

# Linux-Foundation

## Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program



### NEW QUESTION 1

CORRECT TEXT

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

### NEW QUESTION 2

CORRECT TEXT

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 3

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
# create
kubectl create -f pod-pvc.yaml
#edit
kubectl edit pvc pv-volume --record
```

#### NEW QUESTION 4

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

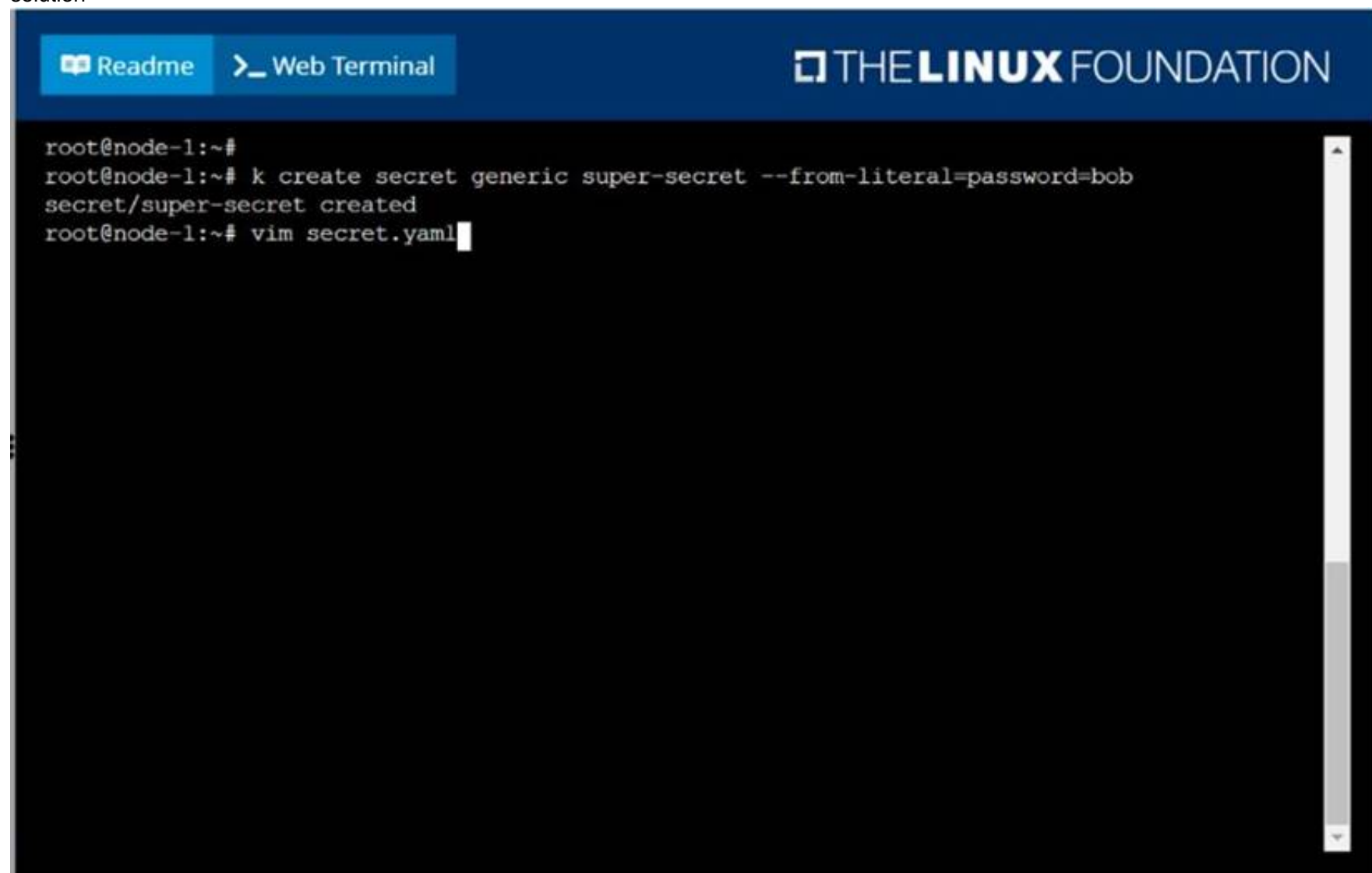
password as CONFIDENTIAL

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution



```
root@node-1:~#
root@node-1:~# k create secret generic super-secret --from-literal=password=bob
secret/super-secret created
root@node-1:~# vim secret.yaml
```

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

ReadmeWeb Terminal

THELINUXFOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-secrets-via-file
spec:
  containers:
  - name: redis
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/secrets"
  volumes:
  - name: foo
    secret:
      secretName: super-secret
~
~
~
~
~
~
~
~
:w
```

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

ReadmeWeb Terminal

THELINUXFOUNDATION

```
root@node-1:~# k create -f secret.yaml
pod/pod-secrets-via-file created
root@node-1:~# vim secret1.yaml
root@node-1:~# k create -f secret1.yaml
pod/pod-secrets-via-env created
root@node-1:~# k get po
NAME                                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se                 1/1     Running   0           6h25m
cpu-utilizer-ab2d3s                 1/1     Running   0           6h25m
cpu-utilizer-kipb9a                 1/1     Running   0           6h25m
ds-kusc00201-2r2k9                  1/1     Running   0           40m
ds-kusc00201-hzm9q                  1/1     Running   0           40m
foo                                  1/1     Running   0           6h28m
front-end                           1/1     Running   0           6h27m
hungry-bear                         1/1     Running   0           36m
kucc8                                3/3     Running   0           34m
nginx-app-848cfcf495-9prjh          1/1     Running   0           19m
nginx-app-848cfcf495-gl2kh          1/1     Running   0           19m
nginx-app-848cfcf495-pg2c8          1/1     Running   0           19m
nginx-kusc00101                     1/1     Running   0           26m
pod-secrets-via-env                 1/1     Running   0           4s
pod-secrets-via-file                1/1     Running   0           106s
webserver-84c55967f4-qzjcv          1/1     Running   0           6h43m
webserver-84c55967f4-t479l          1/1     Running   0           6h43m
root@node-1:~#
```

