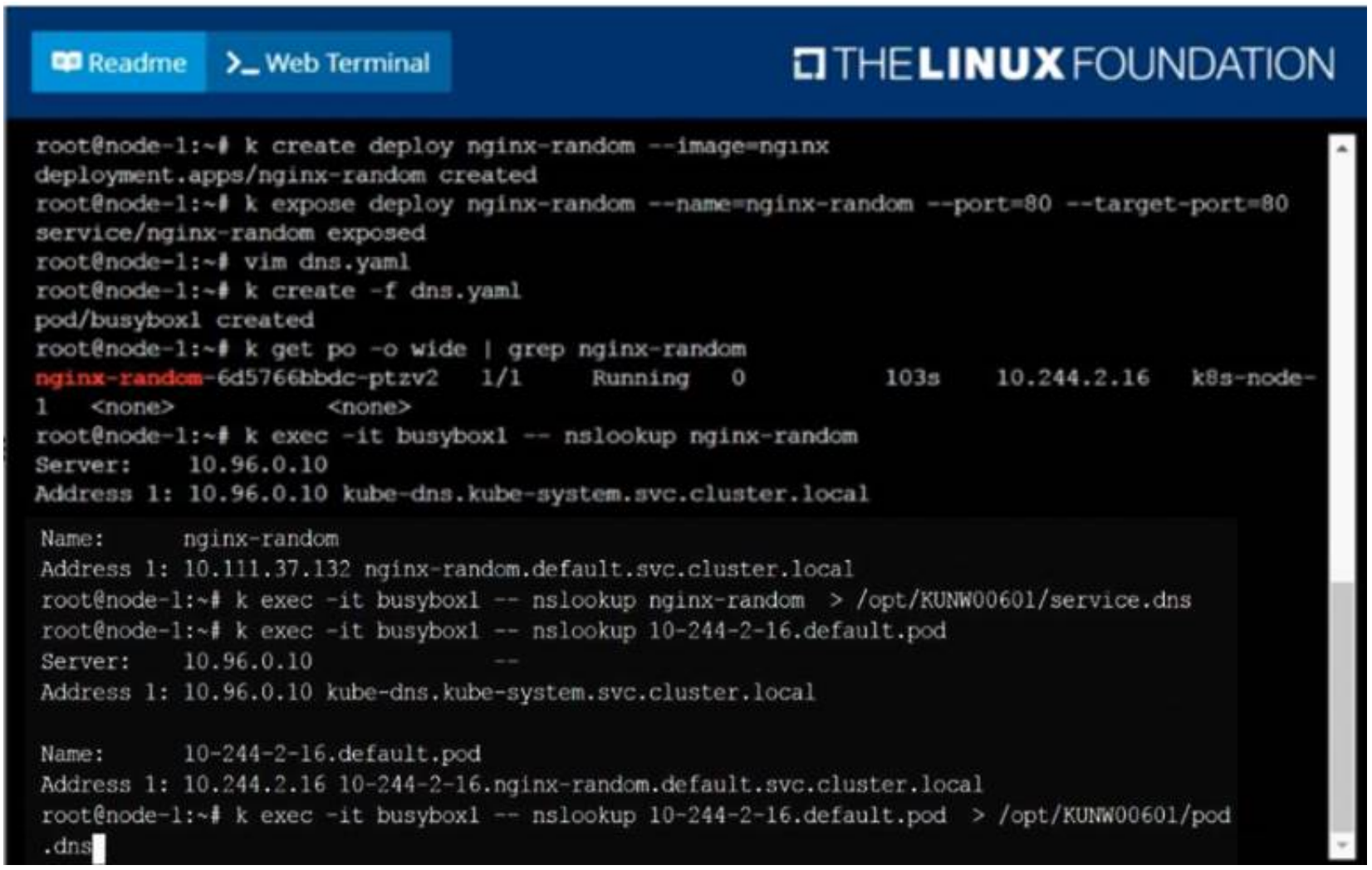
Solution:
F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 C.JPG



