

## CKA Dumps

### Certified Kubernetes Administrator (CKA) Program

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



**NEW QUESTION 1**

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 2**

Create a pod as follows:

- > Name:mongo
- > Using Image:mongo
- > In anew Kubernetes namespacenamed:my-website

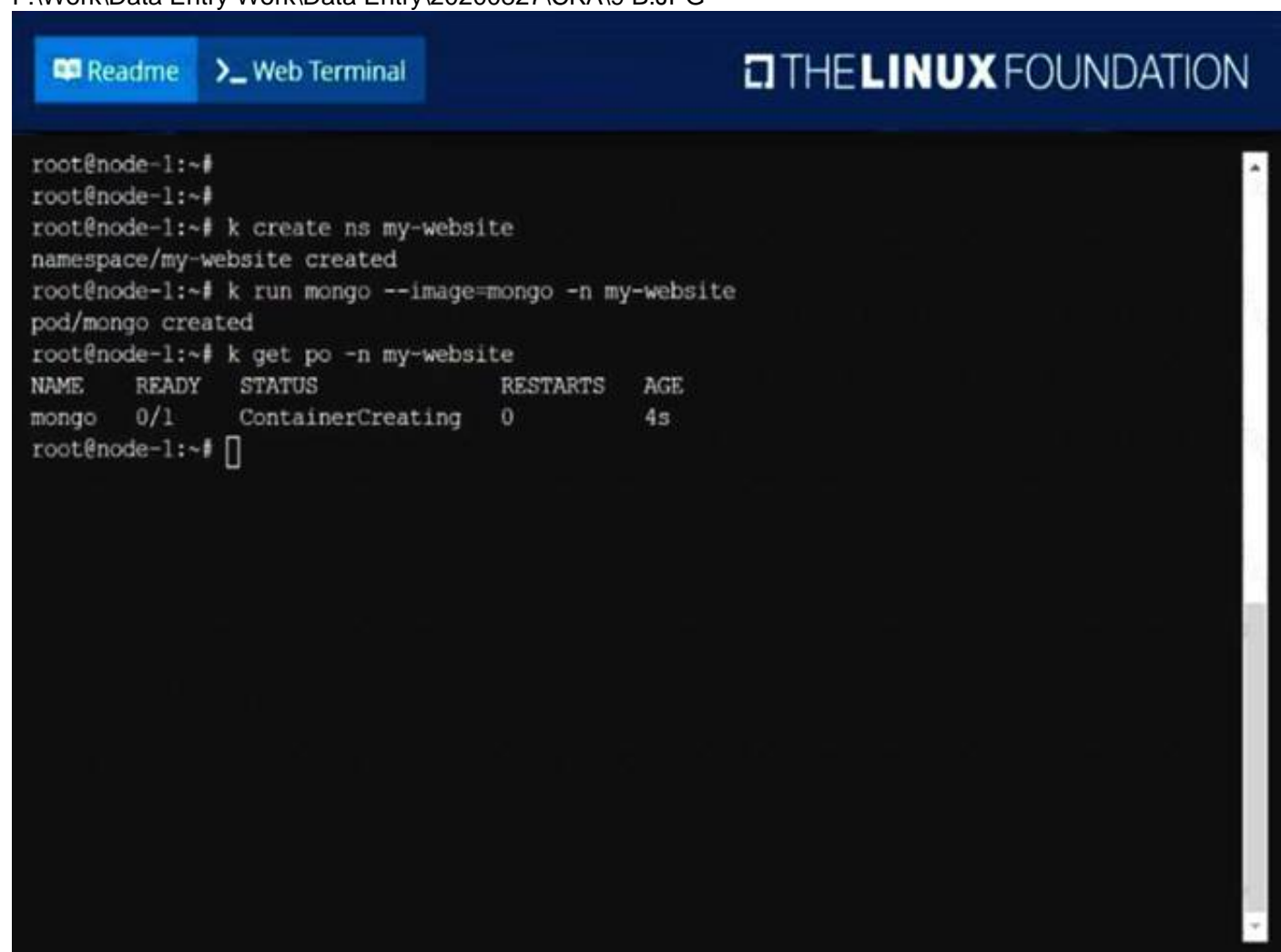
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

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



The screenshot shows a terminal window with a dark background and white text. At the top, there is a blue header bar with the text "THE LINUX FOUNDATION" and two buttons: "Readme" and "Web Terminal". The terminal content shows the following commands and output:

```
root@node-1:~#  
root@node-1:~#  
root@node-1:~# k create ns my-website  
namespace/my-website created  
root@node-1:~# k run mongo --image=mongo -n my-website  
pod/mongo created  
root@node-1:~# k get po -n my-website  
NAME      READY   STATUS             RESTARTS   AGE  
mongo     0/1     ContainerCreating   0           4s  
root@node-1:~#
```

**NEW QUESTION 3**

Create a deployment as follows:

- > Name:nginx-app
- > Using containernginxwithversion 1.11.10-alpine
- > The deployment should contain3replicas

Next, deploy the application with newversion1.11.13-alpine, byperforming a rolling update.

Finally, rollback that update to theprevious version1.11.10-alpine.