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

NEW QUESTION 5

CORRECT TEXT

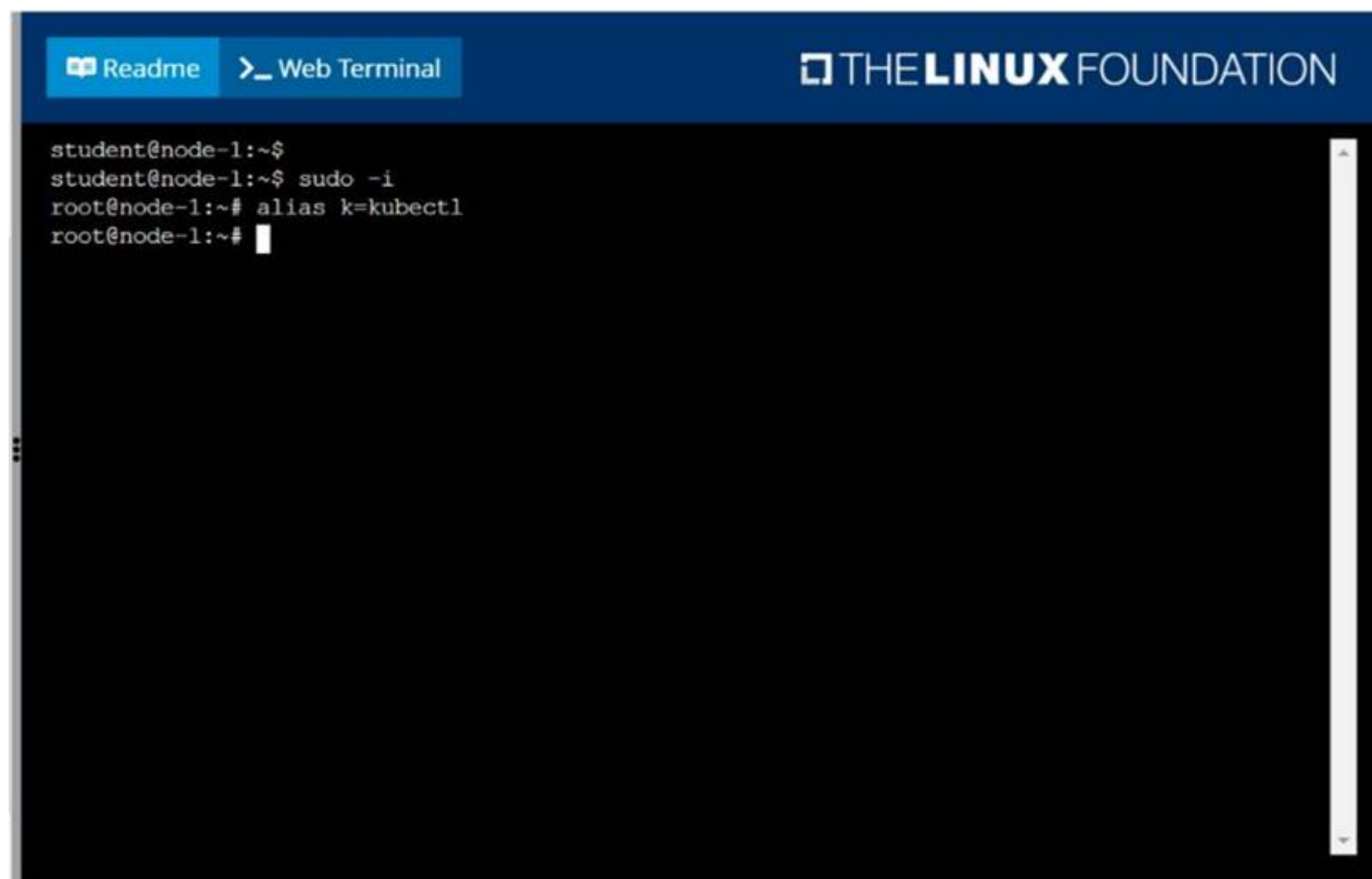
Monitor the logs of pod foo and:

- ? Extract log lines corresponding to error  
unable-to-access-website
- ? Write them to /opt/KULM00201/foo

- A. Mastered
- B. Not Mastered

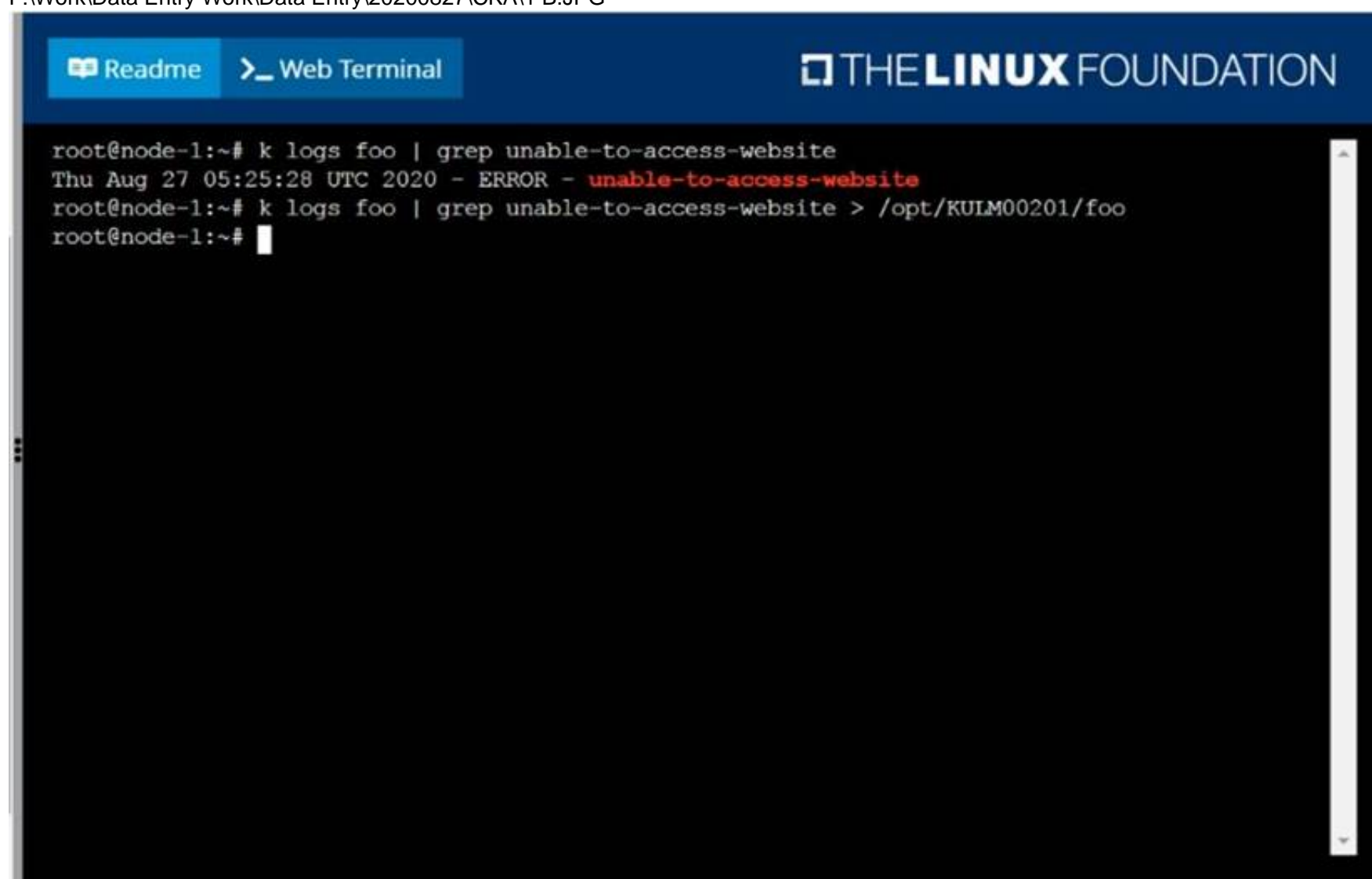
Answer: A

Explanation:  
solution



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

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



```
root@node-1:~# k logs foo | grep unable-to-access-website
Thu Aug 27 05:25:28 UTC 2020 - ERROR - unable-to-access-website
root@node-1:~# k logs foo | grep unable-to-access-website > /opt/KULM00201/foo
root@node-1:~#
```