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



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



NEW QUESTION 30
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

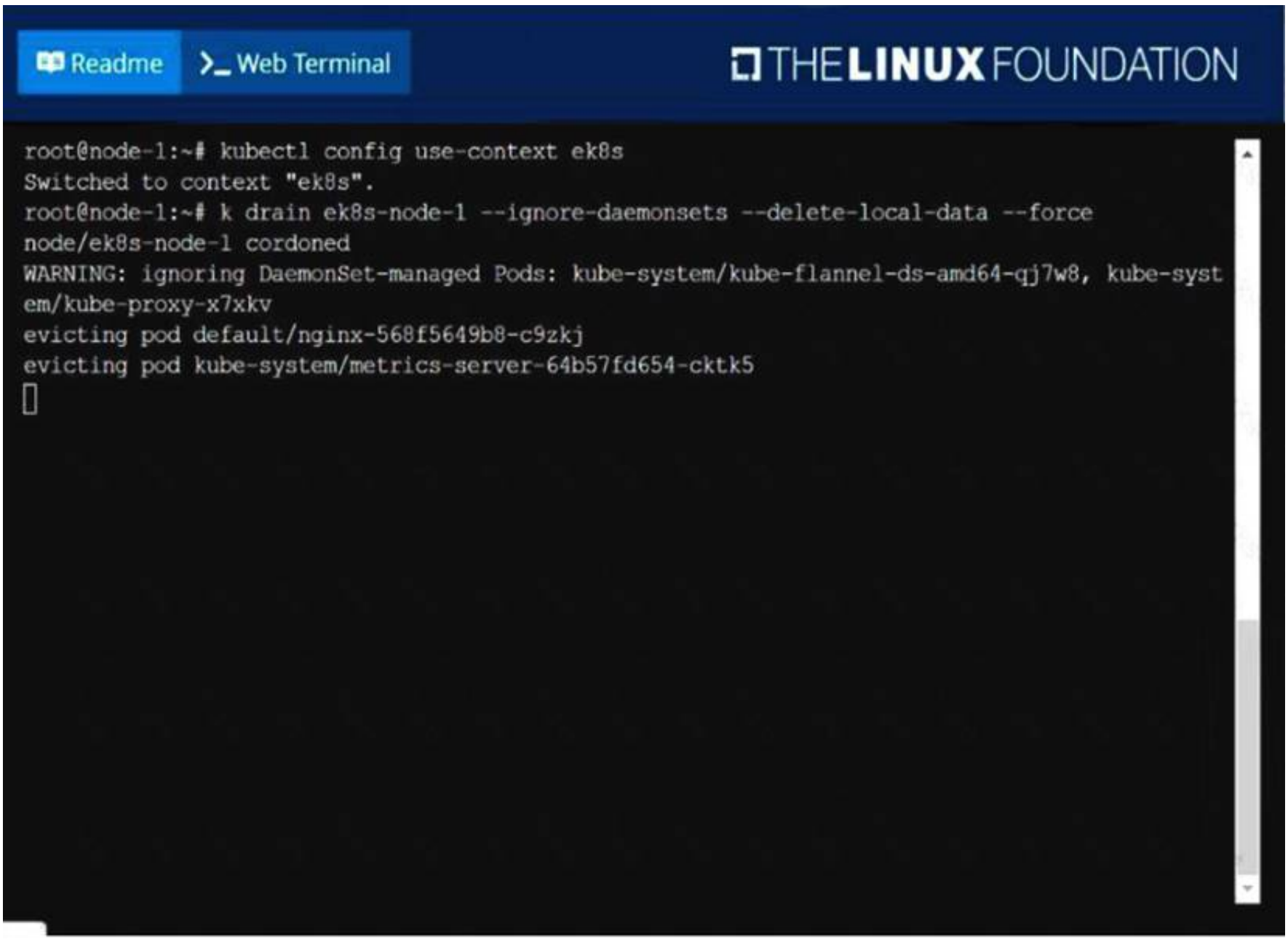
NEW QUESTION 33
Set the node named ek8s-node-1 as unavailable and reschedule all the pods running on it.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