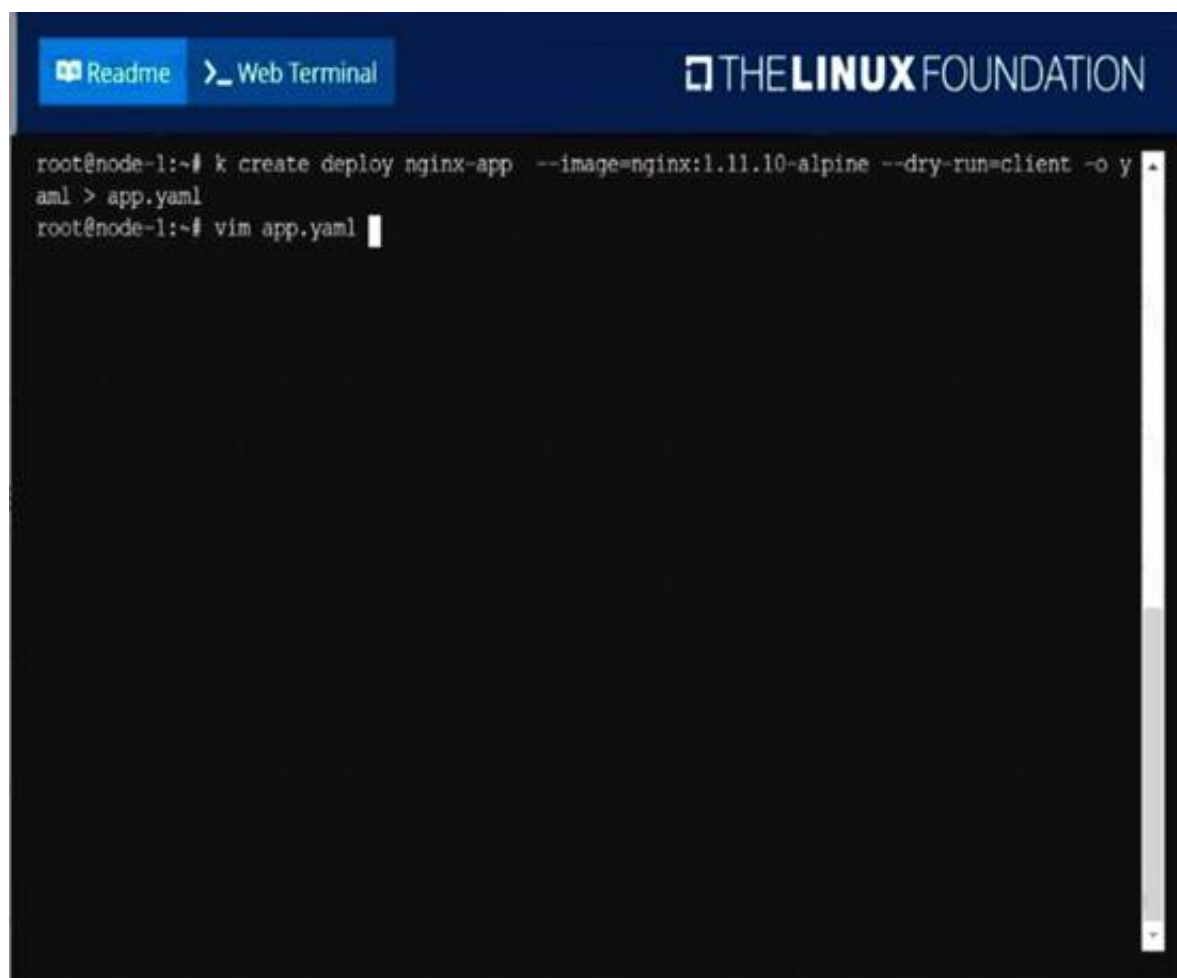
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

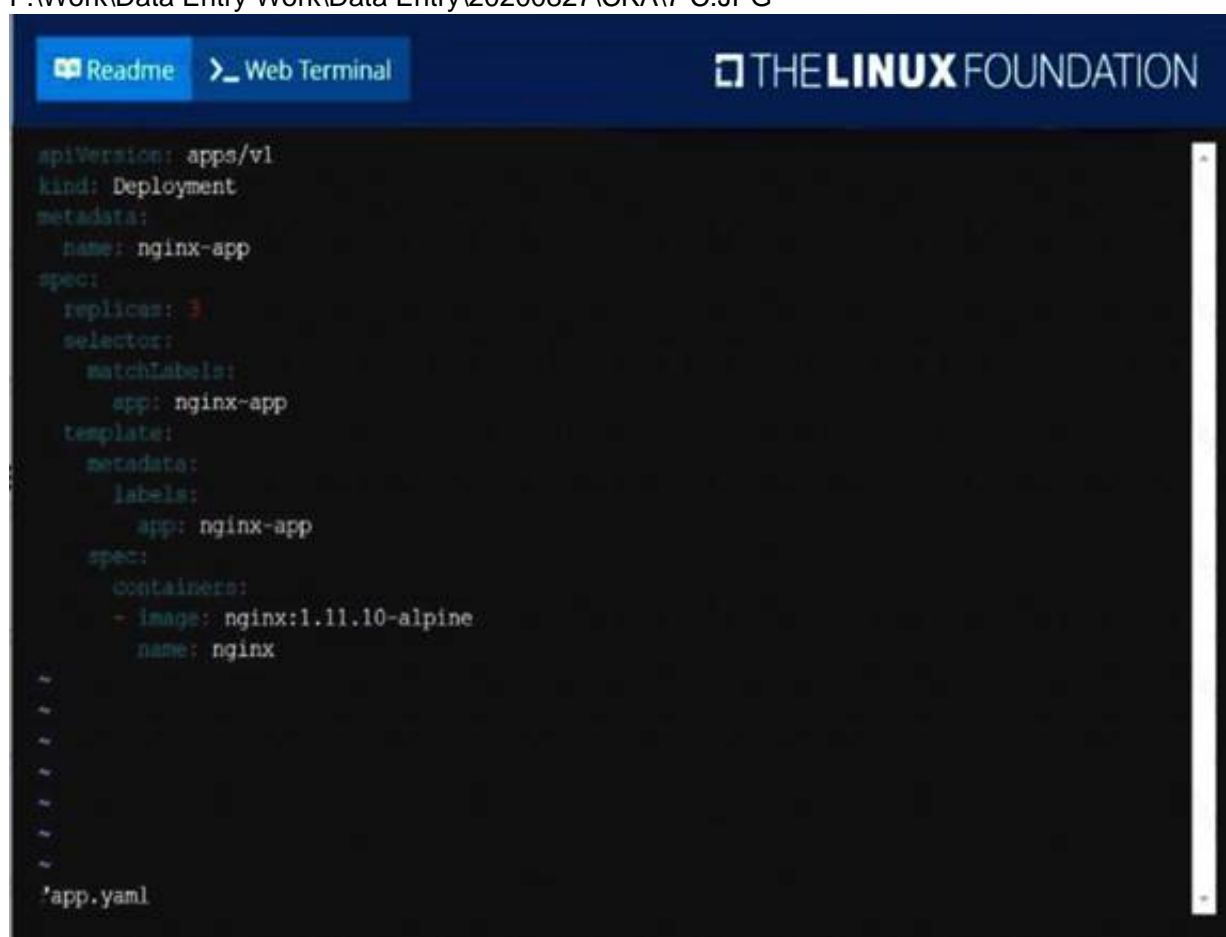
solution

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



```
root@node-1:~# k create deploy nginx-app --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
```

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



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      labels:
        app: nginx-app
    spec:
      containers:
      - image: nginx:1.11.10-alpine
        name: nginx
~
~
~
~
~
~
/app.yaml
```

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

ReadmeWeb Terminal

THE LINUX FOUNDATION

