Linux-Foundation

Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

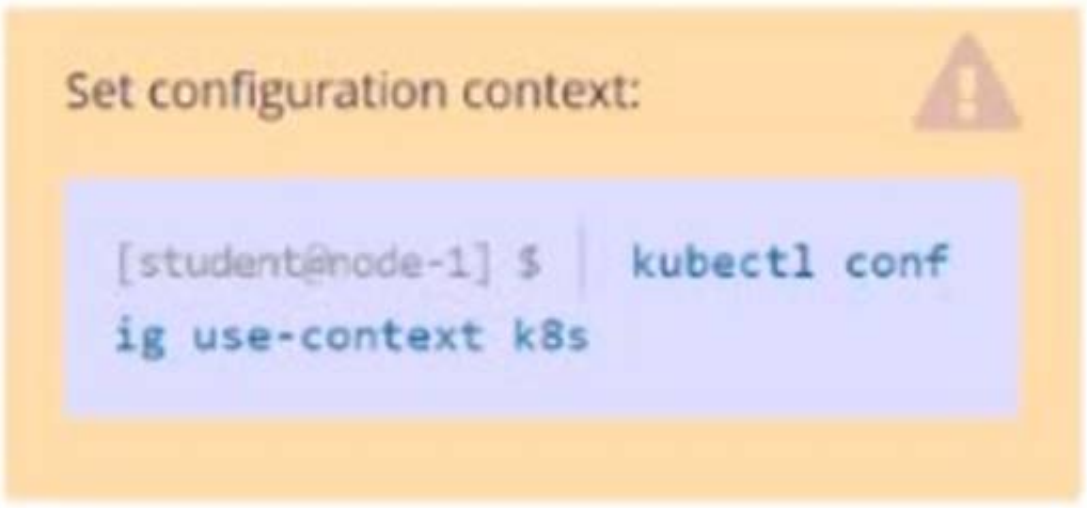
NEW QUESTION 1
CORRECT TEXT
Create a pod with image nginx called nginx and allow traffic on port 80

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%



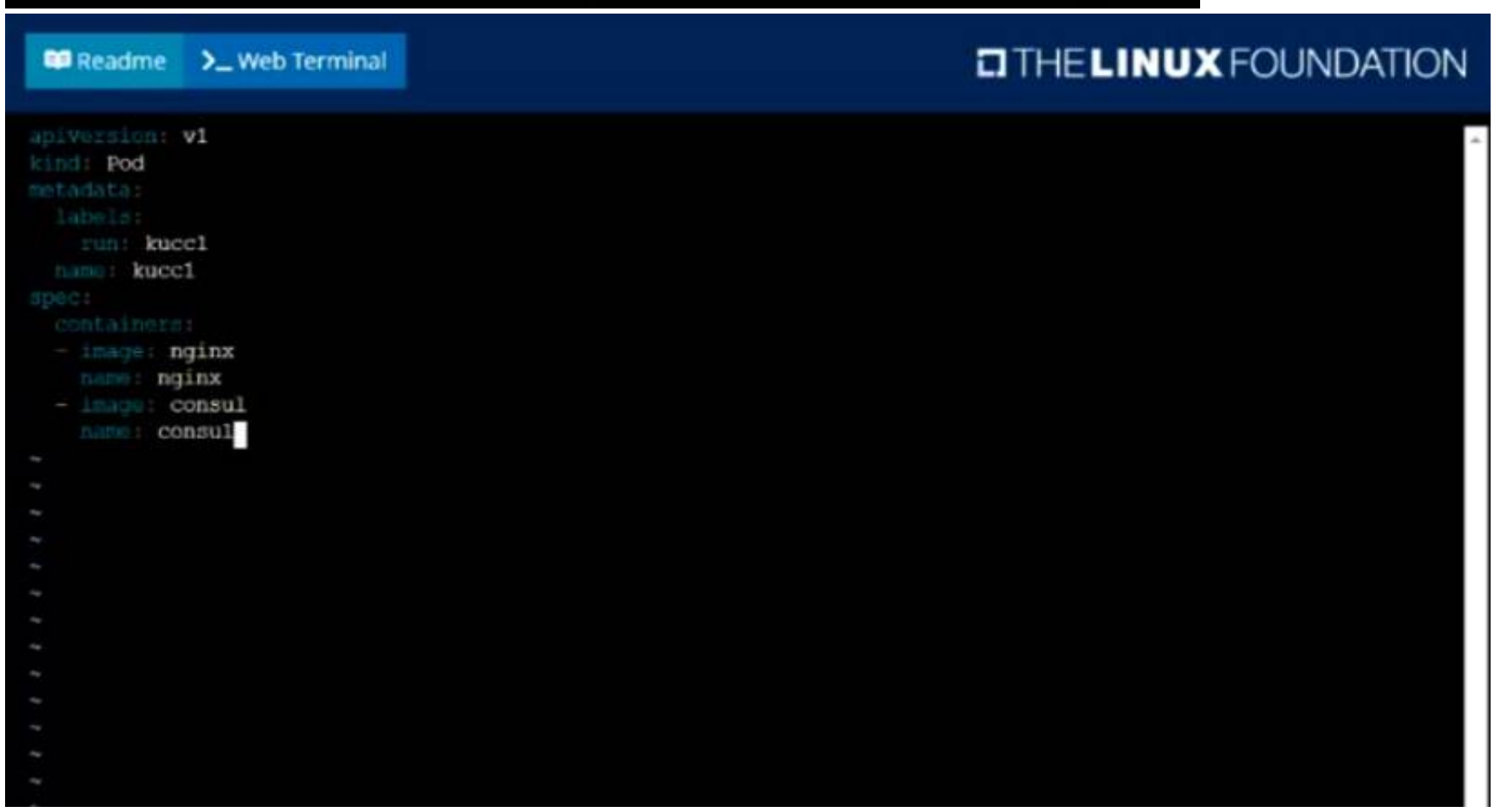
Task
Schedule a Pod as follows:
• Name: kucc1
• App Containers: 2
• Container Name/Images: o nginx
o consul

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
List pod logs named “frontend” and search for the pattern “started” and write it to a file “/opt/error-logs”

A. Mastered
B. Not Mastered

Answer: A

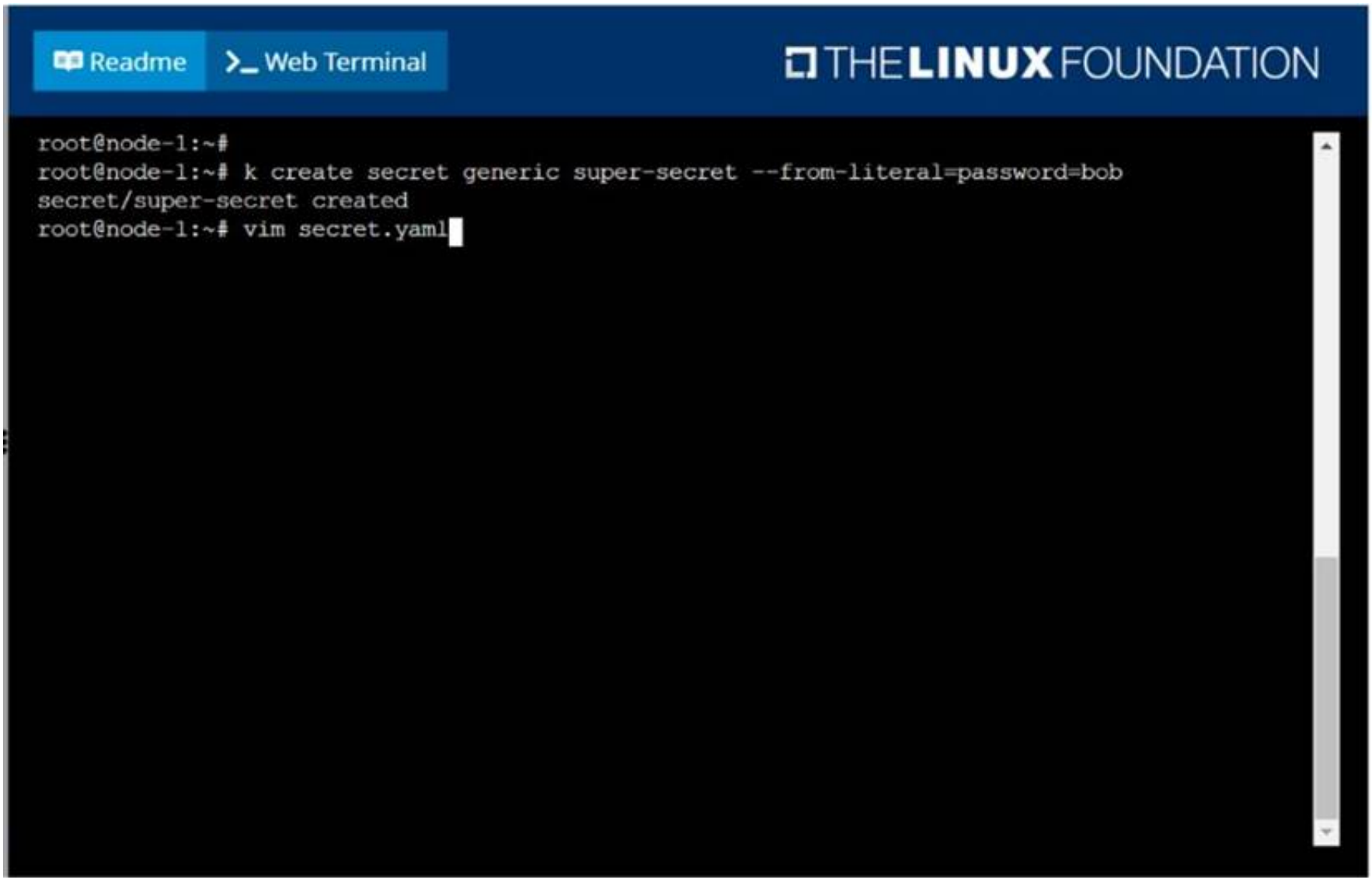
Explanation:
Kubectl logs frontend | grep -i “started” > /opt/error-logs

NEW QUESTION 5
CORRECT TEXT
Create a Kubernetes secret as follows:
? Name: super-secret
? password: bob
Create a pod named pod-secrets-via-file, using the redis Image, which mounts a secret named super-secret at /secrets.
Create a second pod named pod-secrets-via-env, using the redis Image, which exports password as CONFIDENTIAL