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

NEW QUESTION 6  
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 7

CORRECT TEXT

Score: 7%



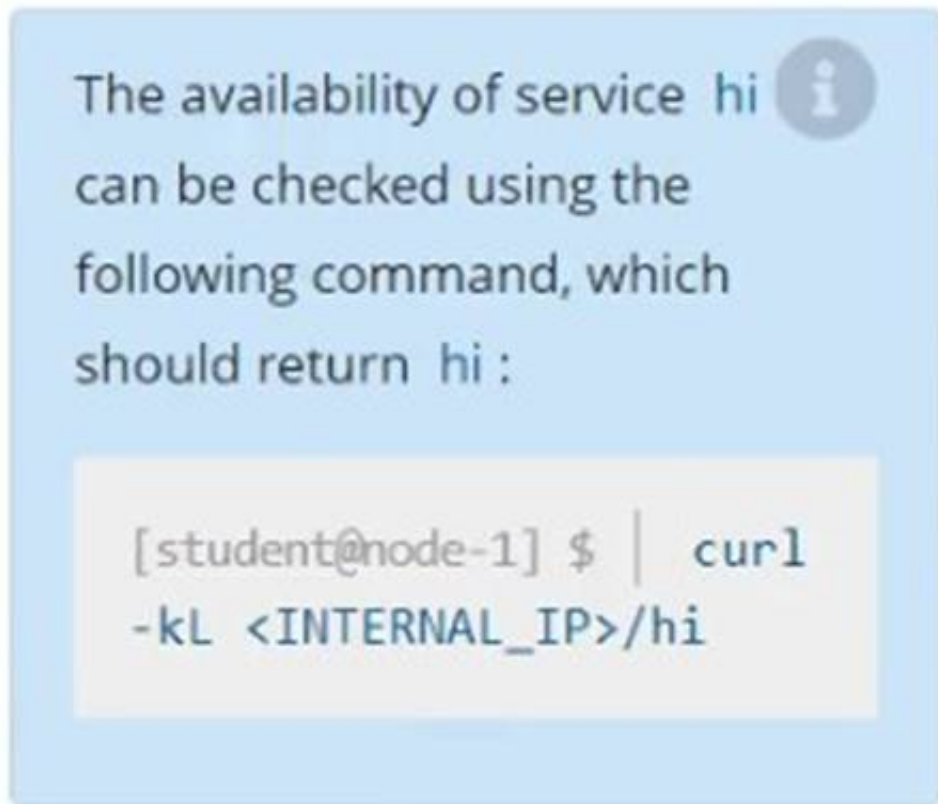
#### Task

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 8

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 9

CORRECT TEXT

Create a snapshot of the etcd instance running at <https://127.0.0.1:2379>, saving the snapshot to the file path /srv/data/etcd-snapshot.db.

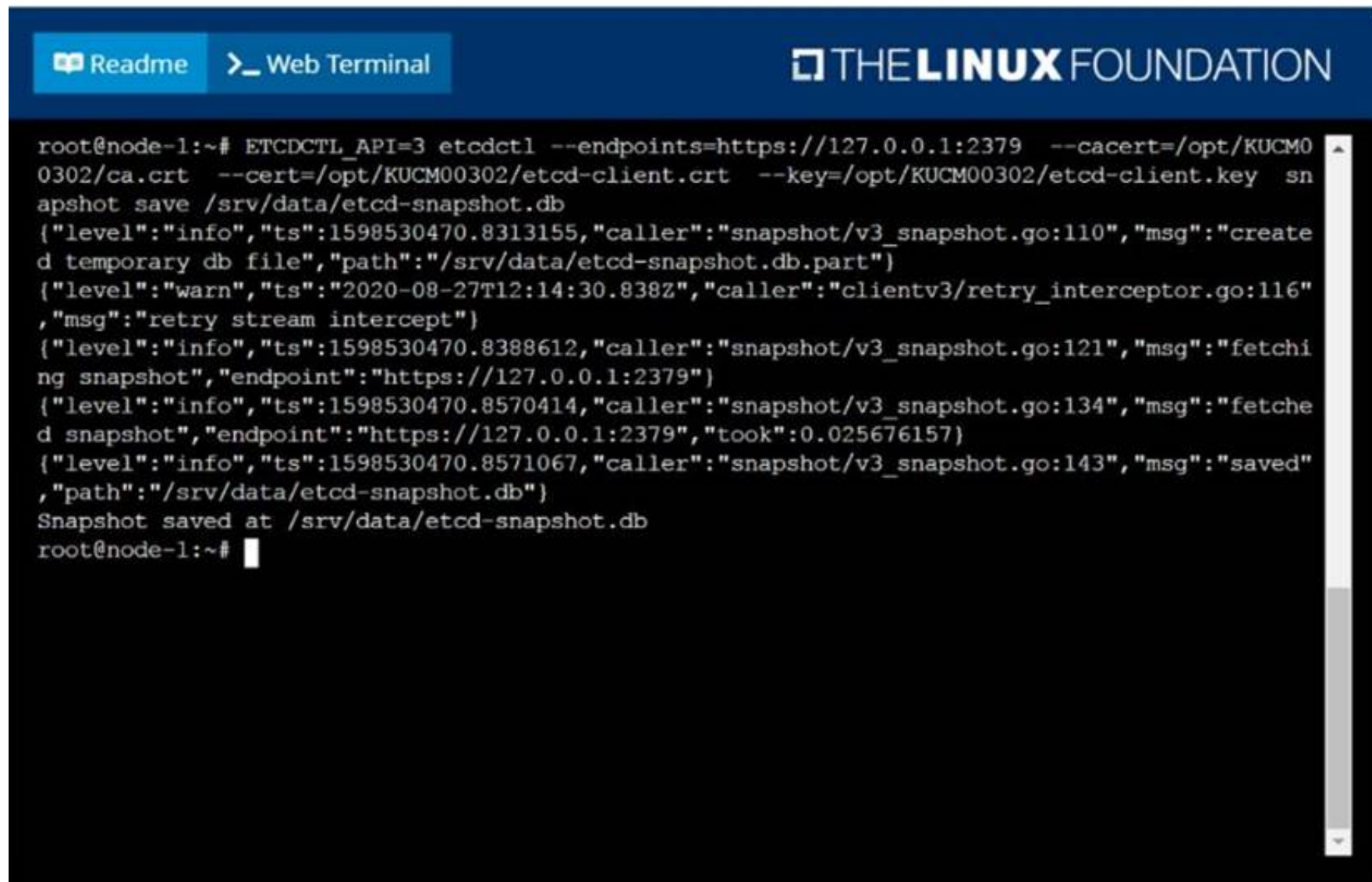
The following TLS certificates/key are supplied for connecting to the server with etcdctl:

- ? CA certificate: /opt/KUCM00302/ca.crt
- ? Client certificate: /opt/KUCM00302/etcd-client.crt
- ? Client key: Topt/KUCM00302/etcd-client.key

- A. Mastered  
B. Not Mastered

**Answer:** A

Explanation:  
 solution



```

root@node-1:~# ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/opt/KUCM00302/ca.crt --cert=/opt/KUCM00302/etcd-client.crt --key=/opt/KUCM00302/etcd-client.key snapshot 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":"fetching snapshot","endpoint":"https://127.0.0.1:2379"}
{"level":"info","ts":1598530470.8570414,"caller":"snapshot/v3_snapshot.go:134","msg":"fetched snapshot","endpoint":"https://127.0.0.1:2379","took":0.025676157}
{"level":"info","ts":1598530470.8571067,"caller":"snapshot/v3_snapshot.go:143","msg":"saved","path":"/srv/data/etcd-snapshot.db"}
Snapshot saved at /srv/data/etcd-snapshot.db
root@node-1:~#
    
```