```
root@node-1:~# k create deploy nginx-app --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
root@node-1:~# k create -f app.yaml
deployment.apps/nginx-app created
root@node-1:~#
root@node-1:~#
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine --record
deployment.apps/nginx-app image updated
root@node-1:~# k rollout undo deploy nginx-app
deployment.apps/nginx-app rolled back
root@node-1:~#
```

NEW QUESTION 4

Create a pod namedkucc8with a single app container for each of the following images running inside(there may be between 1 and 4images specified): nginx + redis + memcached.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

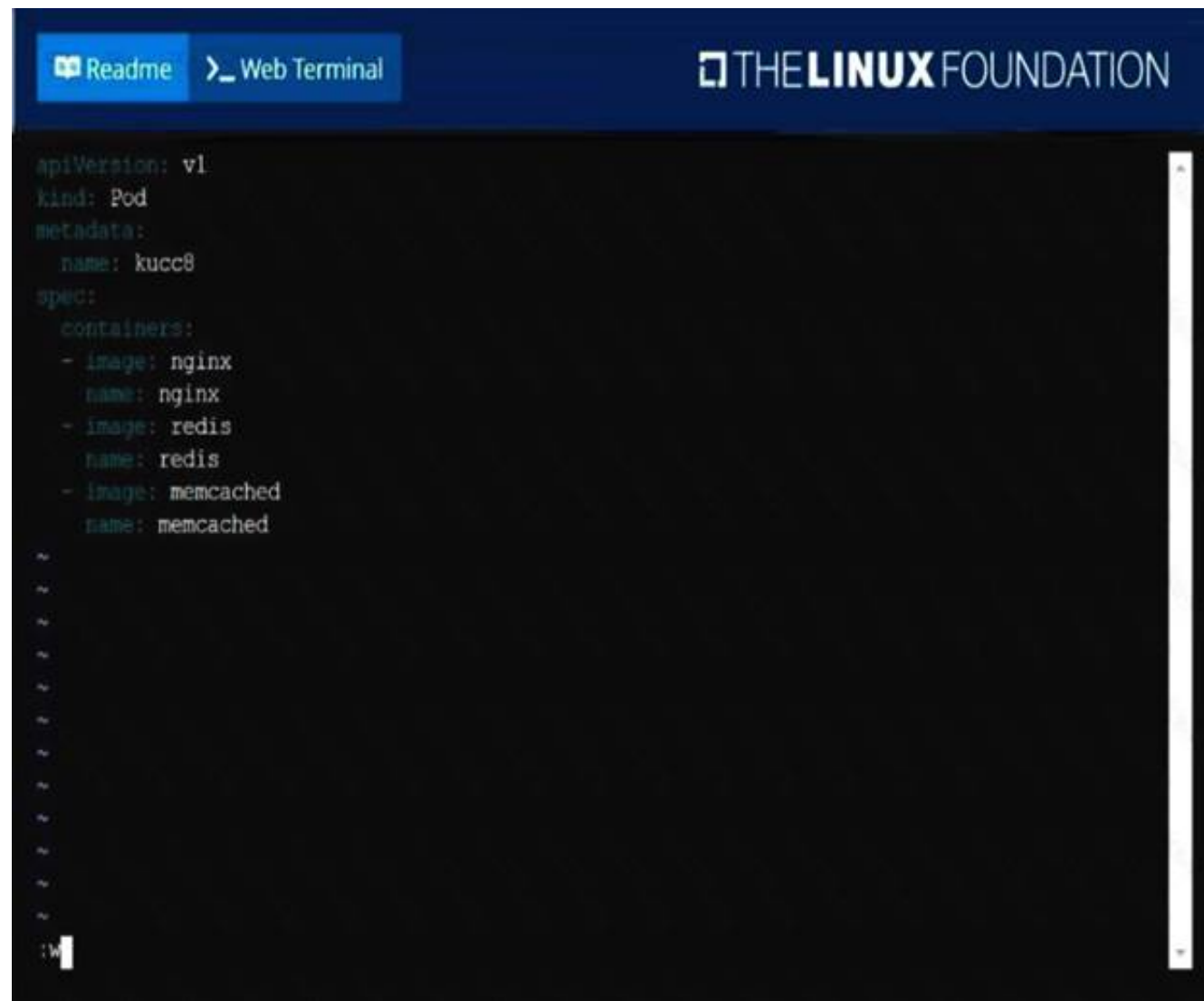
solution  
F:\Work\Data Entry Work\Data Entry\20200827\CKA\5 B.JPG

ReadmeWeb Terminal

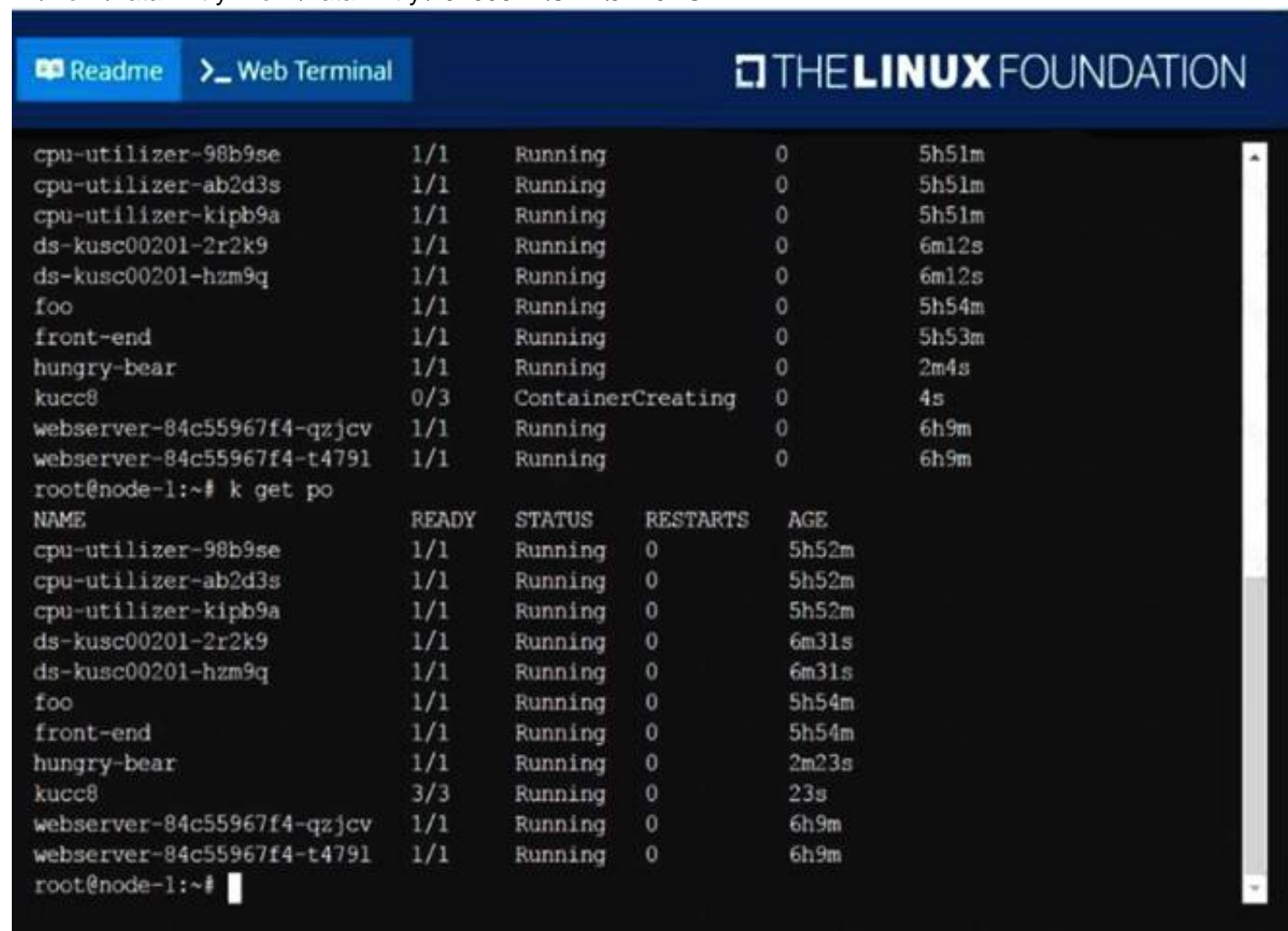
THE LINUX FOUNDATION