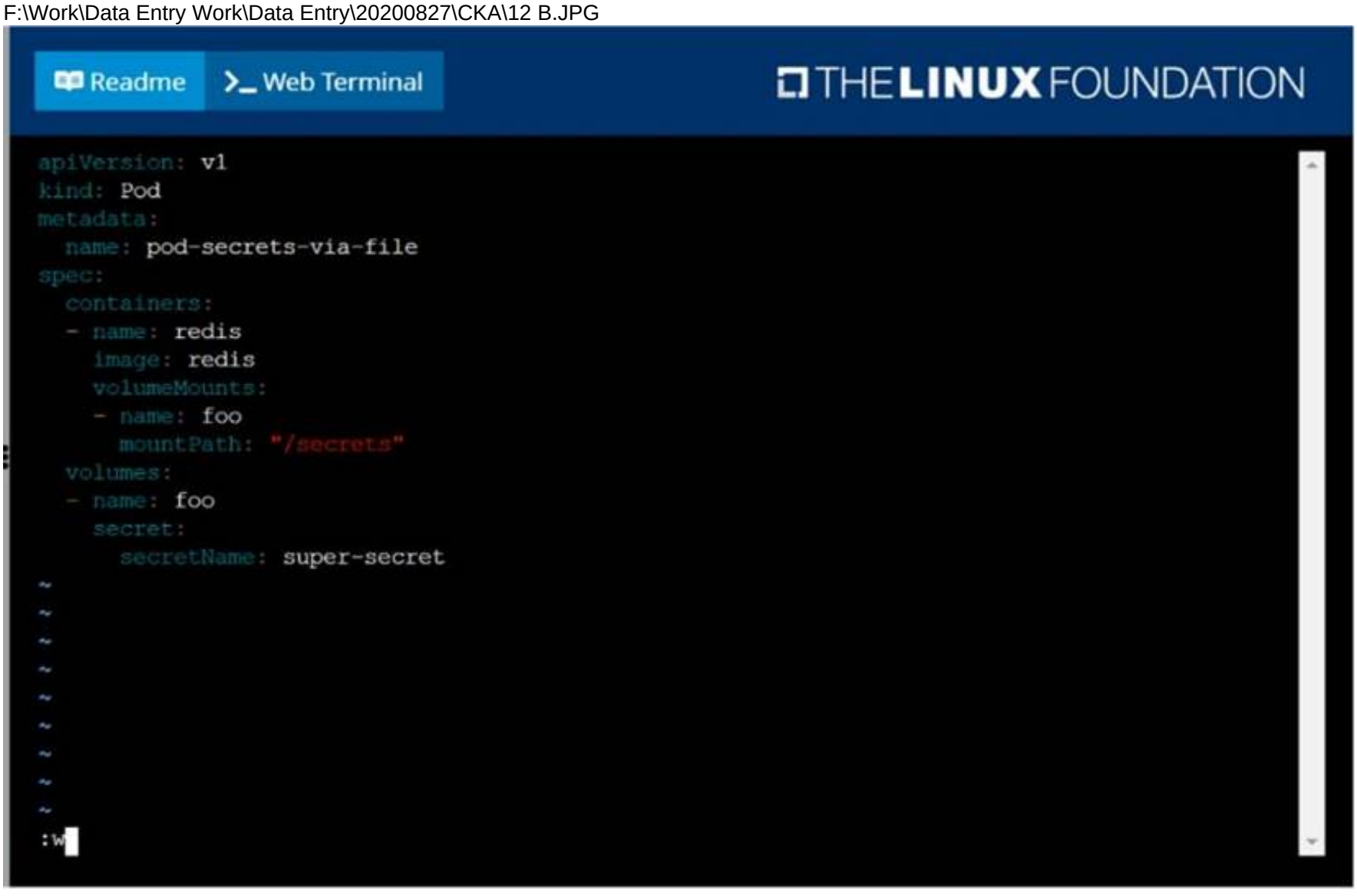
A. Mastered
B. Not Mastered

Answer: A

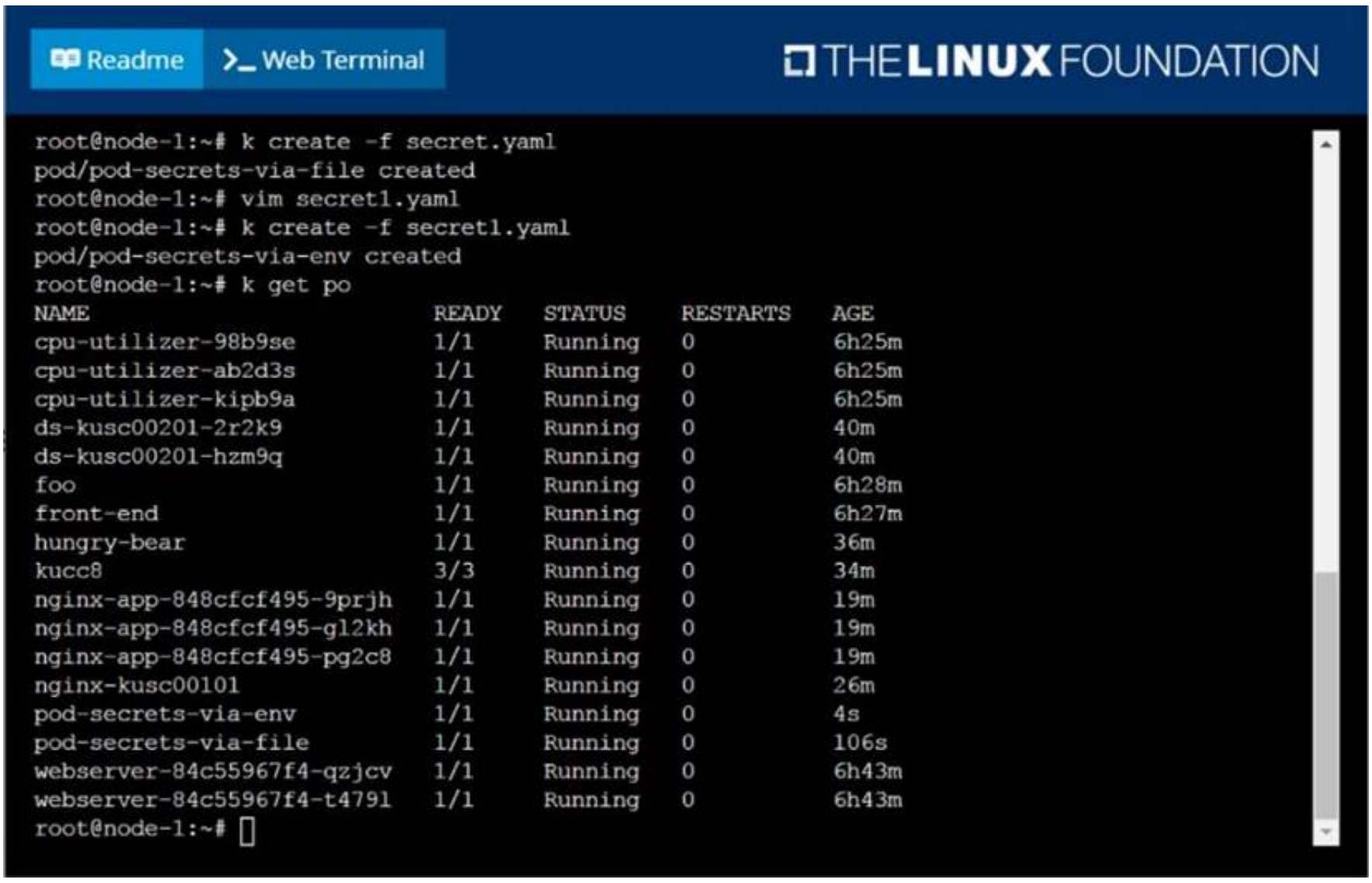
Explanation:
solution



F:\Work\Data Entry Work\Data Entry\20200827\CKA\12 B.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\12 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\12 D.JPG

NEW QUESTION 6
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:

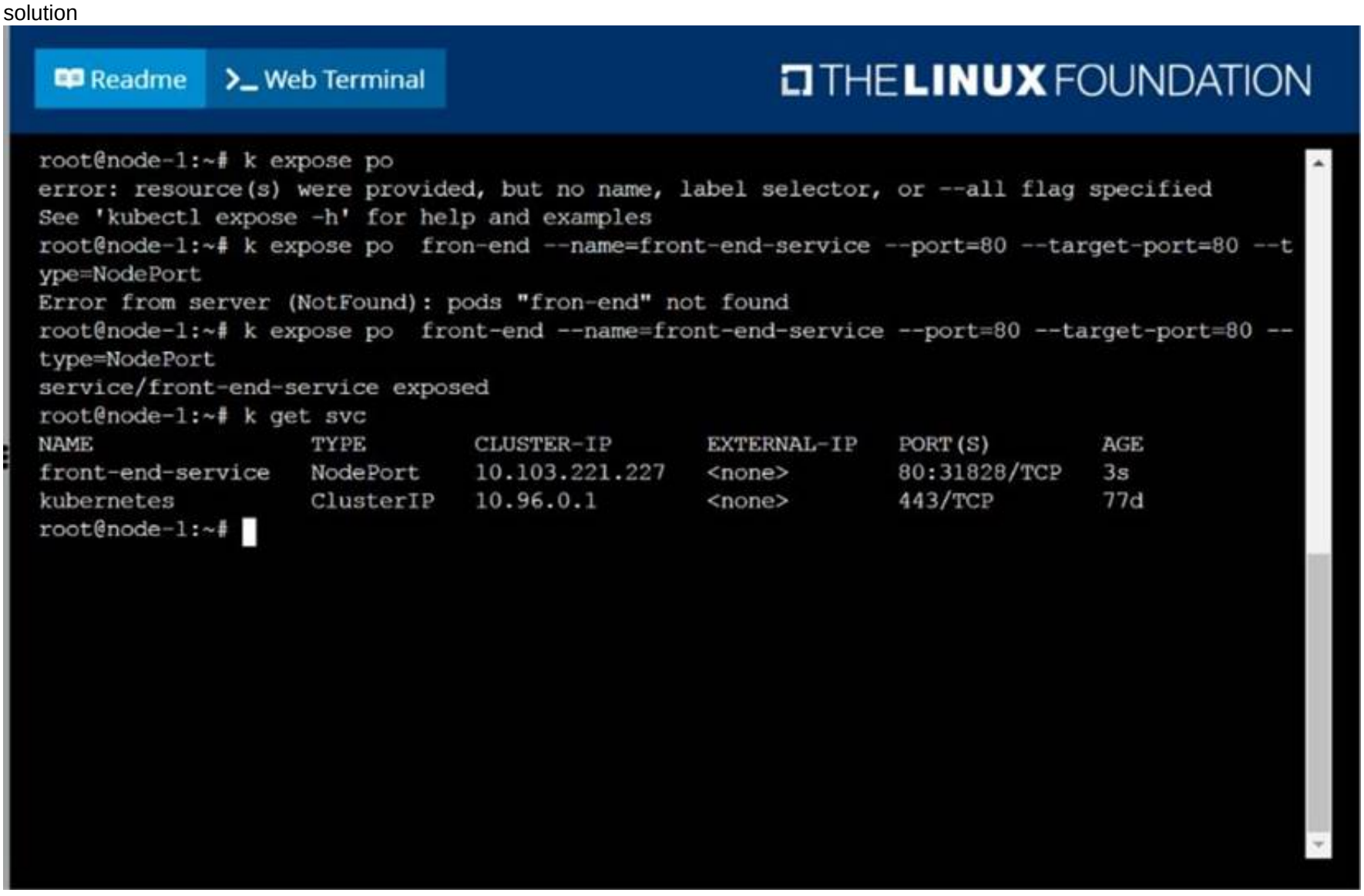
Solution:
kubectl describe nodes | grep ready|wc -l
kubectl describe nodes | grep -i taint | grep -i noschedule |wc -l
echo 3 > /opt/KUSC00402/kusc00402.txt
#
kubectl get node | grep -i ready |wc -l
taintsnoSchedule
kubectl describe nodes | grep -i taints | grep -i noschedule |wc -l
#
echo 2 > /opt/KUSC00402/kusc00402.txt