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



NEW QUESTION 34
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

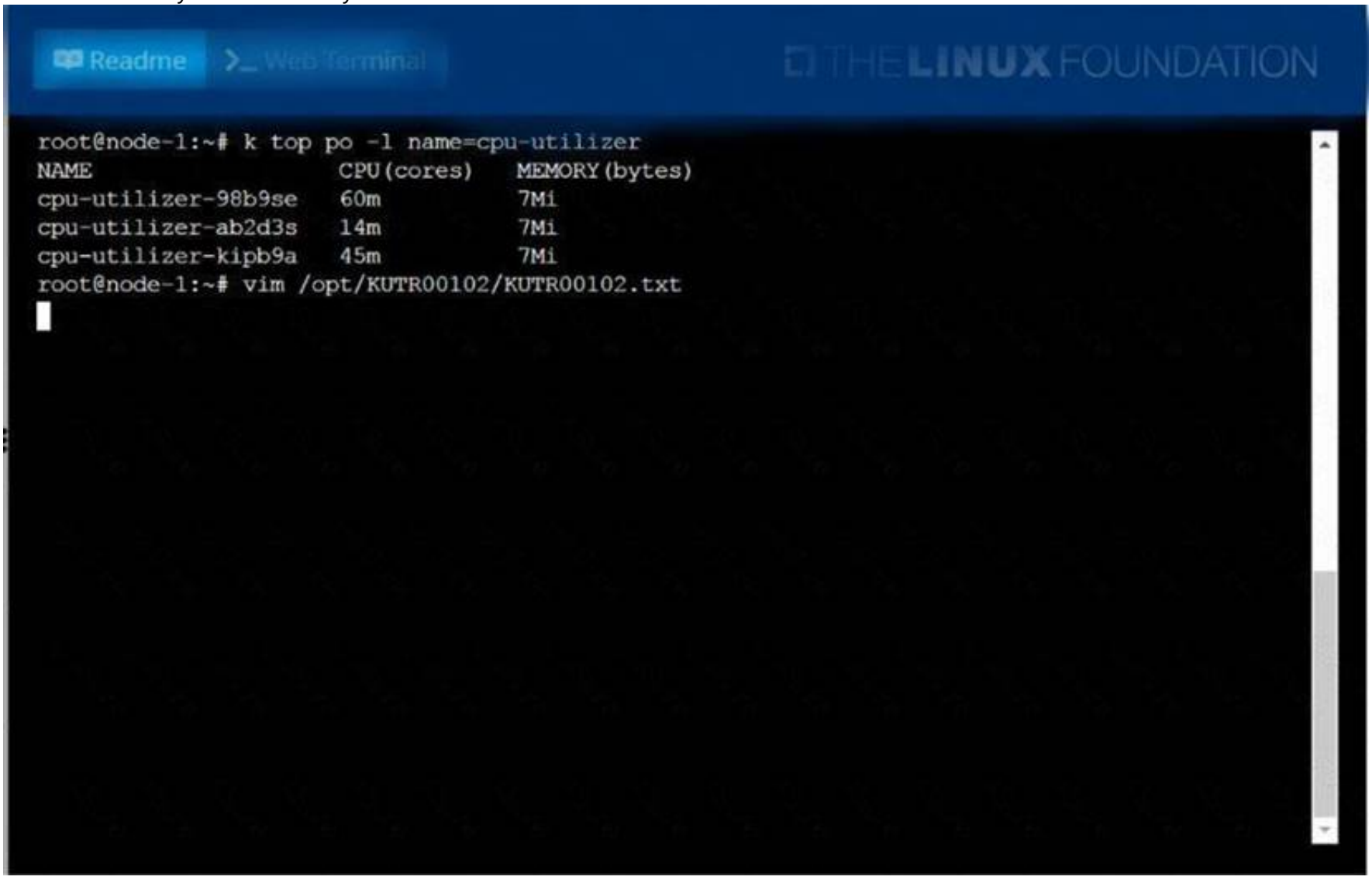
NEW QUESTION 36
From the pod label name=cpu-utilizer, find pods running high CPU workloads and write the name of the pod consuming most CPU to the file /opt/KUTR00102/KUTR00102.txt (which already exists).