```
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 C.JPG



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



## NEW QUESTION 5

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

### Explanation:

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

## NEW QUESTION 6

Create a pod as follows:

- > Name:non-persistent-redis
- > container Image:redis
- > Volume with name:cache-control
- >



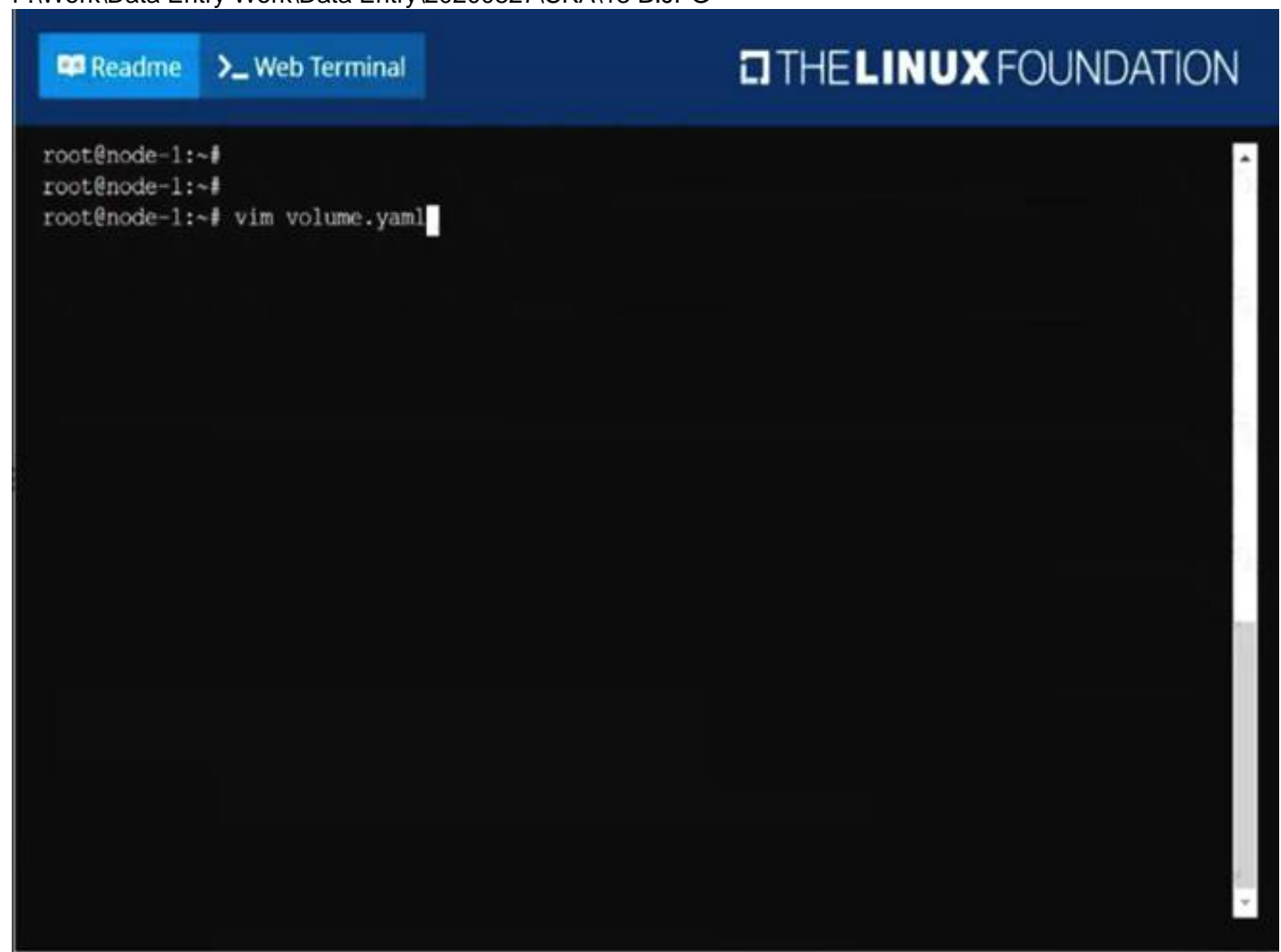
Mount path:/data/redis  
The pod should launch in thestagingnamespace and the volumemust notbe persistent.