NEW QUESTION 7
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



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

NEW QUESTION 8
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

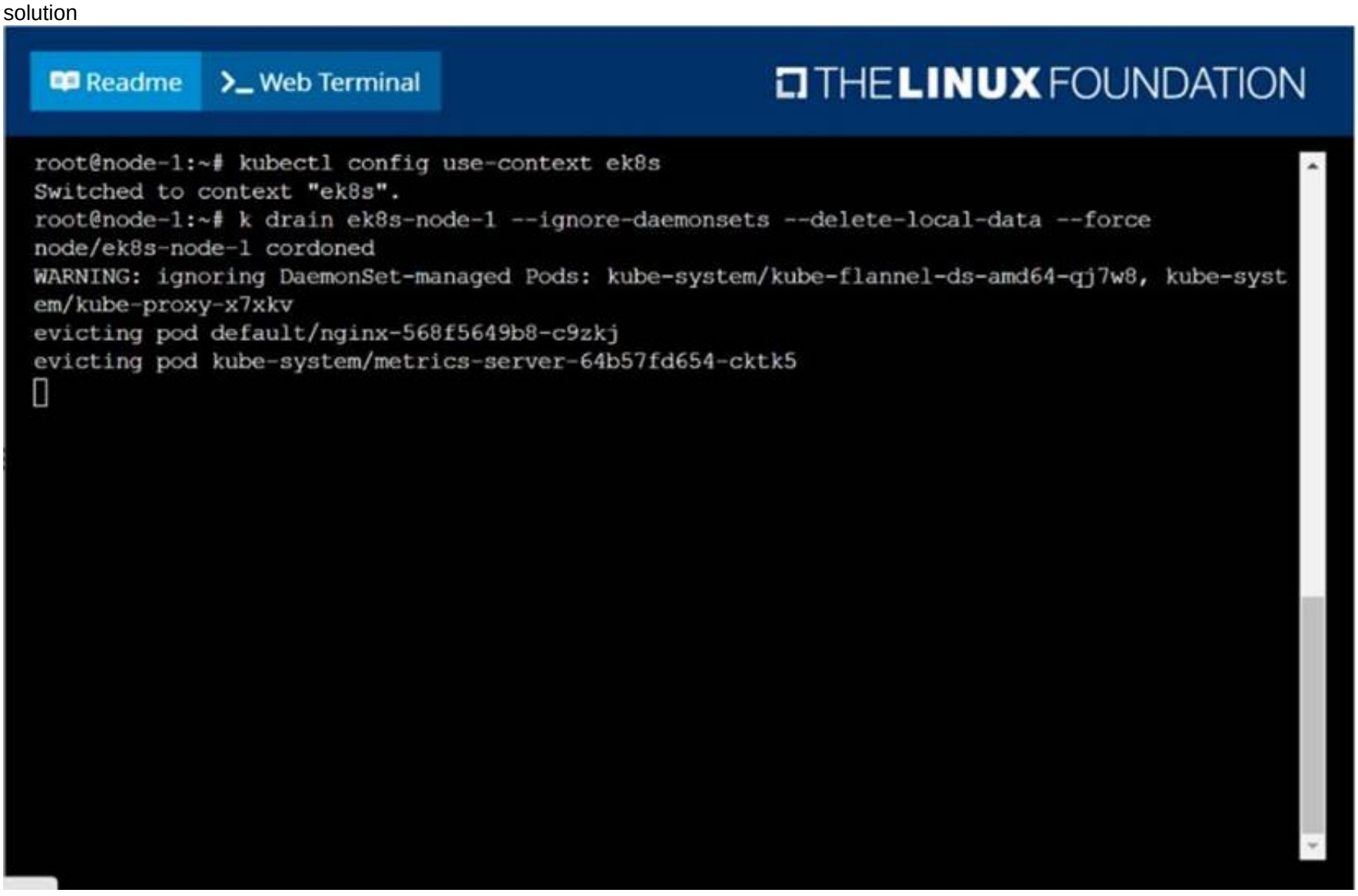
Explanation:
Solution:
kubectl top -l name=cpu-user -A
echo 'pod name' >> /opt/KUT00401/KUT00401.txt

NEW QUESTION 9
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



F:\Work\Data Entry Work\Data Entry\20200827\CKA\19 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

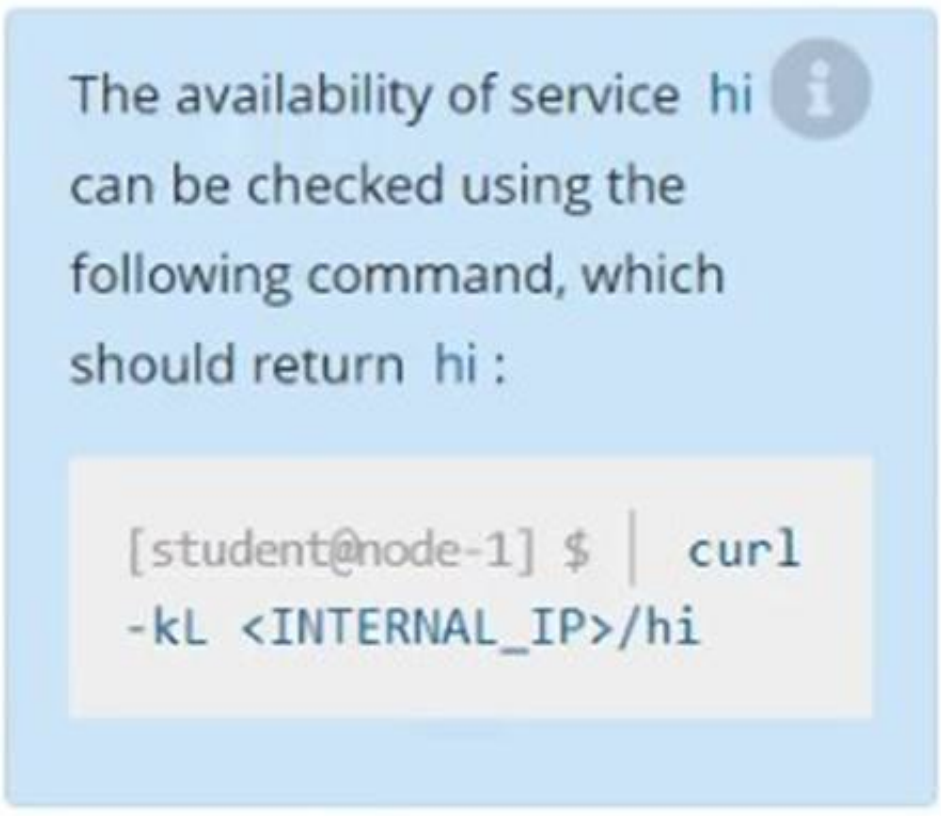
Explanation:

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 10
CORRECT TEXT
Score: 7%

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



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 14
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 18
CORRECT TEXT
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