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

#### NEW QUESTION 10

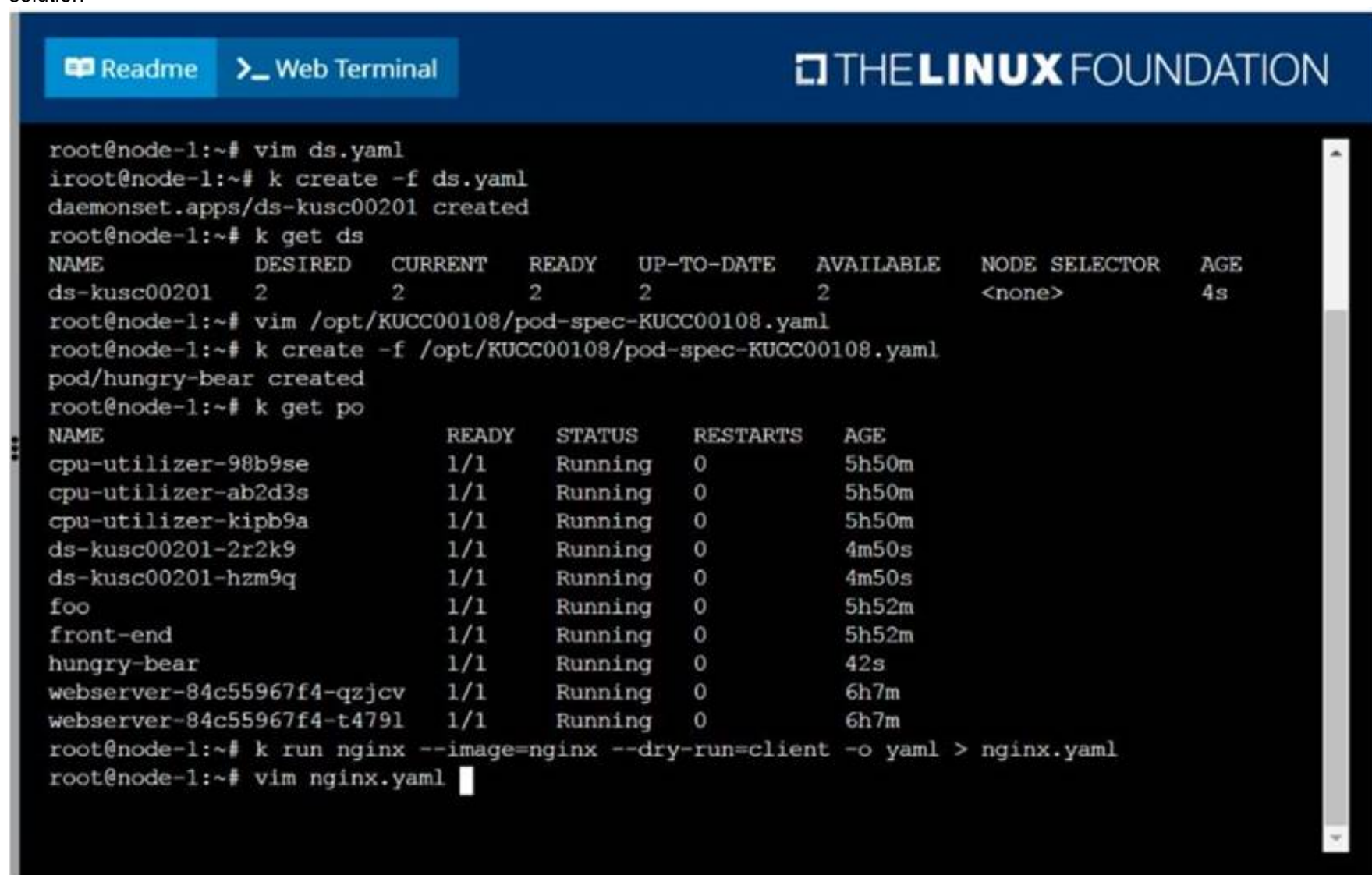
CORRECT TEXT

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

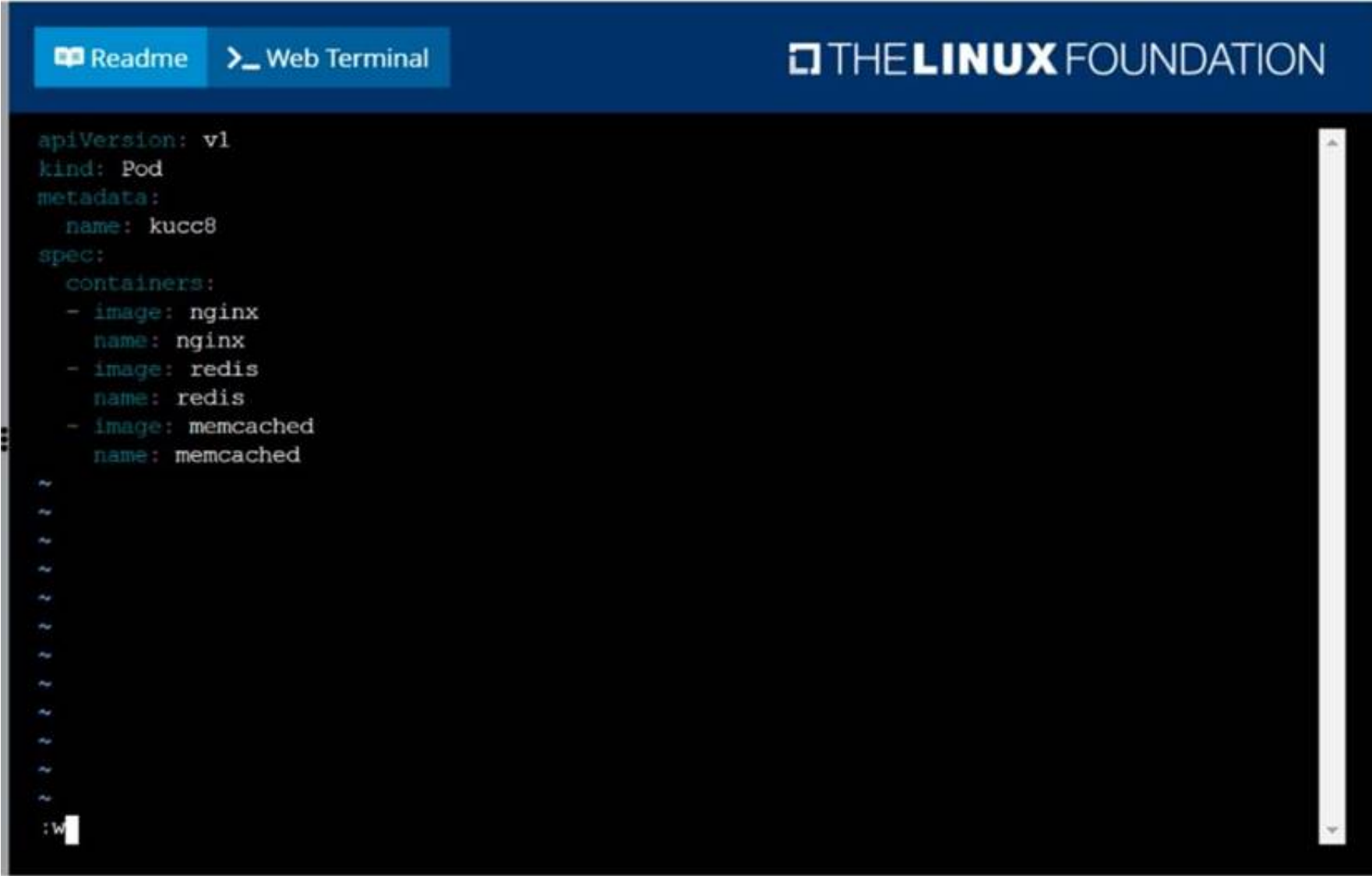


```

root@node-1:~# vim ds.yaml
root@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201    2         2         2       2            2           <none>          4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~# k get po
NAME                                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se                 1/1     Running   0          5h50m
cpu-utilizer-ab2d3s                 1/1     Running   0          5h50m
cpu-utilizer-kipb9a                 1/1     Running   0          5h50m
ds-kusc00201-2r2k9                  1/1     Running   0          4m50s
ds-kusc00201-hzm9q                  1/1     Running   0          4m50s
foo                                 1/1     Running   0          5h52m
front-end                           1/1     Running   0          5h52m
hungry-bear                         1/1     Running   0          42s
webserver-84c55967f4-qzjcv          1/1     Running   0          6h7m
webserver-84c55967f4-t479l          1/1     Running   0          6h7m
root@node-1:~# k run nginx --image=nginx --dry-run=client -o yaml > nginx.yaml
root@node-1:~# vim nginx.yaml
    
```