- A. Mastered
- B. Not Mastered

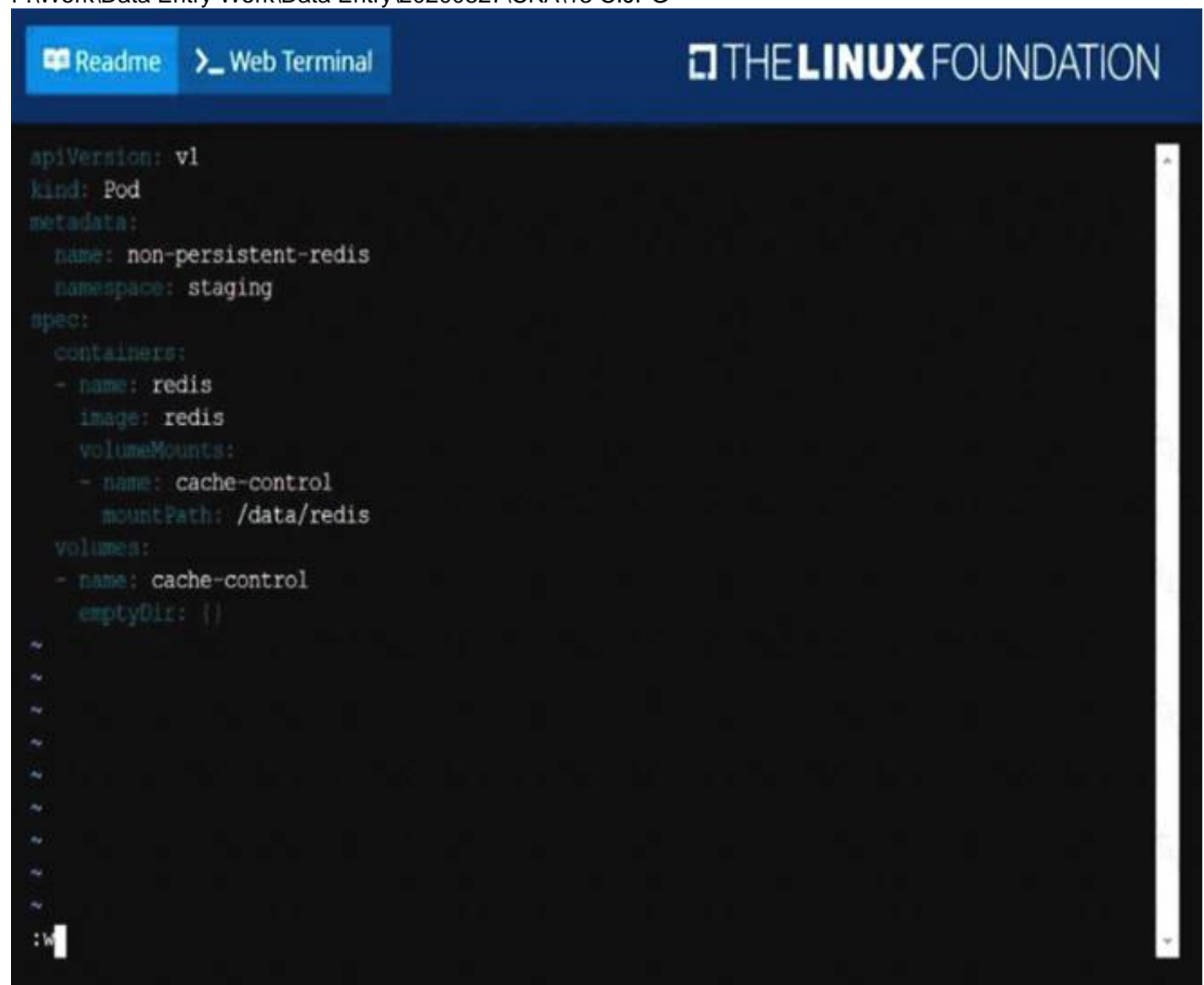
**Answer:** A

**Explanation:**

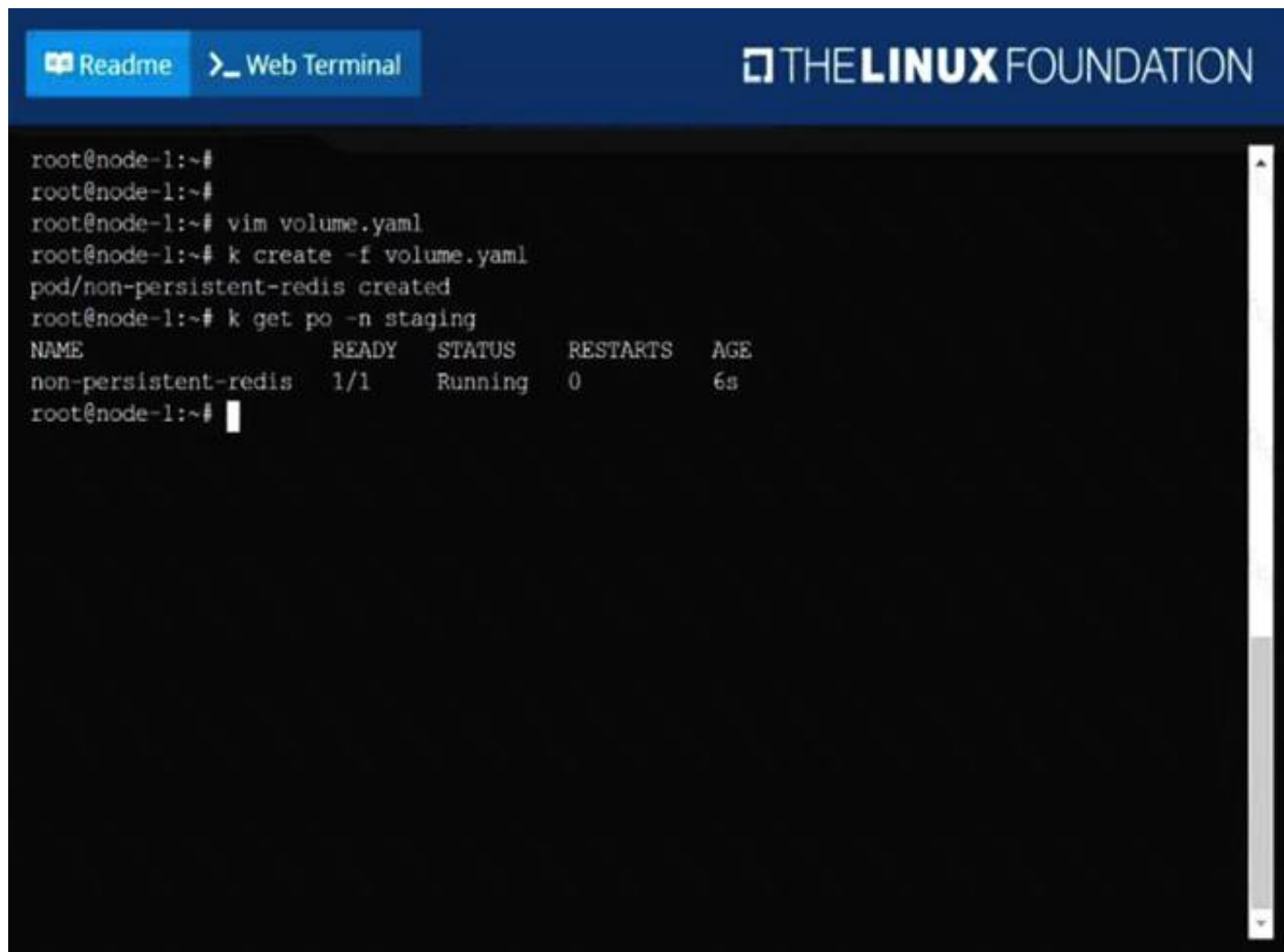
solution  
F:\Work\Data Entry Work\Data Entry\20200827\CKA\13 B.JPG