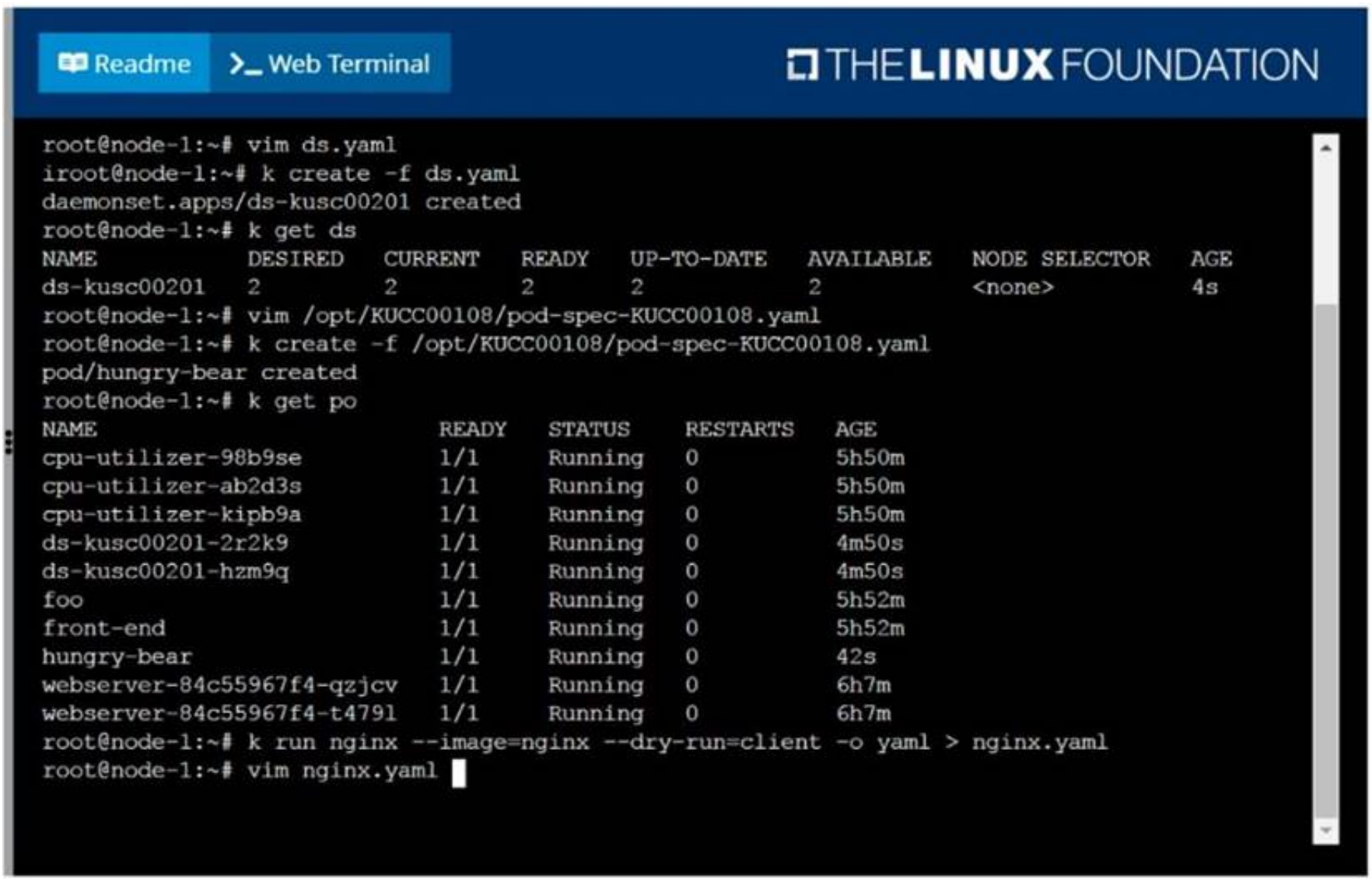
Explanation:
solution
You must use the kubeadm configuration file located at /etc/kubeadm.conf when initializingyour cluster.
You may use any CNI plugin to complete this task, but if you don't have your favourite CNI plugin's manifest URL at hand, Calico is one popular option:
https://docs.projectcalico.org/v3.14/manifests/calico.yaml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

NEW QUESTION 19
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

Readme | Web Terminal — THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: kucc8
spec:
  containers:
  - image: nginx
    name: nginx
  - image: redis
    name: redis
  - image: memcached
    name: memcached
~
~
~
~
~
~
~
~
~
~
:w
```

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

Readme | Web Terminal — THE LINUX FOUNDATION