A. Mastered
B. Not Mastered

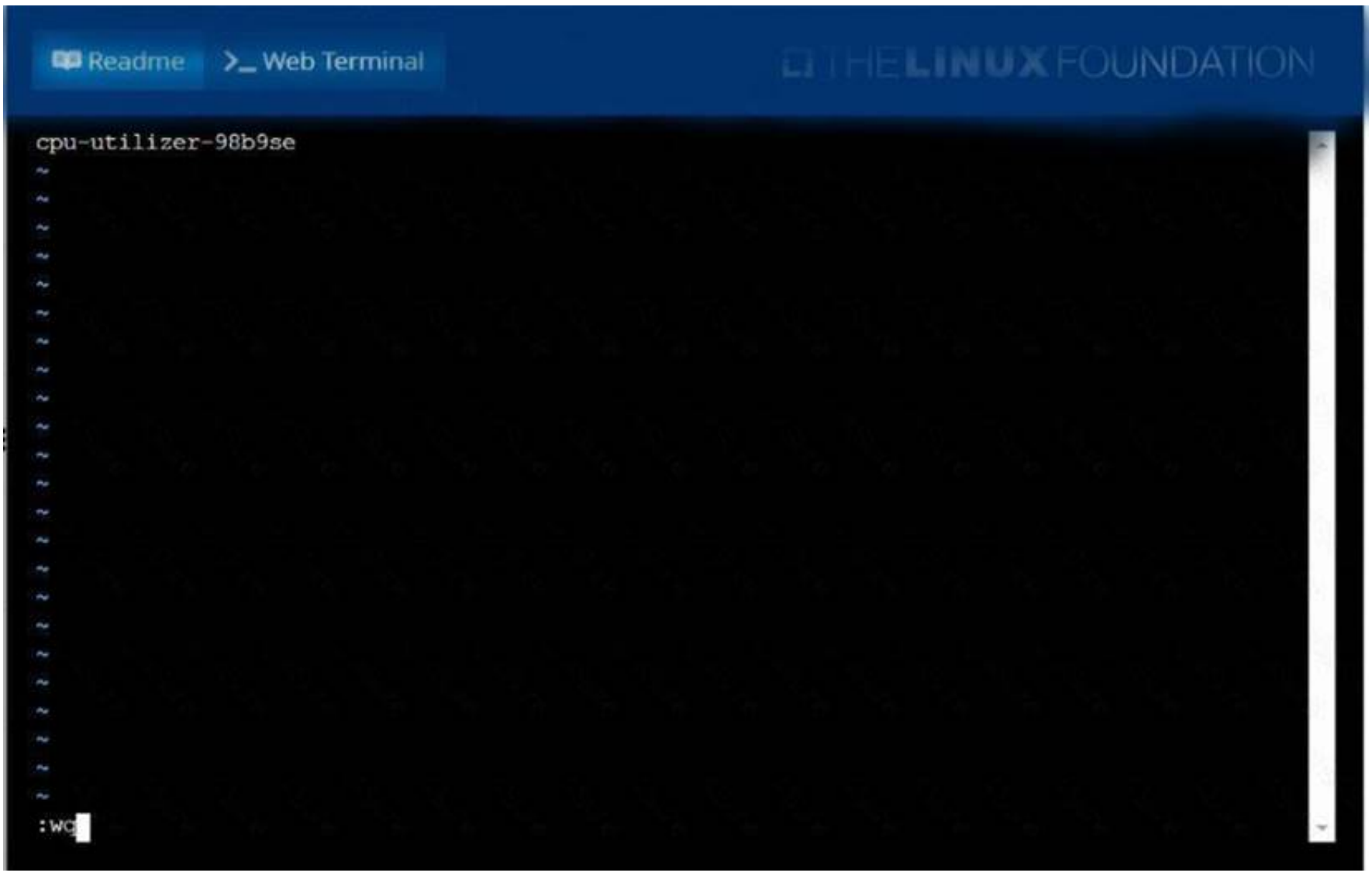
Answer: A

Explanation:

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



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



NEW QUESTION 40
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 43
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