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



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



```
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:~#
```

#### NEW QUESTION 7

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 8

Create a file:

/opt/KUCC00302/kucc00302.txtthatlists all pods that implement servicebazin namespacedevelopment.

The format of the file should be onepod name per line.

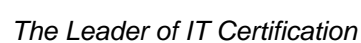
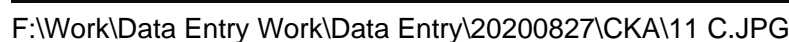
- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

solution

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





From the pod labelname=cpu-utilizer, find podsrunning high CPU workloads and write the name of the pod consumingmost CPU to thefile/opt/KUTR00102/KUTR00102.txt(which already exists).

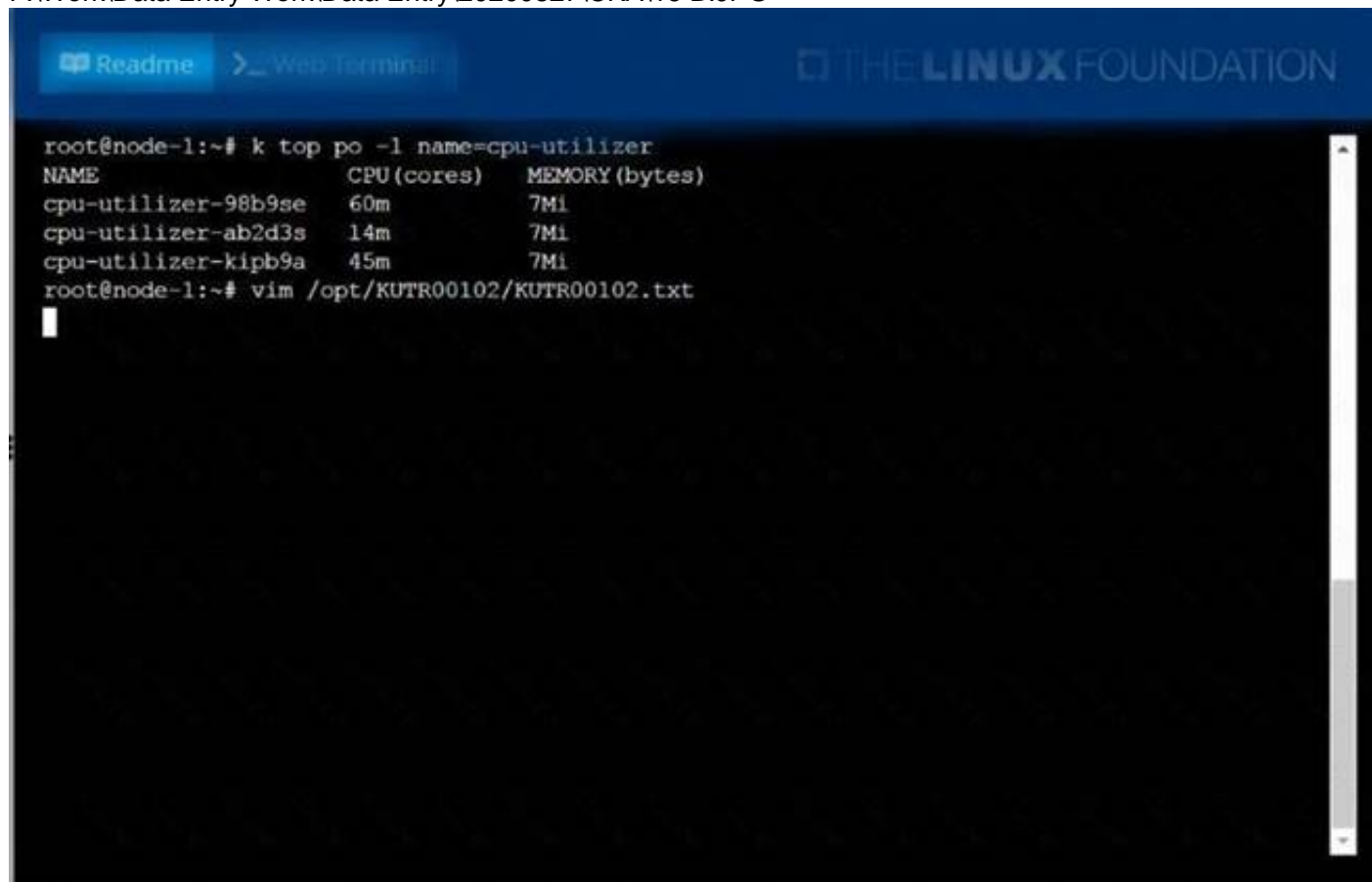
- A. Mastered  
B. Not Mastered

**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



Perform the following tasks:

- Add an init container tohungry-bear(which has been defined in spec file /opt/KUCC00108/pod-spec-KUCC00108.yaml)
- The init container should createan empty file named/workdir/calm.txt
- If/workdir/calm.txtis notdetected, the pod should exit
- Once the spec file has beenupdatedwith the init containerdefinition, the pod should becreated