```
cpu-utilizer-98b9se          1/1     Running             0          5h51m
cpu-utilizer-ab2d3s          1/1     Running             0          5h51m
cpu-utilizer-kipb9a          1/1     Running             0          5h51m
ds-kusc00201-2r2k9           1/1     Running             0          6m12s
ds-kusc00201-hzm9q           1/1     Running             0          6m12s
foo                          1/1     Running             0          5h54m
front-end                    1/1     Running             0          5h53m
hungry-bear                  1/1     Running             0          2m4s
kucc8                        0/3     ContainerCreating   0          4s
webserver-84c55967f4-qzjcv   1/1     Running             0          6h9m
webserver-84c55967f4-t4791   1/1     Running             0          6h9m
root@node-1:~# k get po
NAME                         READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se          1/1     Running   0          5h52m
cpu-utilizer-ab2d3s          1/1     Running   0          5h52m
cpu-utilizer-kipb9a          1/1     Running   0          5h52m
ds-kusc00201-2r2k9           1/1     Running   0          6m31s
ds-kusc00201-hzm9q           1/1     Running   0          6m31s
foo                          1/1     Running   0          5h54m
front-end                    1/1     Running   0          5h54m
hungry-bear                  1/1     Running   0          2m23s
kucc8                        3/3     Running   0          23s
webserver-84c55967f4-qzjcv   1/1     Running   0          6h9m
webserver-84c55967f4-t4791   1/1     Running   0          6h9m
root@node-1:~#
```

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

NEW QUESTION 20
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

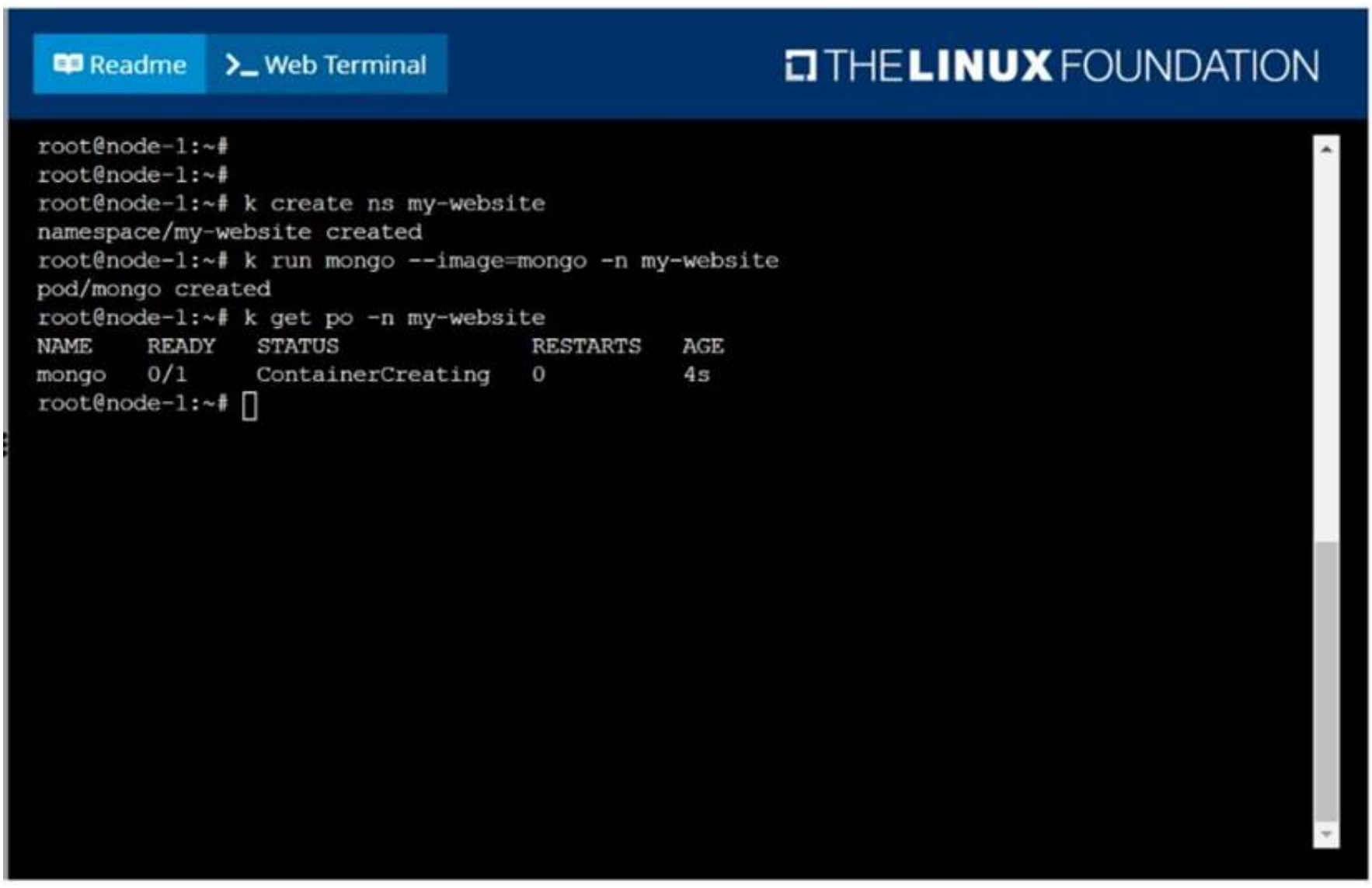
kubectl get po nginx --v=8
kubectl get po nginx --v=9

NEW QUESTION 21
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



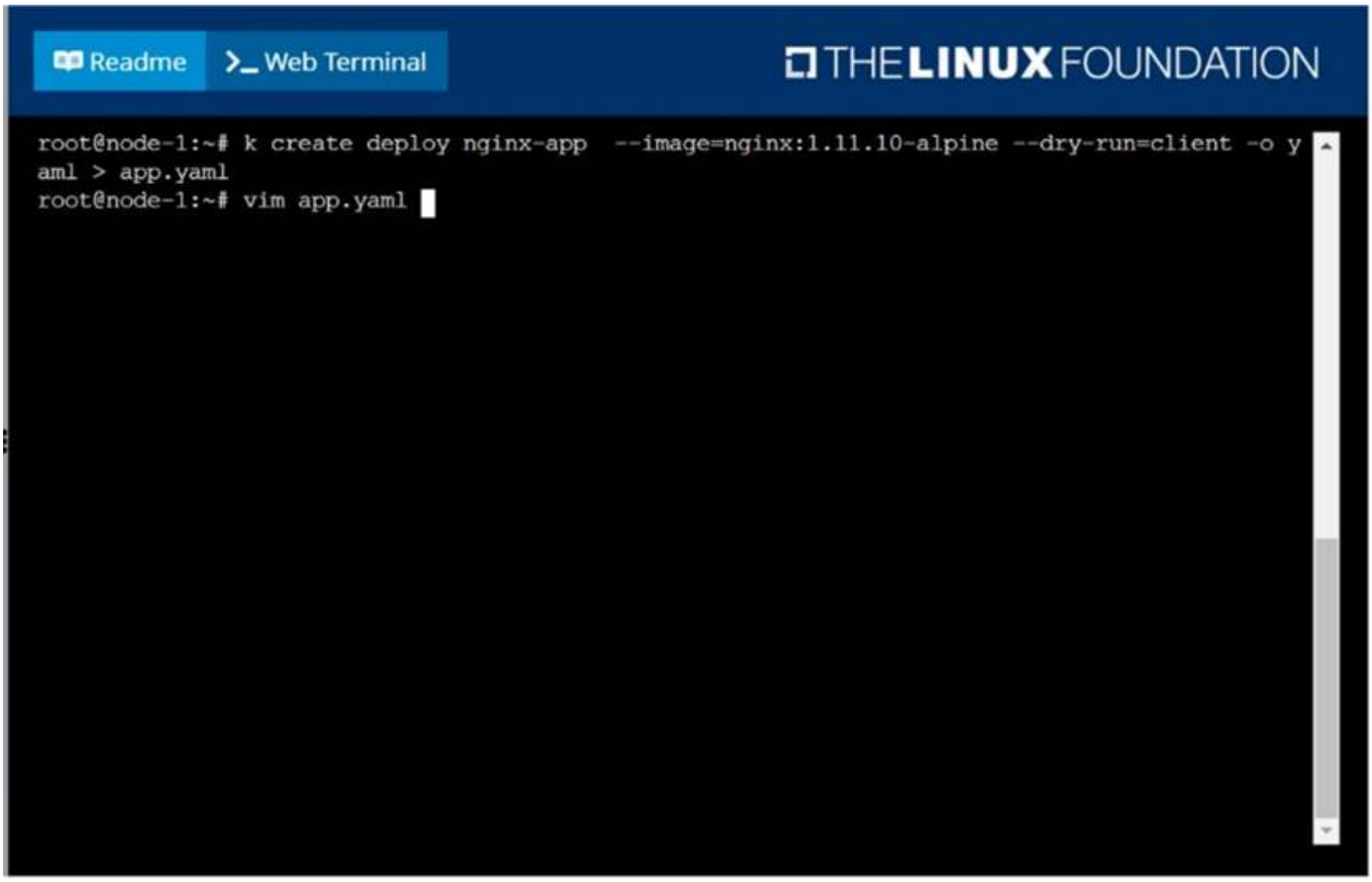
F:\Work\Data Entry Work\Data Entry\20200827\CKA\9 B.JPG

NEW QUESTION 24
CORRECT TEXT
Create a deployment as follows:
? Name: nginx-app
? Using container nginx with version 1.11.10-alpine
? The deployment should contain 3 replicas
Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update.
Finally, rollback that update to the previous version 1.11.10-alpine.

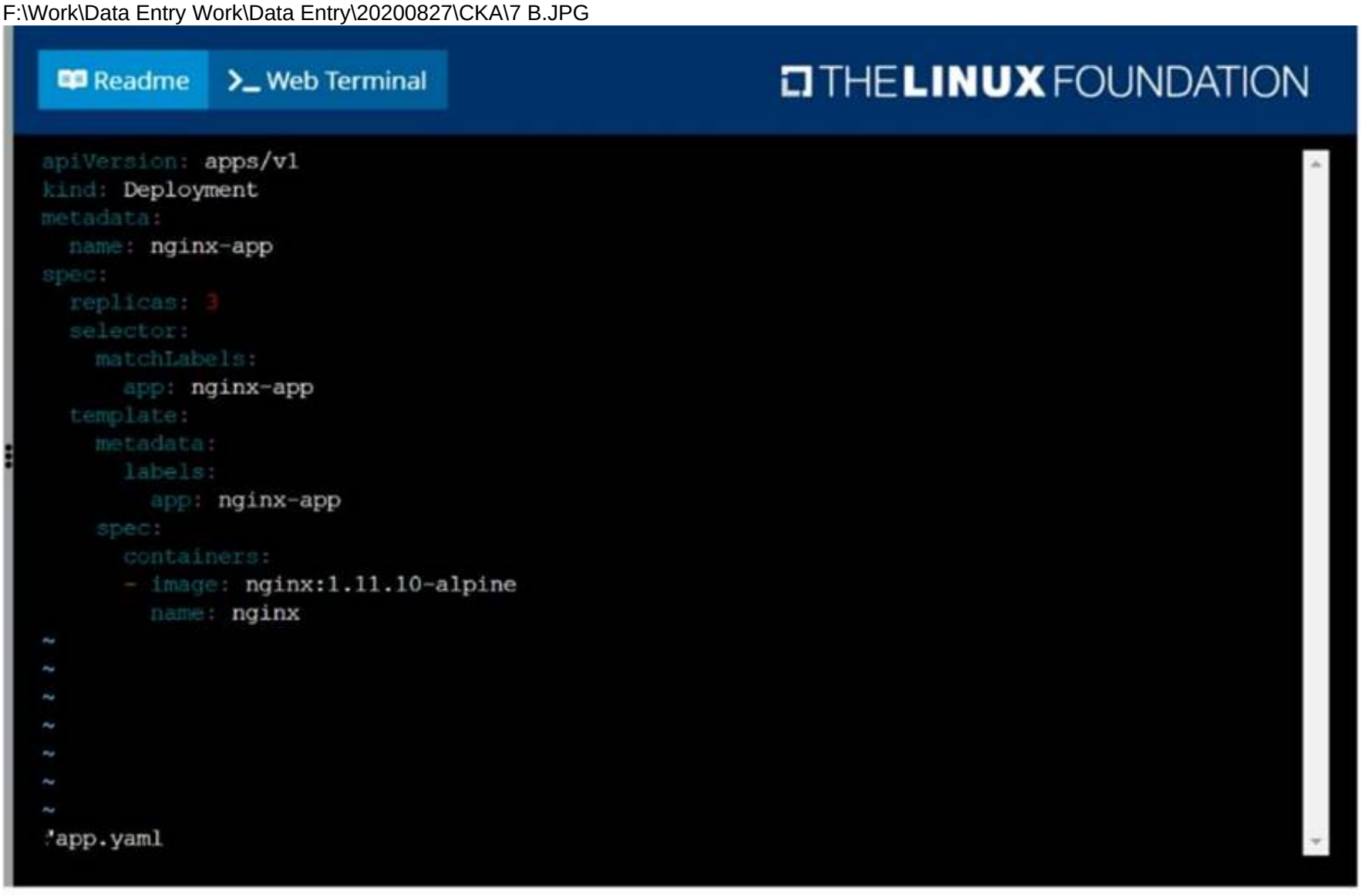
A. Mastered
B. Not Mastered

Answer: A

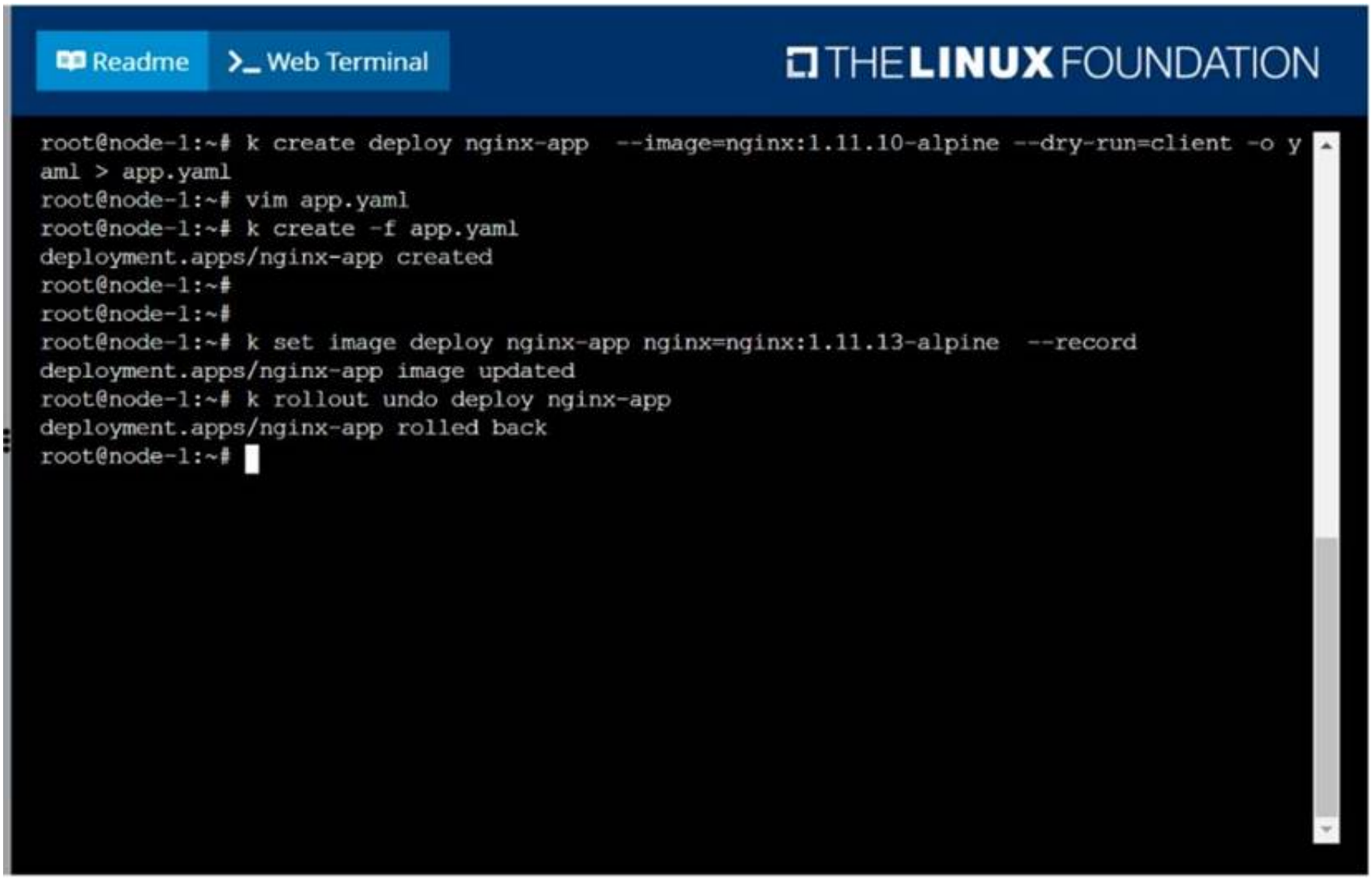
Explanation:
solution



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



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



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

NEW QUESTION 27
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"

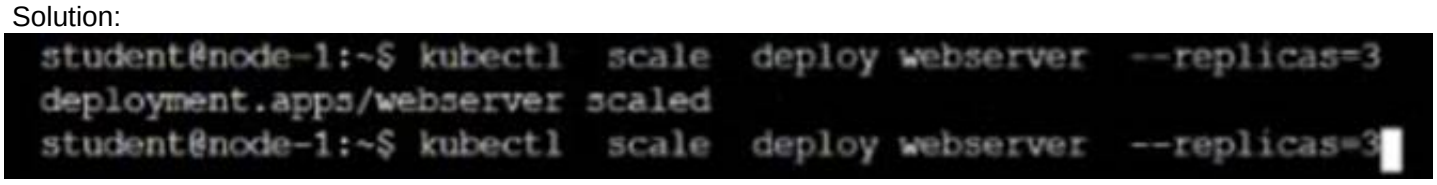
NEW QUESTION 30