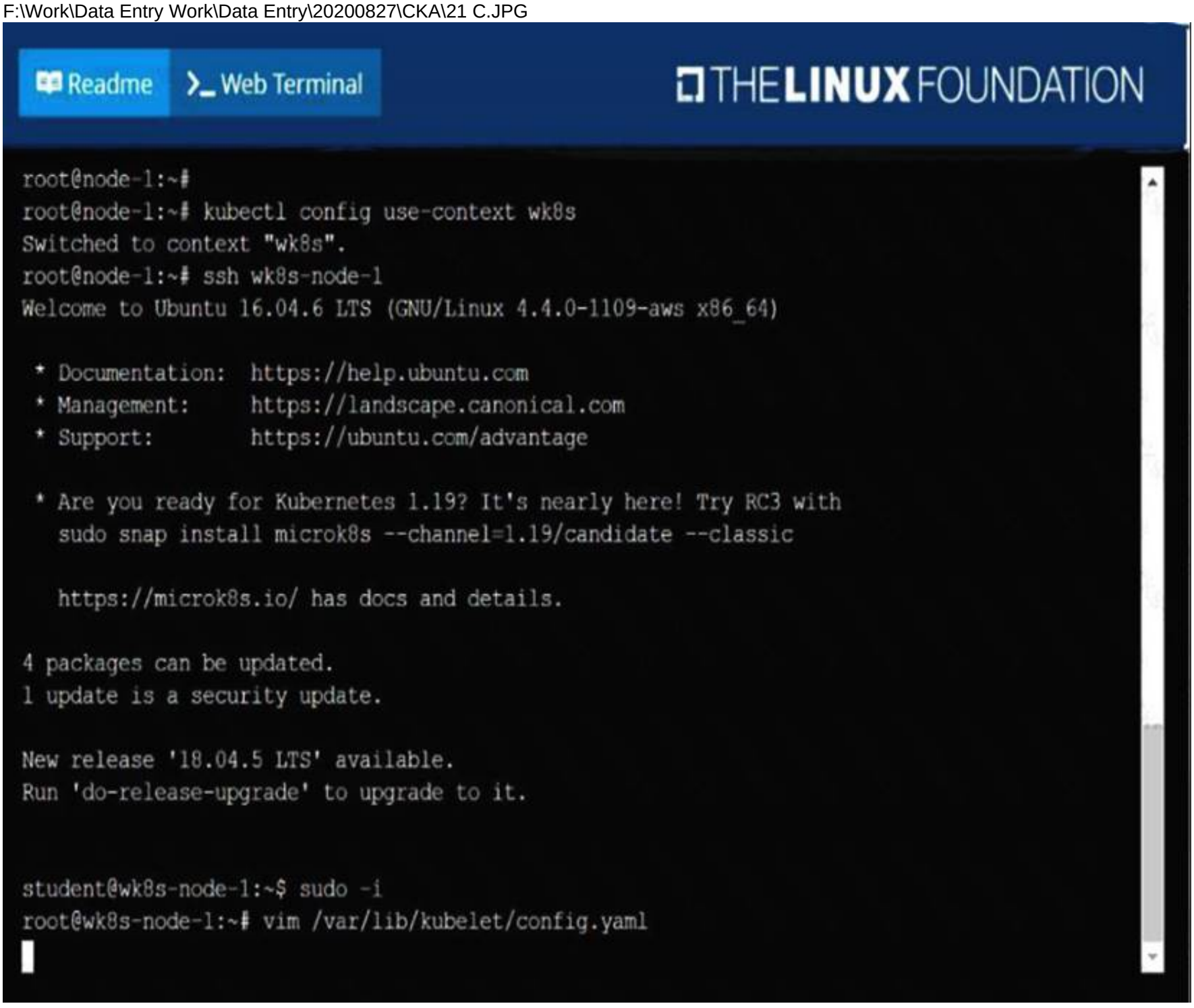
NEW QUESTION 48
Configure the kubelet systemd- managed service, on the node labelled with name=wk8s-node-1, to launch a pod containing a single container of Image httpd named webtool automatically. Any spec files required should be placed in the /etc/kubernetes/manifests directory on the node.
You can 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
F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 D.JPG

Readme | Web Terminal — THE LINUX FOUNDATION

```
    clientCAFile: /etc/kubernetes/pki/ca.crt
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
:wq
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 E.JPG

Readme | Web Terminal — THE LINUX FOUNDATION

```
root@node-1:~# ssh wk8s-node-1
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-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests#
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 F.JPG

Readme >_ Web Terminal THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: webtool
spec:
  containers:
  - name: webtool
    image: httpd
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:w
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 G.JPG

Readme >_ Web Terminal THE LINUX FOUNDATION

```
  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-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests#
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml
root@wk8s-node-1:/etc/kubernetes/manifests# systemctl  restart kubelet
root@wk8s-node-1:/etc/kubernetes/manifests# systemctl  enable kubelet
root@wk8s-node-1:/etc/kubernetes/manifests# exit
logout
student@wk8s-node-1:~$ exit
logout
Connection to 10.250.5.39 closed.
root@node-1:~# k get po
NAME                 READY   STATUS    RESTARTS   AGE
webtool-wk8s-node-1  1/1     Running   0          11s
root@node-1:~#
```

NEW QUESTION 52
Score: 4%

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 56
.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your CKA Exam with Our Prep Materials Via below:

https://www.certleader.com/CKA-dumps.html