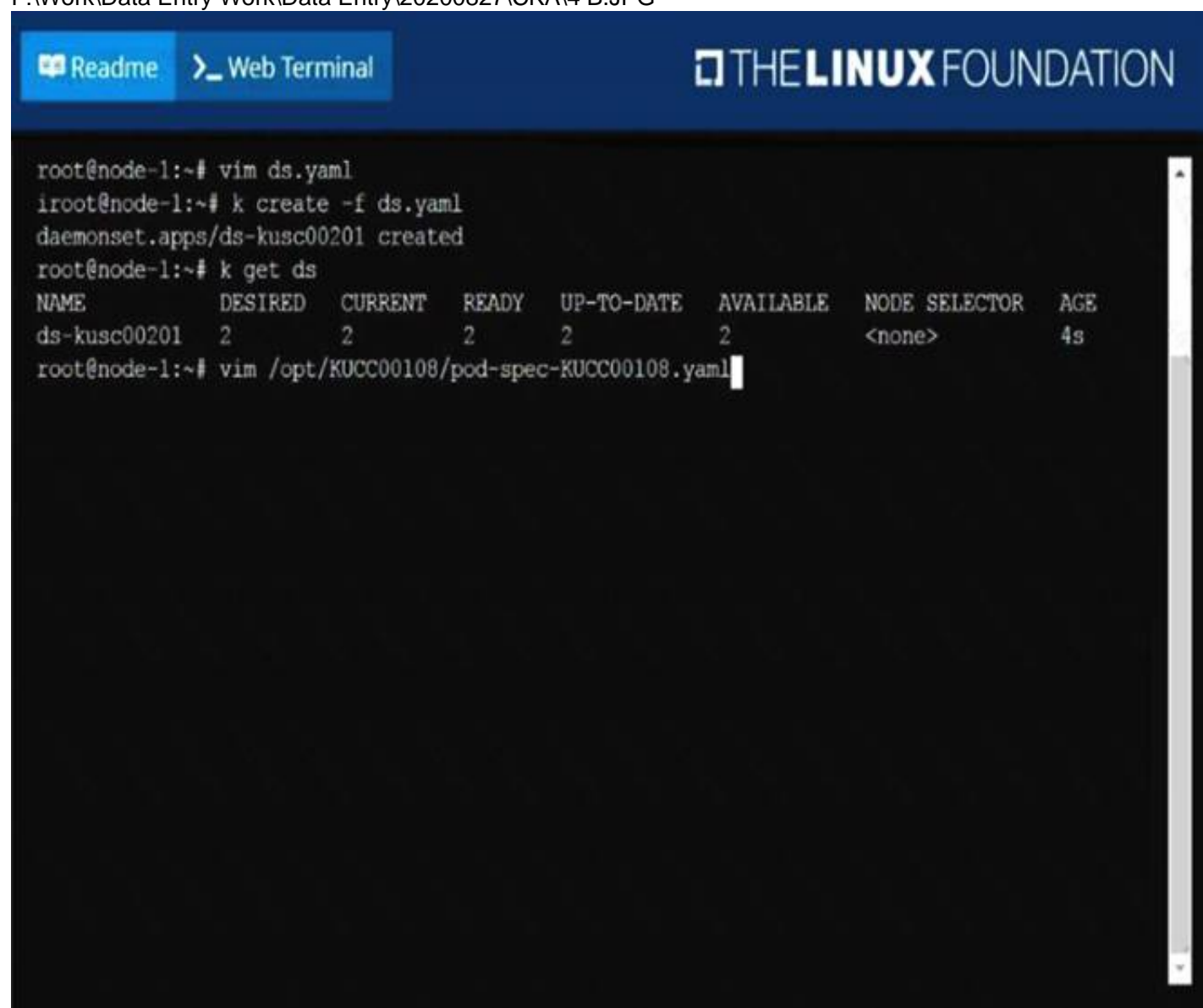
- A. Mastered  
B. Not Mastered

**Answer: A**

**Explanation:**

solution

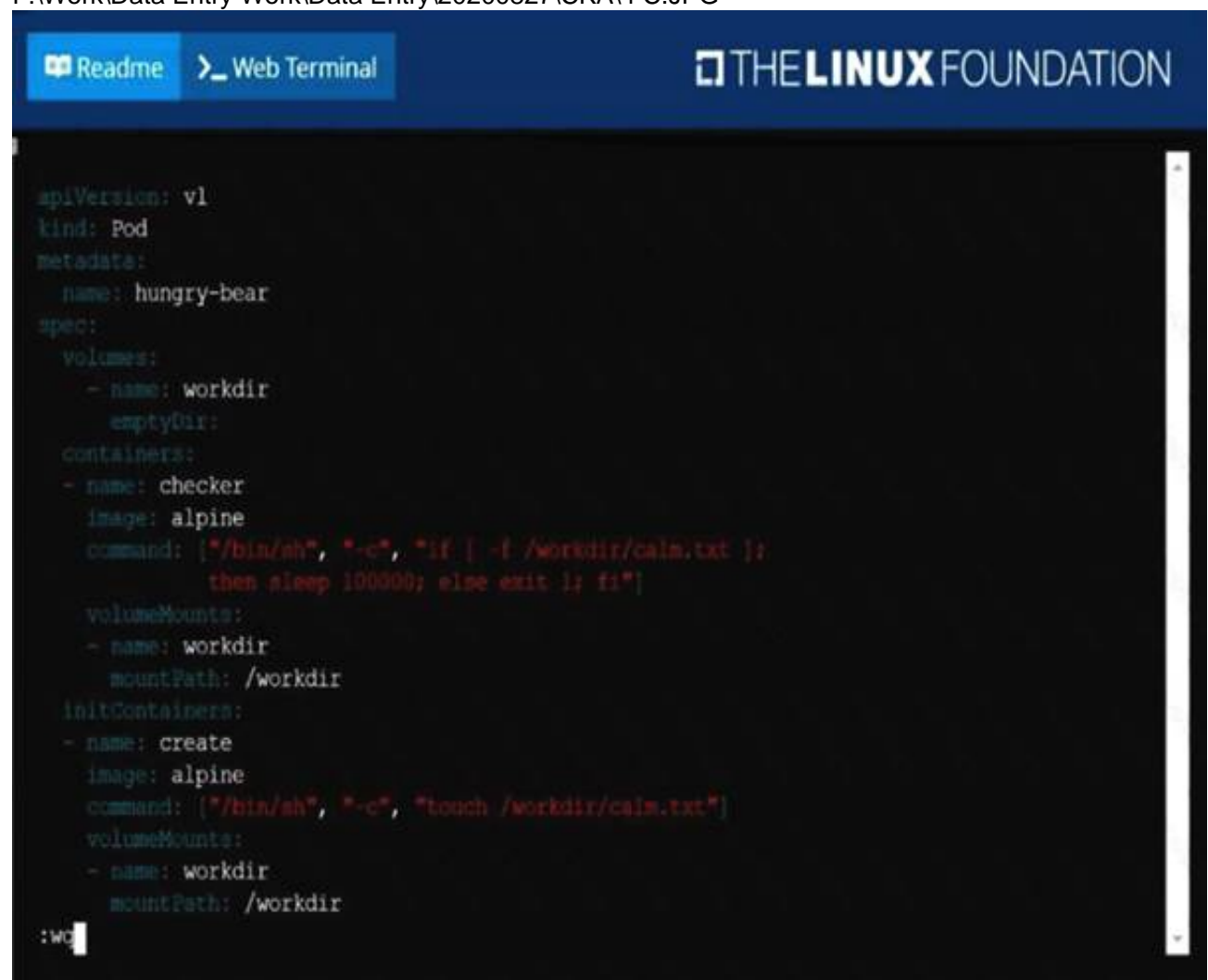
F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 B.JPG