CORRECT TEXT
Task Weight: 4%

Task
Scale the deployment webserver to 3 pods.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:



NEW QUESTION 35
CORRECT TEXT
List the nginx pod with custom columns POD_NAME and POD_STATUS

A. Mastered
B. Not Mastered

Answer: A

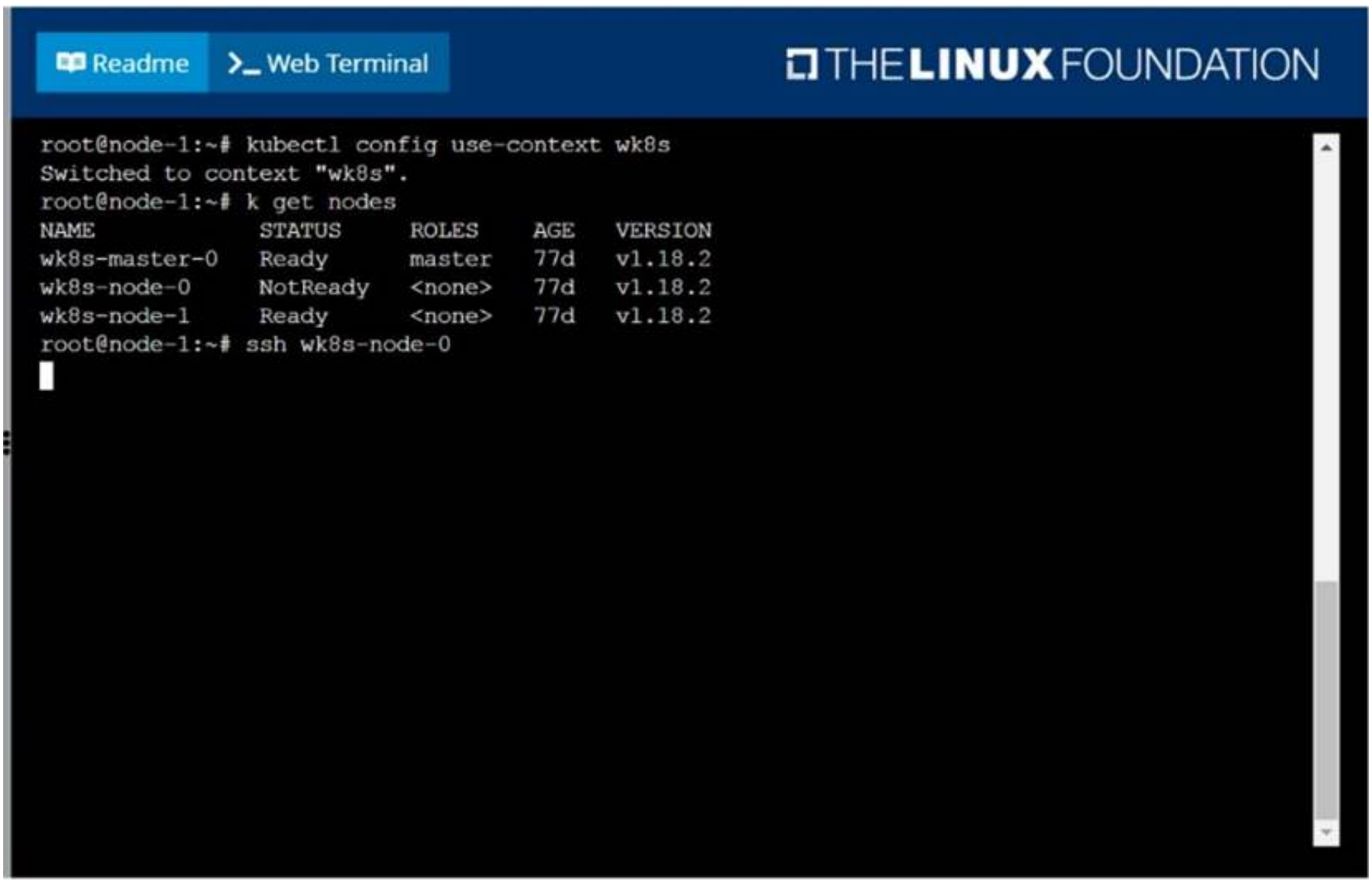
Explanation:
kubectl get po -o=custom-columns="POD_NAME:.metadata.name, POD_STATUS:.status.containerStatuses[].state"

NEW QUESTION 40
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

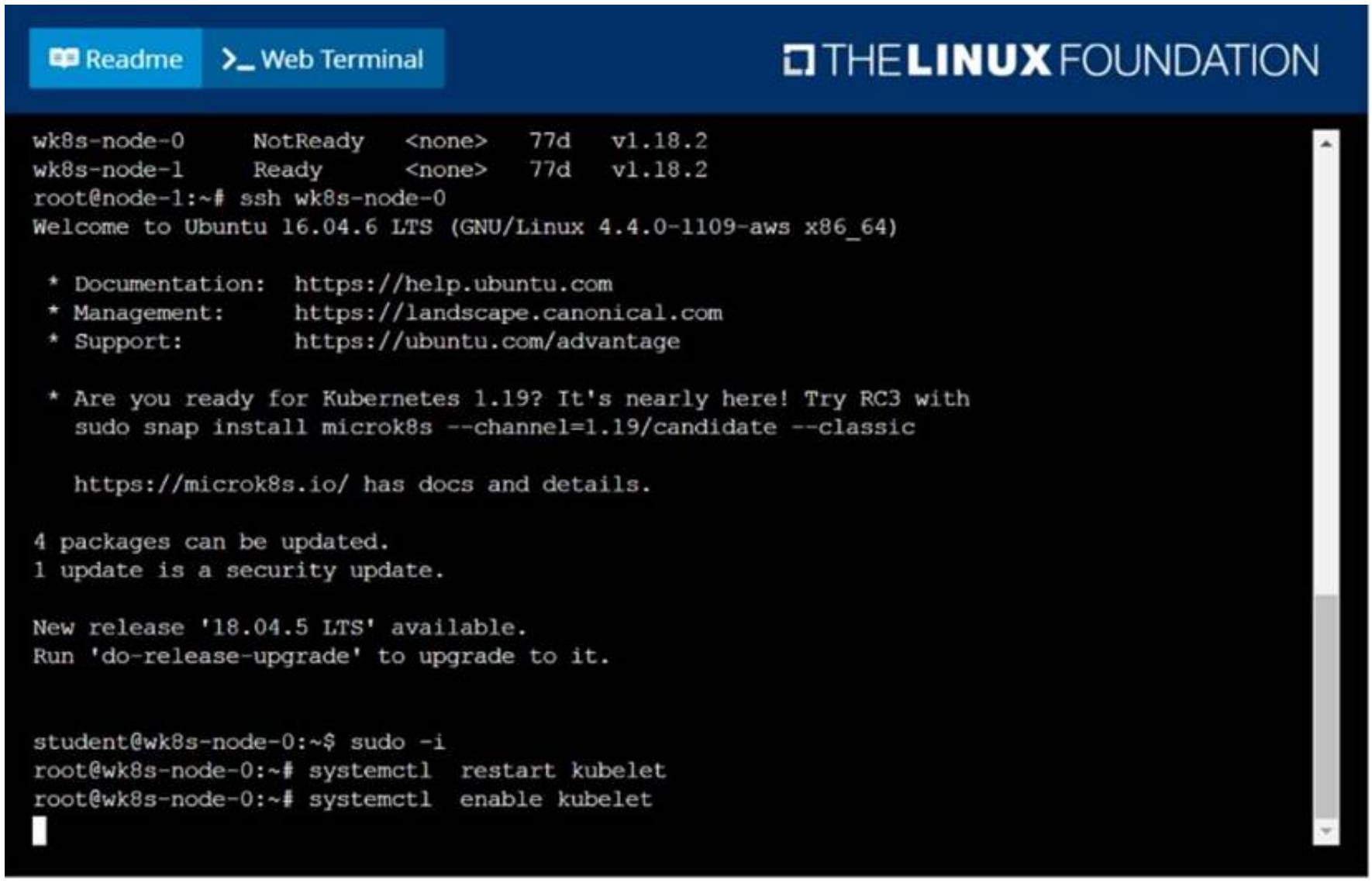
A. Mastered
B. Not Mastered

Answer: A

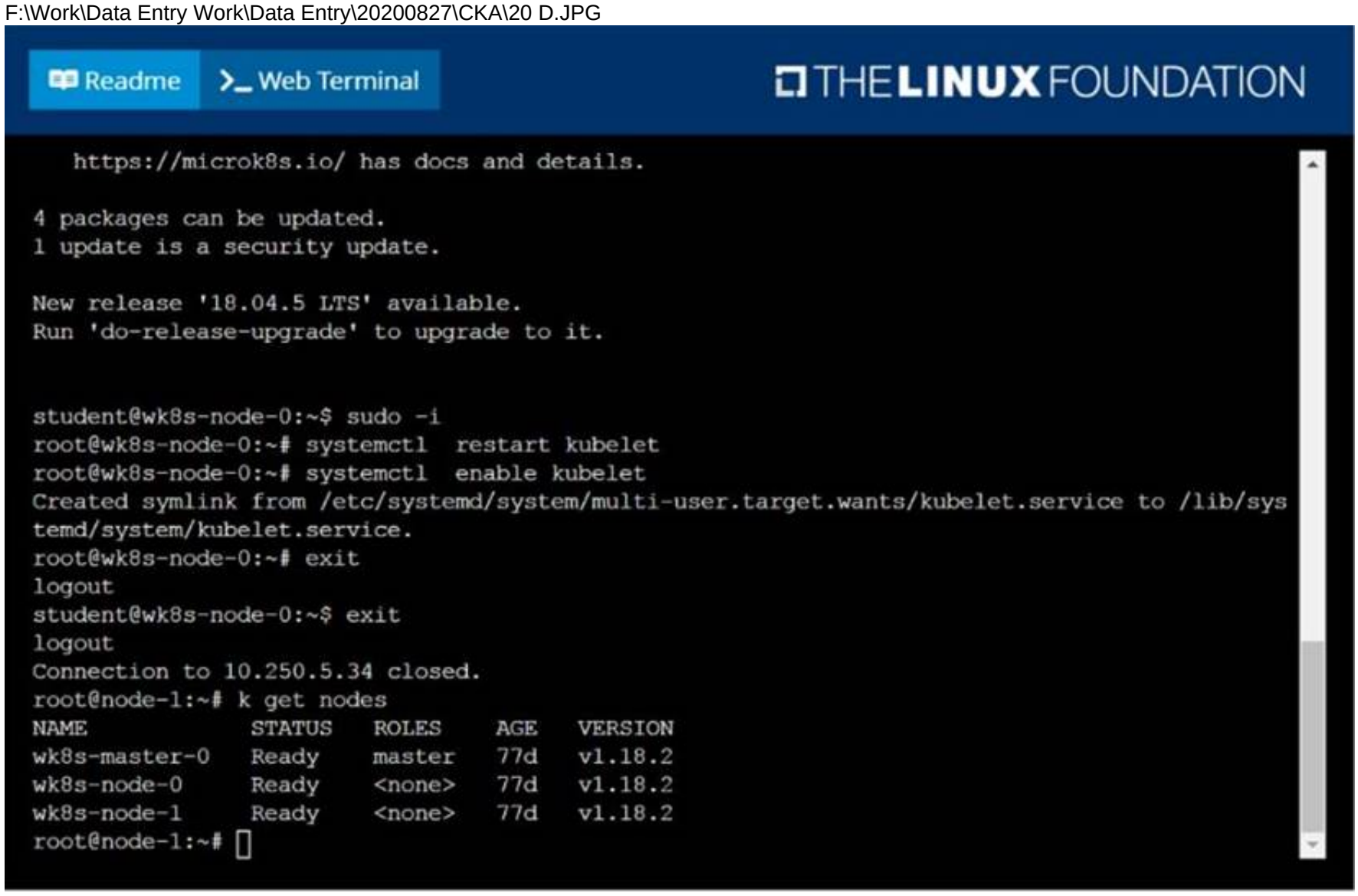
Explanation:
solution



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



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



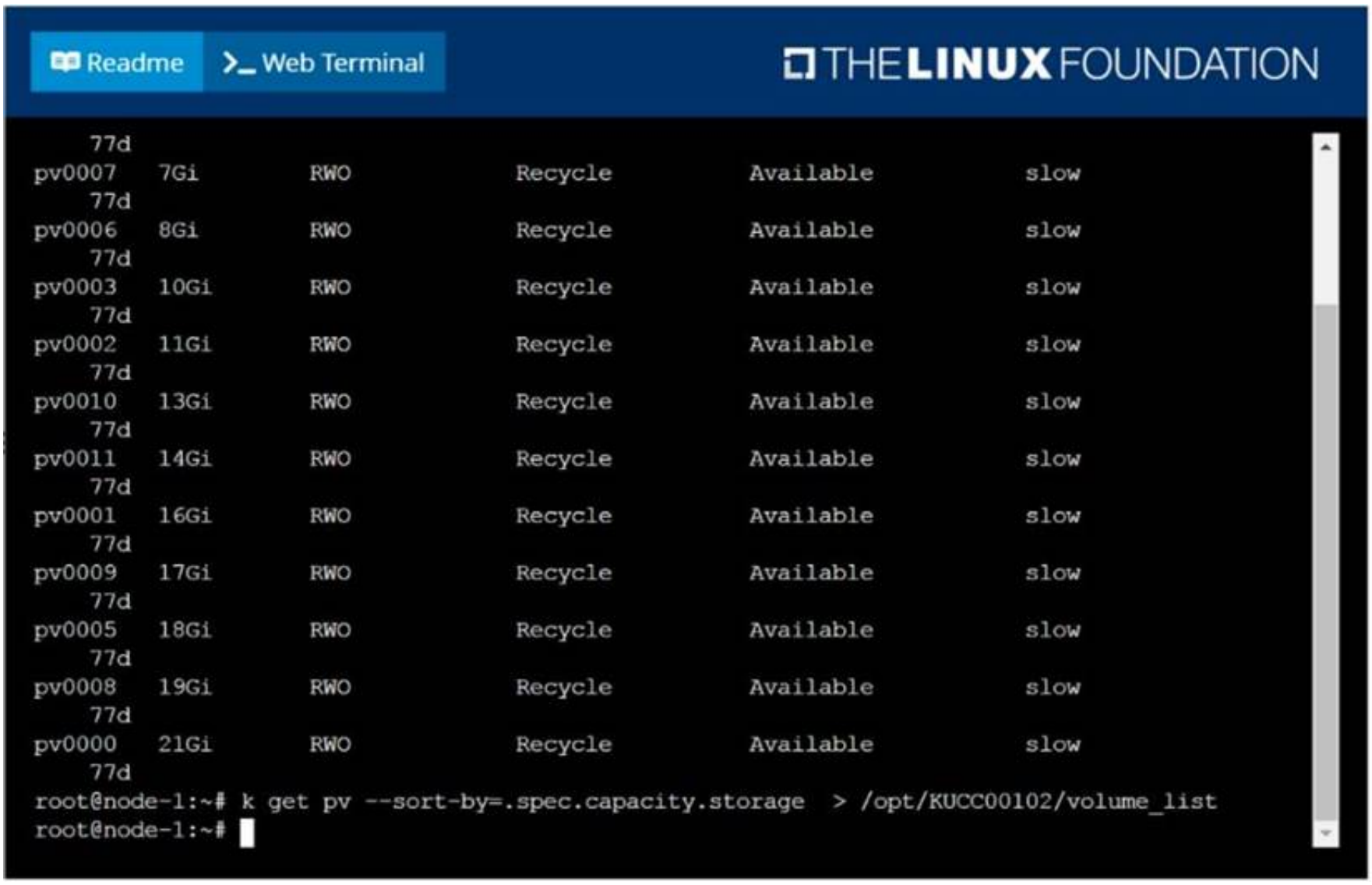
F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 E.JPG

NEW QUESTION 41
CORRECT TEXT
List all persistent volumes sorted by capacity, saving the full kubectl output to
/opt/KUCC00102/volume_list. Use kubectl 's own functionality for sorting the output, and do not manipulate it any further.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
solution



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

NEW QUESTION 44
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 49

CORRECT TEXT

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