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

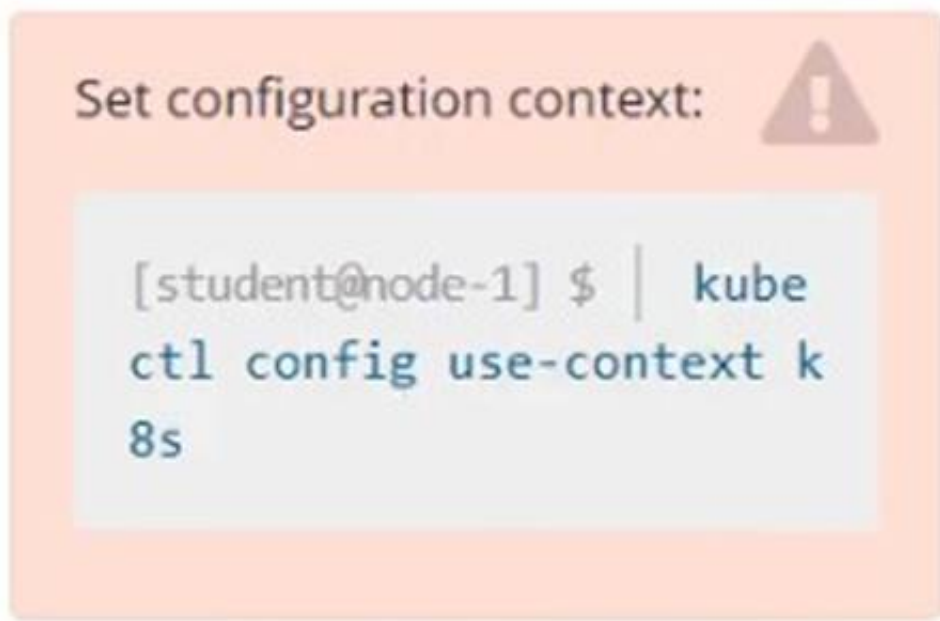


F:\Work\Data Entry Work\Data Entry\20200827\CKA\5 C.JPG



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

NEW QUESTION 10  
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
  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 12

CORRECT TEXT

Create a file:

/opt/KUCC00302/kucc00302.txt that lists all pods that implement service baz in namespace development.

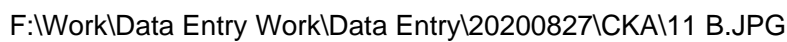
The format of the file should be one pod name per line.