The screenshot shows a web terminal interface with a dark background. At the top, there is a blue header bar with a 'Readme' button and a 'Web Terminal' button. The terminal content shows a series of commands and their outputs:

```
root@node-1:~# vim ds.yaml
iroot@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
```

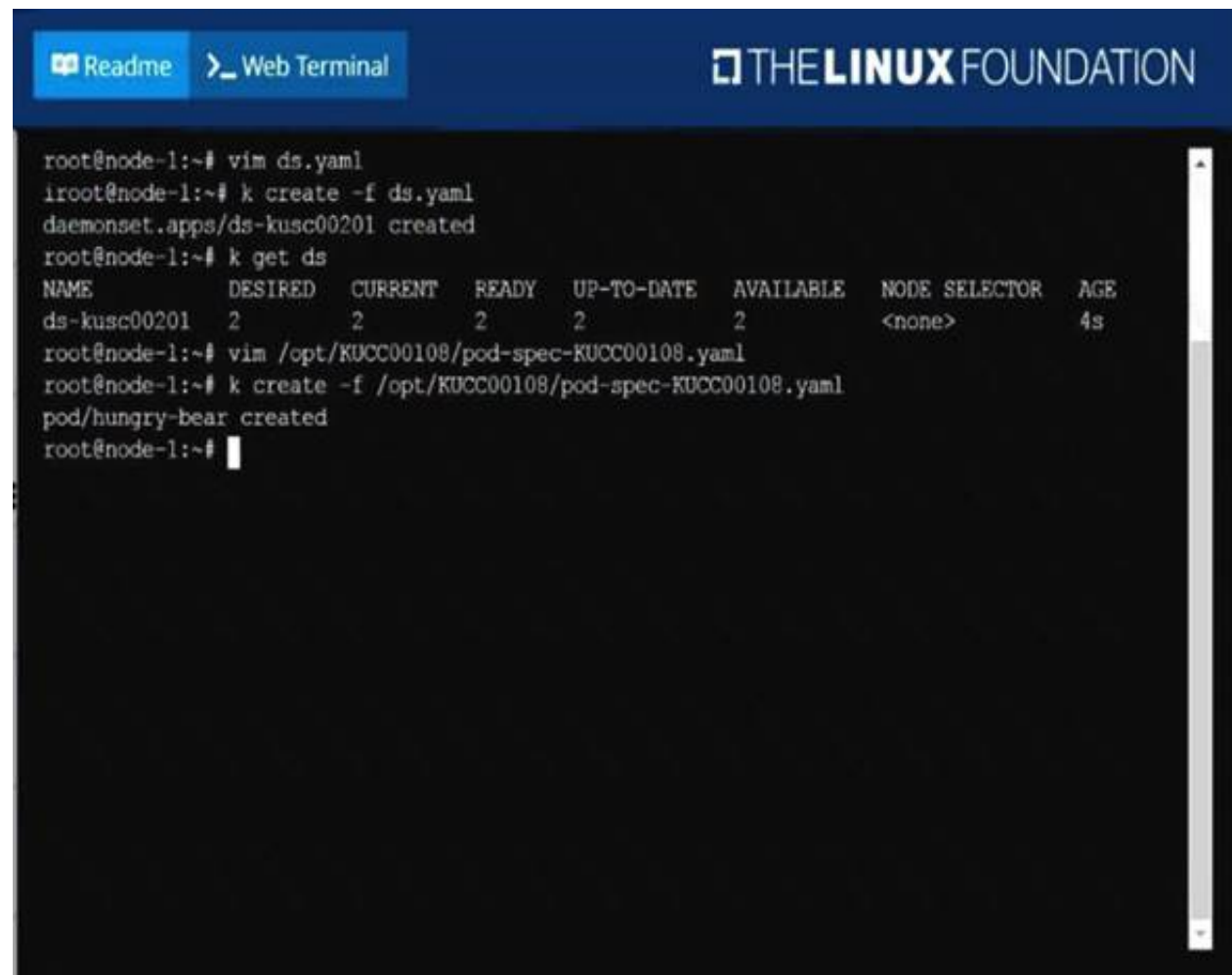
F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 C.JPG



The screenshot shows a web terminal interface with a dark background. At the top, there is a blue header bar with a 'Readme' button and a 'Web Terminal' button. The terminal content shows a Kubernetes pod specification in YAML format:

```
apiVersion: v1
kind: Pod
metadata:
  name: hungry-bear
spec:
  volumes:
  - name: workdir
    emptyDir: {}
  containers:
  - name: checker
    image: alpine
    command: ["/bin/sh", "-c", "if [ -f /workdir/calm.txt ];
      then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
  initContainers:
  - name: create
    image: alpine
    command: ["/bin/sh", "-c", "touch /workdir/calm.txt"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
:wc
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 D.JPG



NEW QUESTION 10

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}'

NEW QUESTION 11

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 15

Create and configure the servicefront-end-serviceso it's accessiblethroughNodePortand routes to theexisting pod namedfront-end.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

ReadmeWeb Terminal

THE LINUX FOUNDATION

```
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po  fron-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "fron-end" not found
root@node-1:~# k expose po  front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
front-end-service    NodePort     10.103.221.227   <none>       80:31828/TCP     3s
kubernetes           ClusterIP    10.96.0.1        <none>       443/TCP          77d
root@node-1:~#
```

NEW QUESTION 20

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 24

.....

## 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>