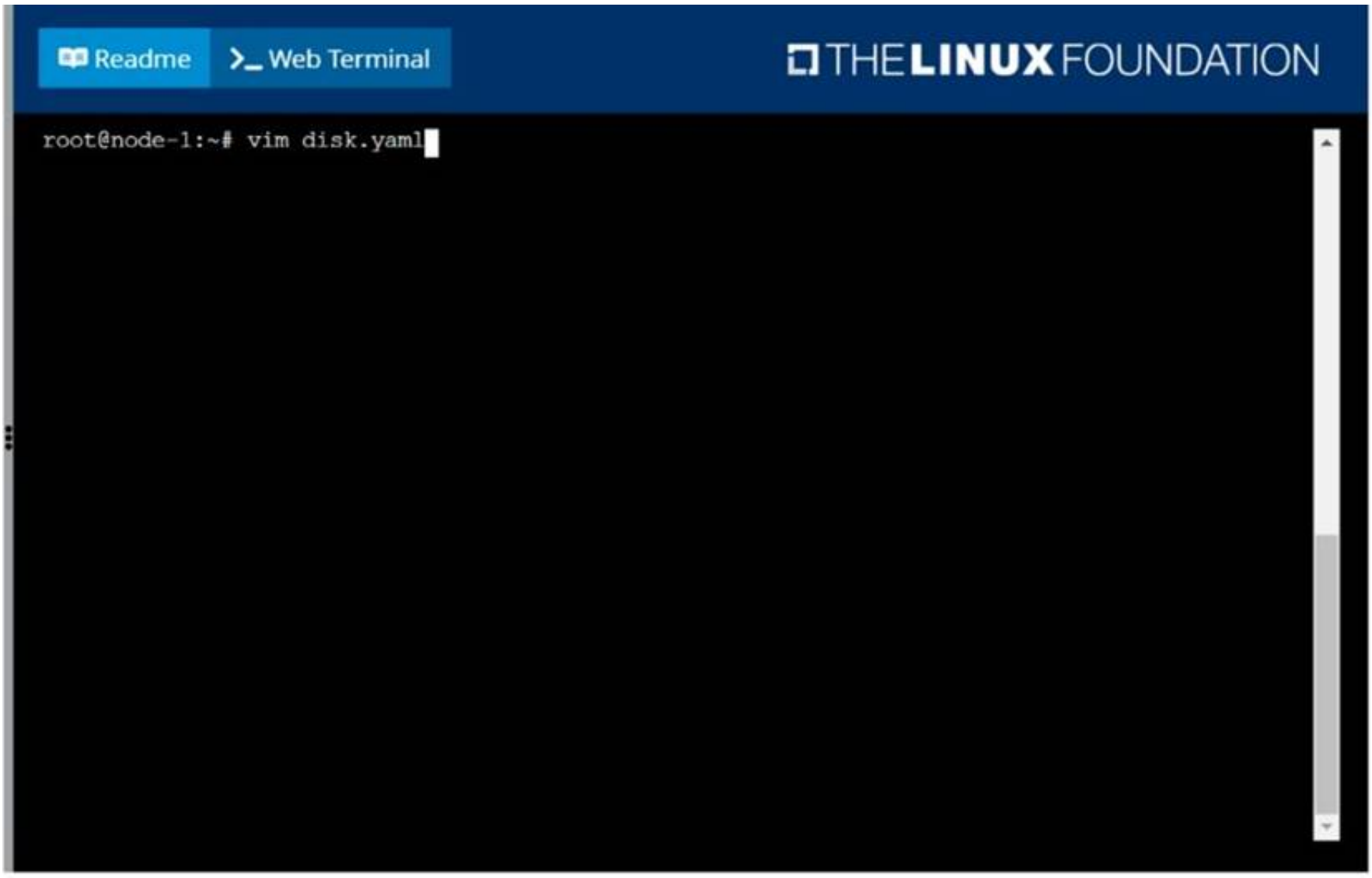
"sleep 3600"

NEW QUESTION 51
CORRECT TEXT
Schedule a pod as follows:
? Name: nginx-kusc00101
? Image: nginx
? Node selector: disk=ssd

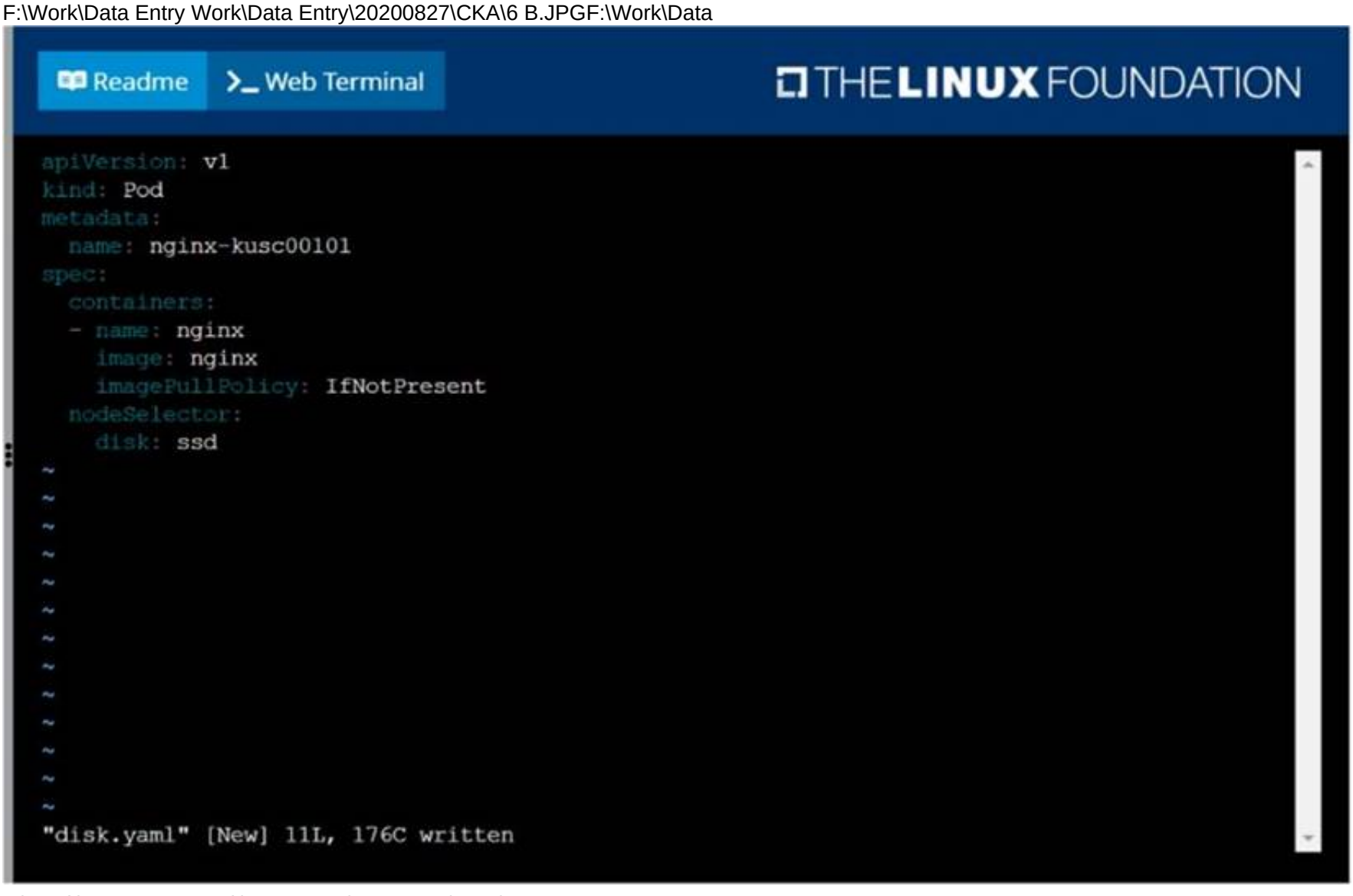
A. Mastered
B. Not Mastered

Answer: A

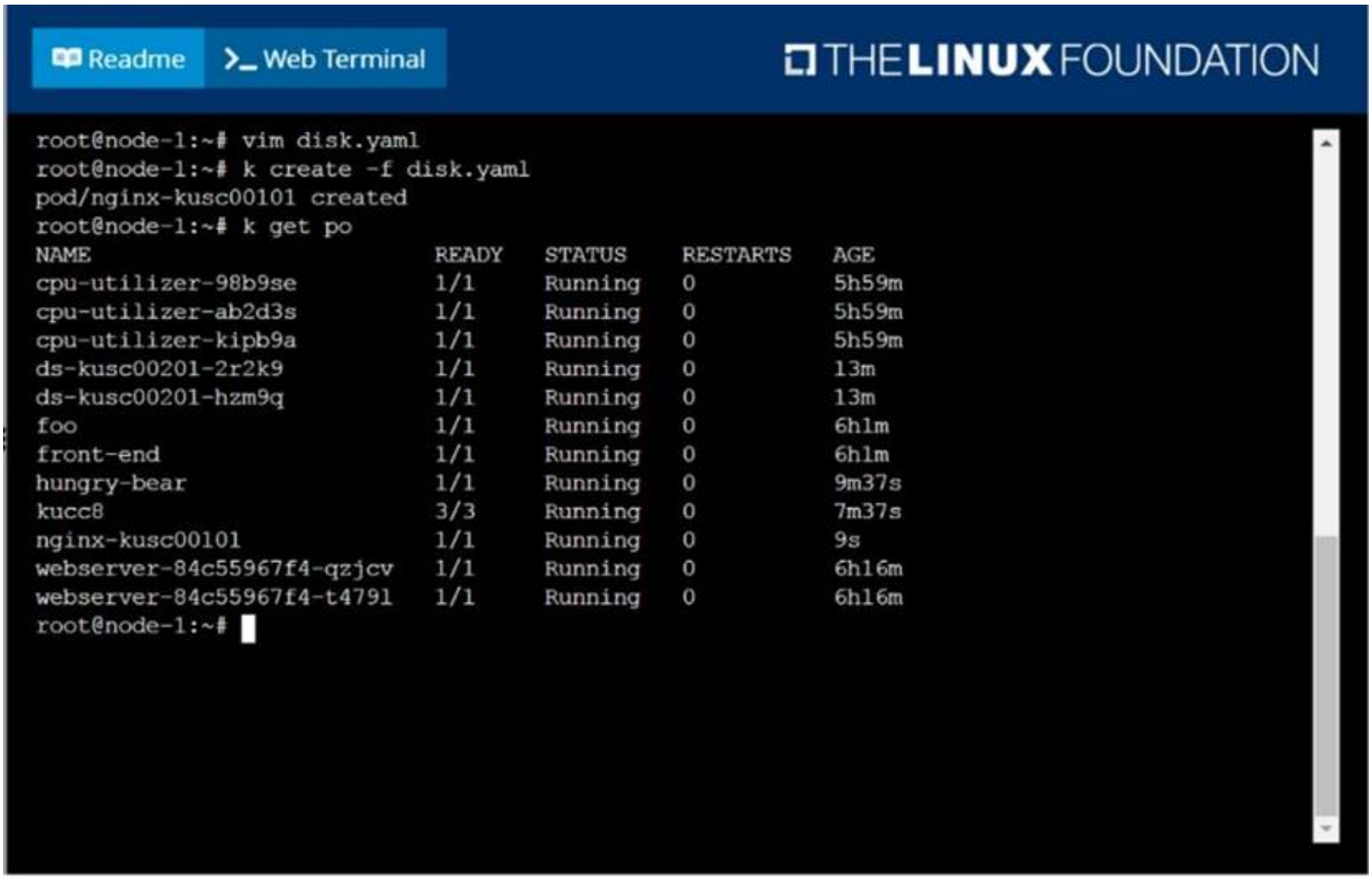
Explanation:
solution



F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 B.JPGF:\Work\Data



F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 D.JPG

NEW QUESTION 56
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 61
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 62
CORRECT TEXT
Create a nginx pod with label env=test in engineering namespace

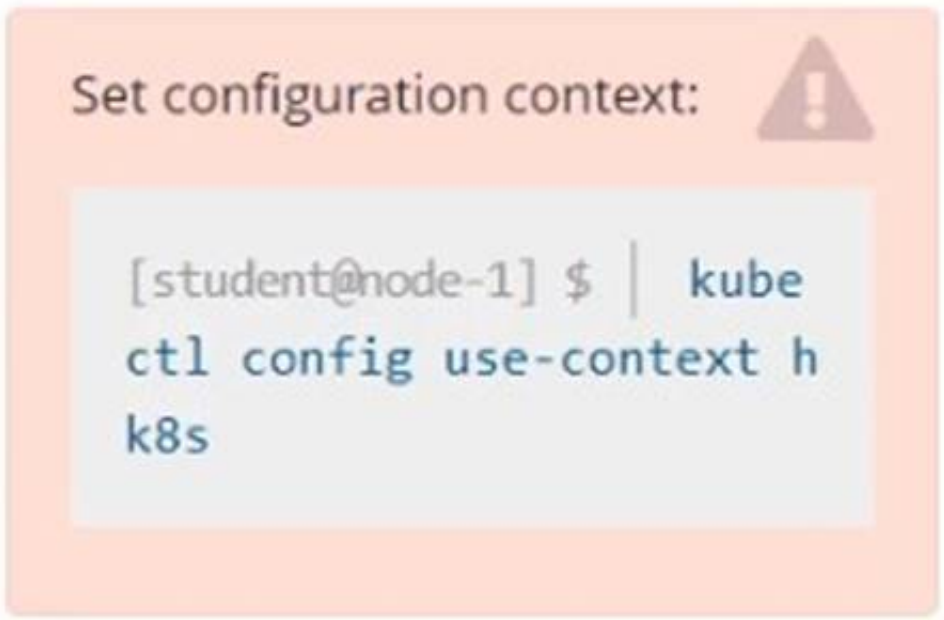
A. Mastered
B. Not Mastered

Answer: A

Explanation:
kubectl run nginx --image=nginx --restart=Never --labels=env=test
-- namespace=engineering --dry-run -o yaml > nginx-pod.yaml
kubectl run nginx --image=nginx --restart=Never --labels=env=test --
namespace=engineering --dry-run -o yaml | kubectl create -n engineering -f –
YAML File:
apiVersion: v1
kind: Pod
metadata:
name: nginx
namespace: engineering
labels:

env: test
spec:
containers:
- name: nginx
image: nginx
imagePullPolicy: IfNotPresent
restartPolicy: Never
kubectl create -f nginx-pod.yaml

NEW QUESTION 66
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 70
.....

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