- A. Mastered
- B. Not Mastered

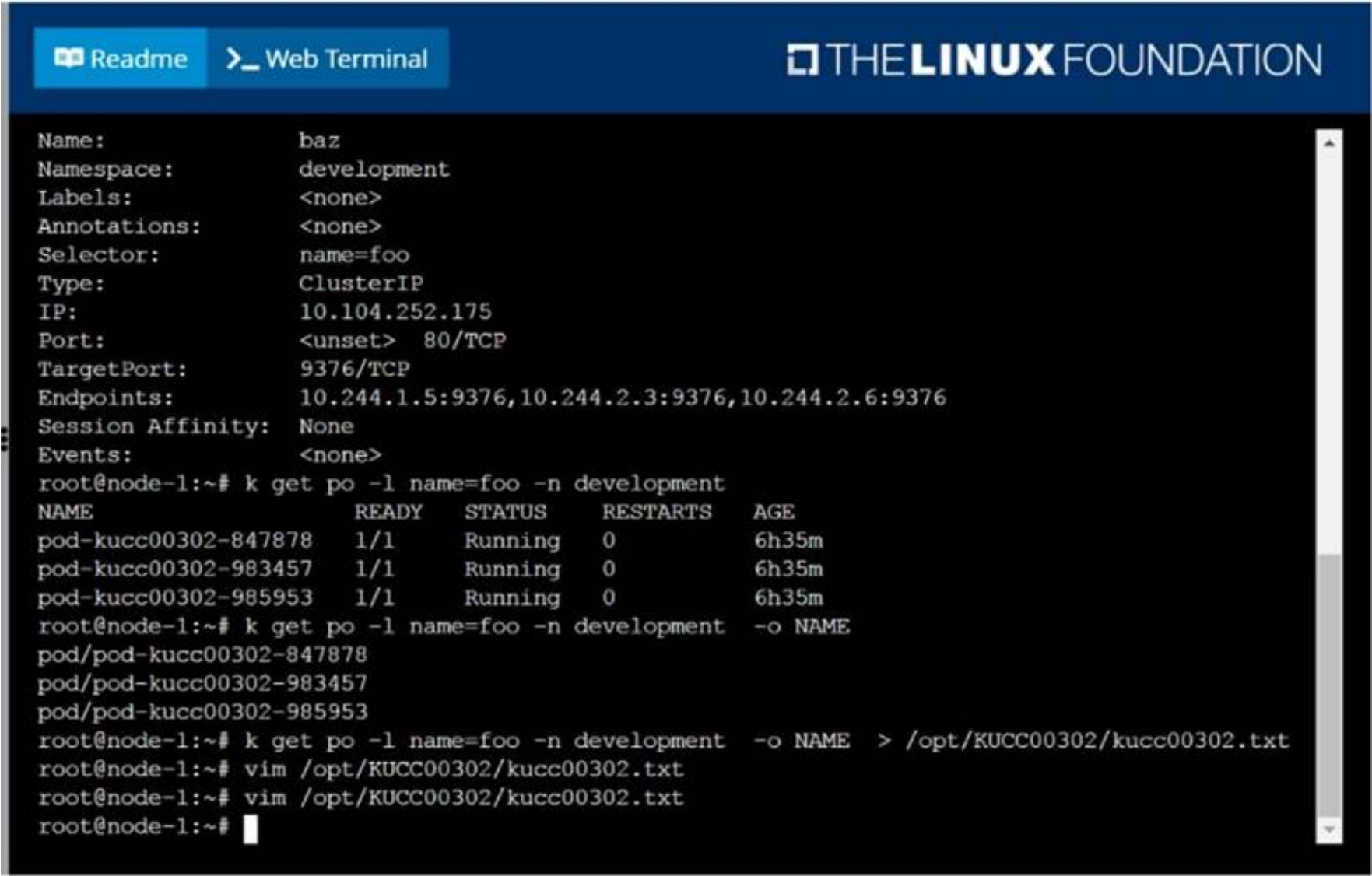
**Answer:** A

**Explanation:**

solution

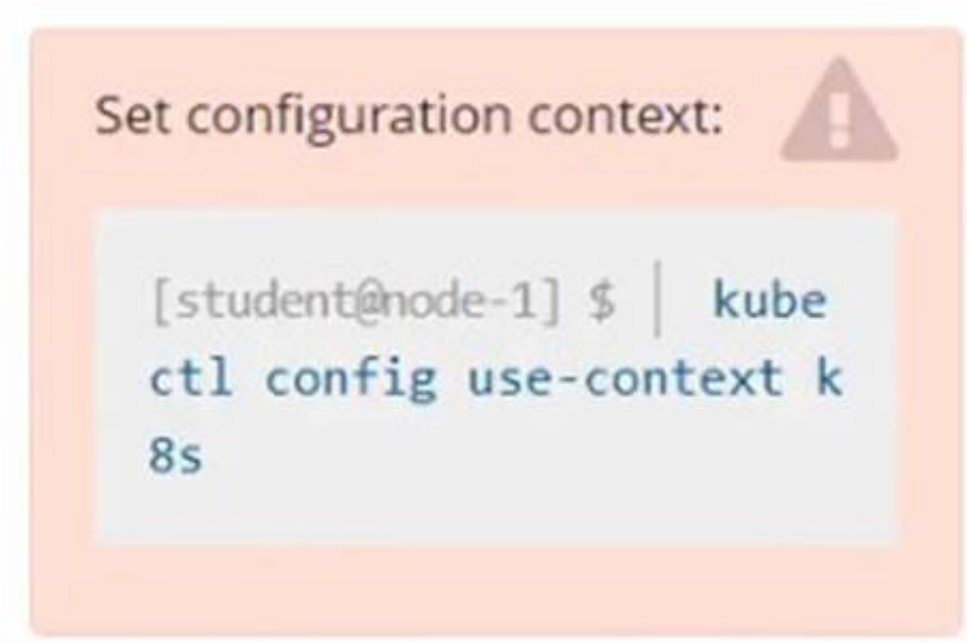






F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 D.JPG

NEW QUESTION 17  
 CORRECT TEXT  
 Score: 4%



Task  
 Scale the deployment presentation to 6 pods.  
 A. Mastered  
 B. Not Mastered

Answer: A  
 Explanation:  
 Solution:  
 kubectl get deployment  
 kubectl scale deployment.apps/presentation --replicas=6

NEW QUESTION 19  
 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 24

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.

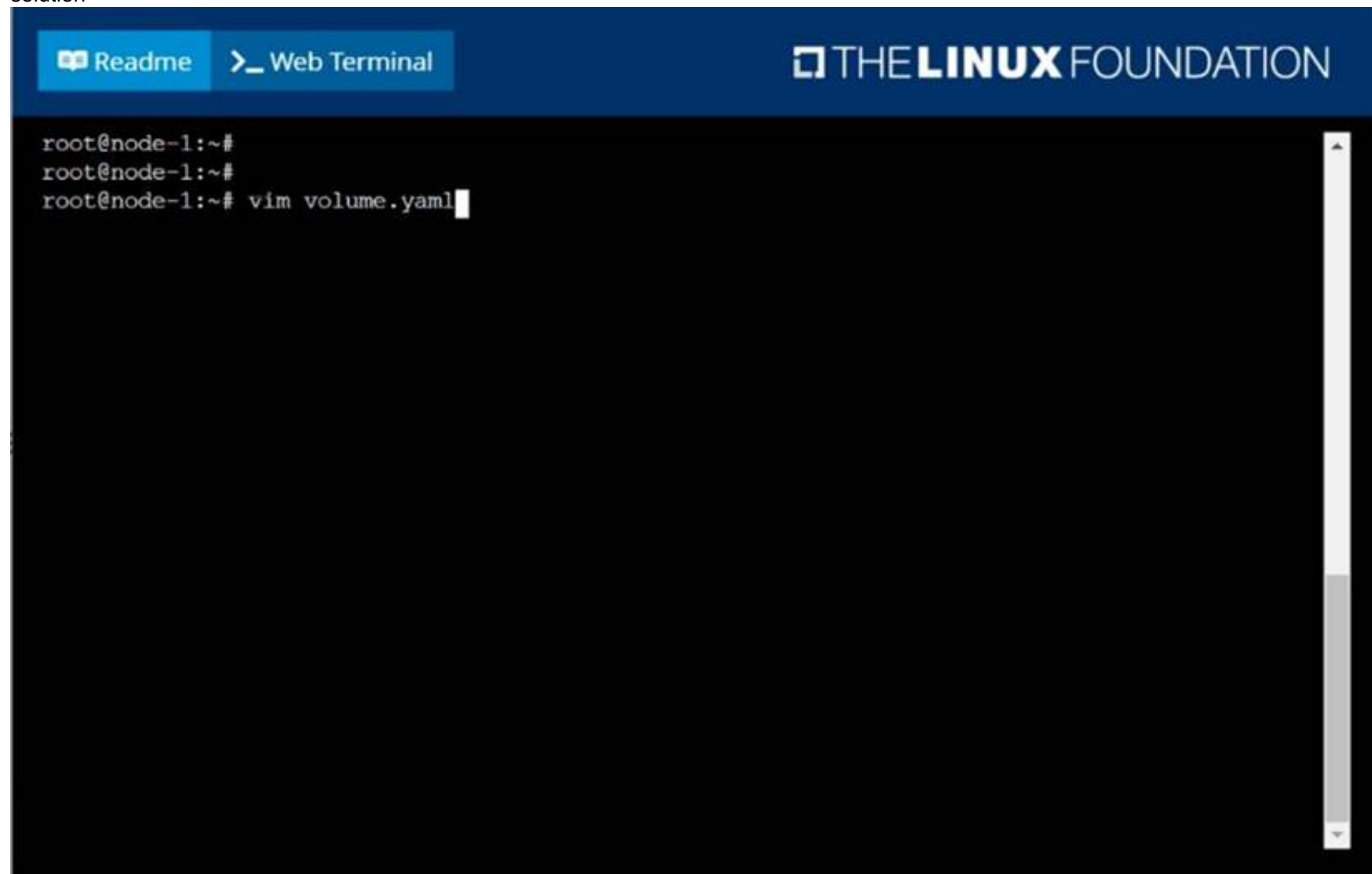
A. Mastered

B. Not Mastered

**Answer:** A

**Explanation:**

solution



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

ReadmeWeb Terminal

THE **LINUX** FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: non-persistent-redis
  namespace: staging
spec:
  containers:
  - name: redis
    image: redis
    volumeMounts:
    - name: cache-control
      mountPath: /data/redis
  volumes:
  - name: cache-control
    emptyDir: {}

~
~
~
~
~
~
~
~
~
~
:w
```

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

ReadmeWeb Terminal

THE **LINUX** FOUNDATION

```
root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml
root@node-1:~# k create -f volume.yaml
pod/non-persistent-redis created
root@node-1:~# k get po -n staging
NAME                READY   STATUS    RESTARTS   AGE
non-persistent-redis 1/1     Running   0           6s
root@node-1:~#
```

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

NEW QUESTION 26

CORRECT TEXT

List the nginx pod with custom columns POD\_NAME and POD\_STATUS

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubect! get po -o=custom-columns="POD\_NAME:.metadata.name, POD\_STATUS:.status.containerStatuses[.state]"

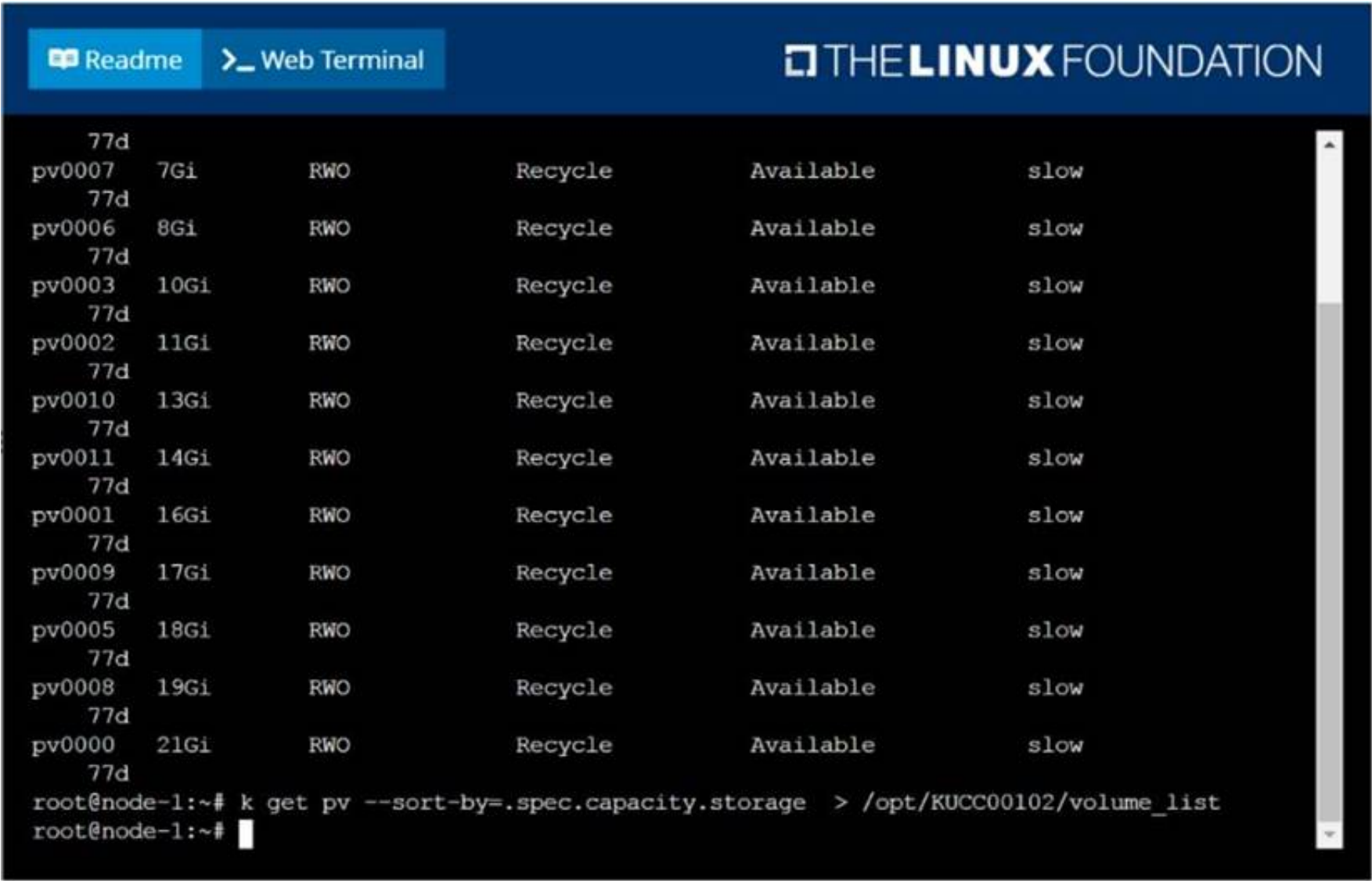
NEW QUESTION 31

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



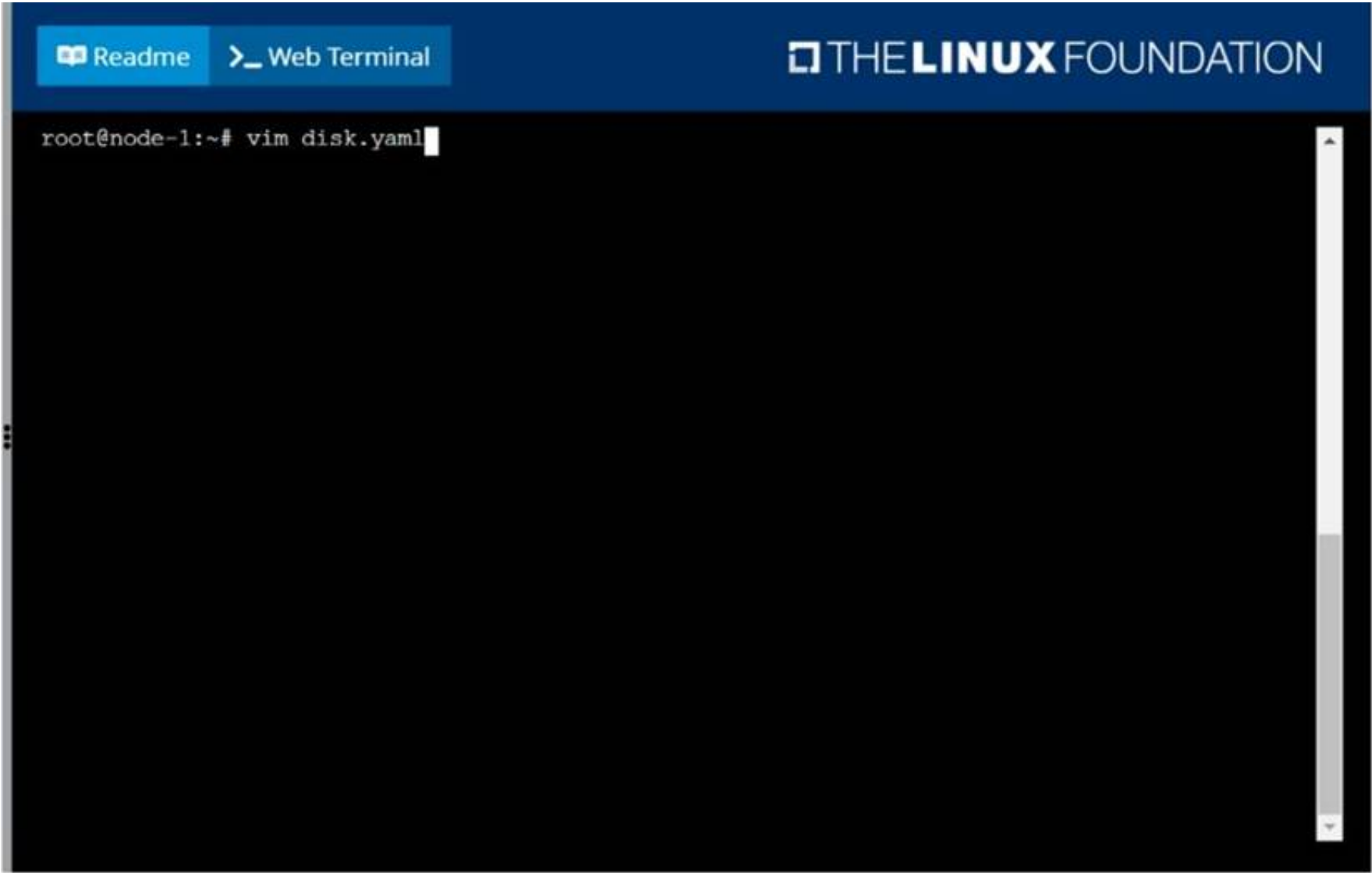
F:\Work\Data Entry Work\Data Entry\20200827\CKA\2 C.JPG

NEW QUESTION 35  
CORRECT TEXT  
Schedule a pod as follows:  
? Name: nginx-kusc00101  
? Image: nginx  
? Node selector: disk=ssd

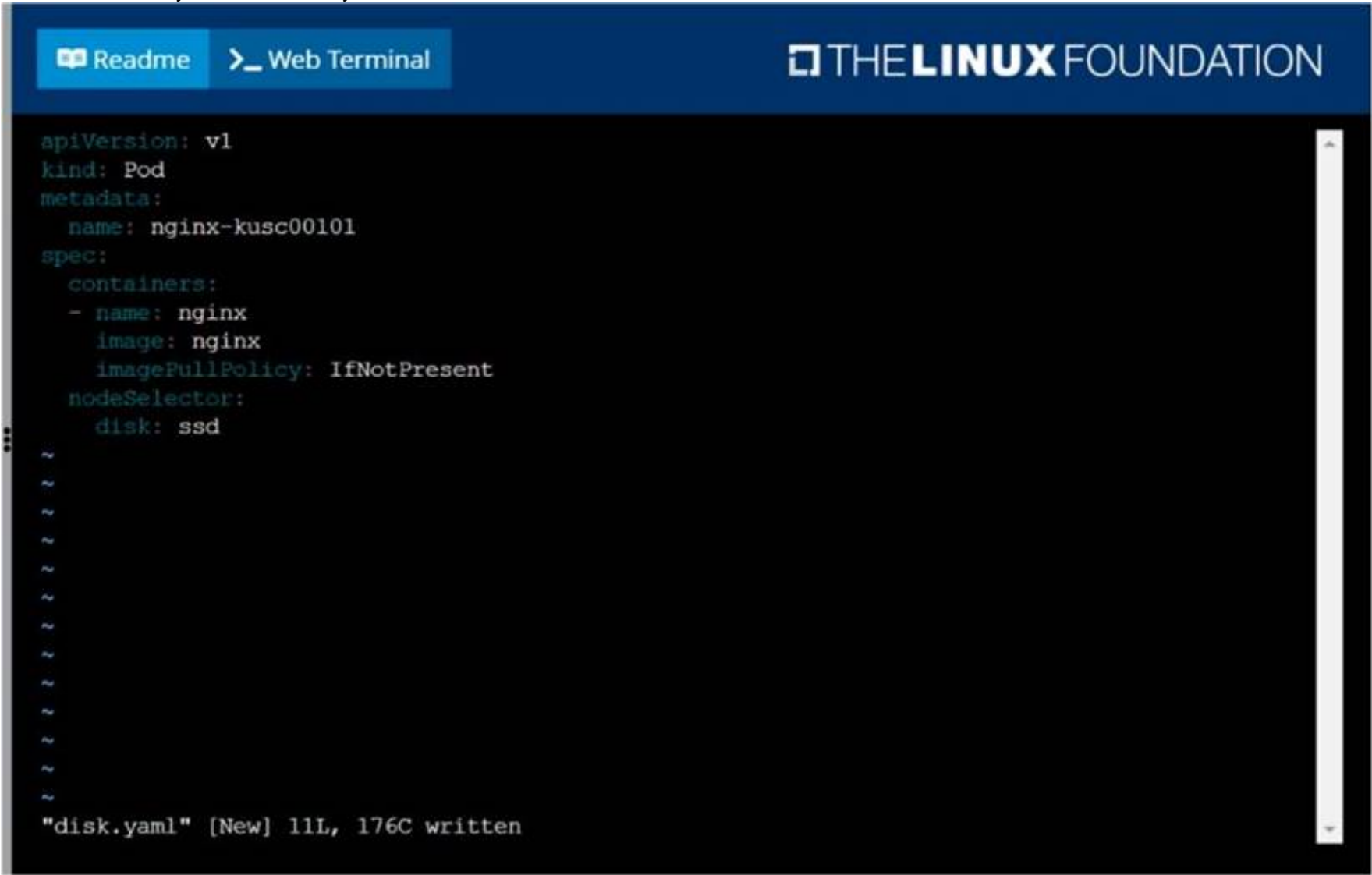
- A. Mastered
- B. Not Mastered

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



ReadmeWeb Terminal

THE LINUX FOUNDATION

```
root@node-1:~# vim disk.yaml
root@node-1:~# k create -f disk.yaml
pod/nginx-kusc00101 created
root@node-1:~# k get po
NAME                                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se                1/1     Running   0           5h59m
cpu-utilizer-ab2d3s                1/1     Running   0           5h59m
cpu-utilizer-kipb9a                1/1     Running   0           5h59m
ds-kusc00201-2r2k9                 1/1     Running   0           13m
ds-kusc00201-hzm9q                 1/1     Running   0           13m
foo                                1/1     Running   0           6h1m
front-end                          1/1     Running   0           6h1m
hungry-bear                        1/1     Running   0           9m37s
kucc8                               3/3     Running   0           7m37s
nginx-kusc00101                    1/1     Running   0           9s
webserver-84c55967f4-qzjcv         1/1     Running   0           6h16m
webserver-84c55967f4-t479l         1/1     Running   0           6h16m
root@node-1:~#
```

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

### NEW QUESTION 37

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 41

CORRECT TEXT

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 45

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 48

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 53

CORRECT TEXT

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
```

kubect! describe po nginx | grep value1

#### NEW QUESTION 56

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:

```
kubect! run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run
-o yam! > nginx-prodpod.yam! Now, edit nginx-prod-pod.yam! file and remove entries like "creationTimestamp: null" "dnsPolicy: ClusterFirst"
vim nginx-prod-pod.yam!
apiVersion: v1
kind: Pod
metadata:
labels:
env: prod
name: nginx-prod
spec:
containers:
- image: nginx
name: nginx-prod
restartPolicy: Always
# kubect! create -f nginx-prod-pod.yam!
kubect! run --generator=run-pod/v1 --image=nginx --
labels=env=dev nginx-dev --dry-run -o yam! > nginx-dev-pod.yam!
apiVersion: v1
kind: Pod
metadata:
labels:
env: dev
name: nginx-dev
spec:
containers:
- image: nginx
name: nginx-dev
restartPolicy: Always
# kubect! create -f nginx-prod-dev.yam!
Verify :
kubect! get po --show-labels
kubect! get po -l env=prod
kubect! get po -l env=dev
```

#### NEW QUESTION 61

.....

## Thank You for Trying Our Product

### We offer two products:

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

2nd - Questions 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](#)**