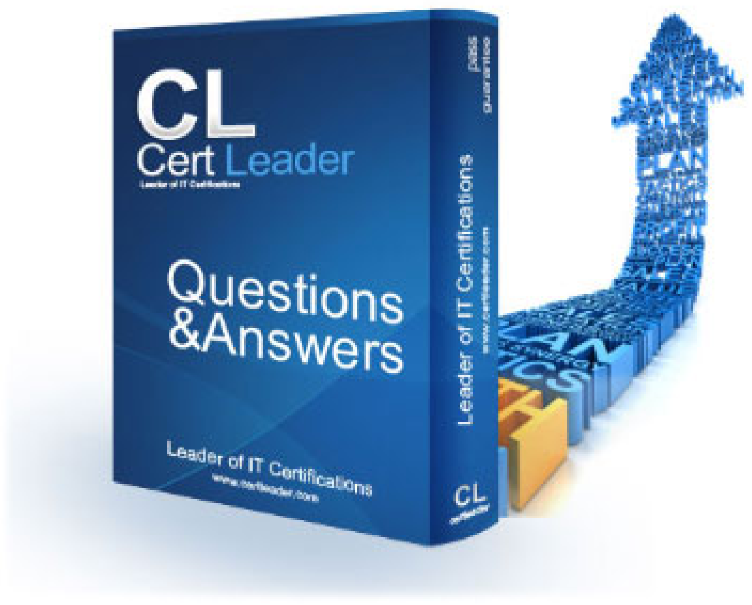
CKA Dumps

Certified Kubernetes Administrator (CKA) Program

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



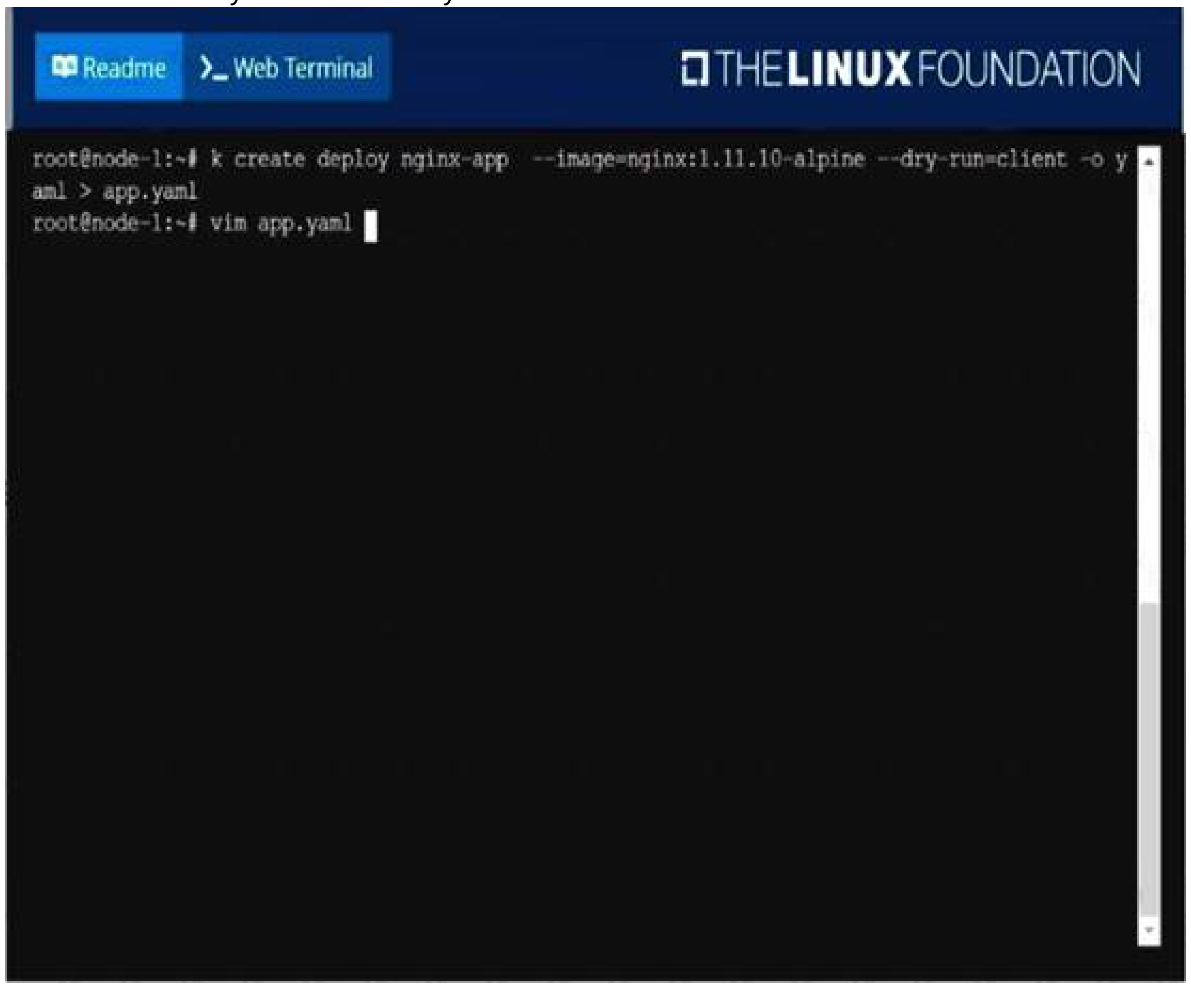
NEW QUESTION 1
Create a deployment as follows:
- Name:nginx-app
- Using containernginxwithversion 1.11.10-alpine
- The deployment should contain3replicas

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

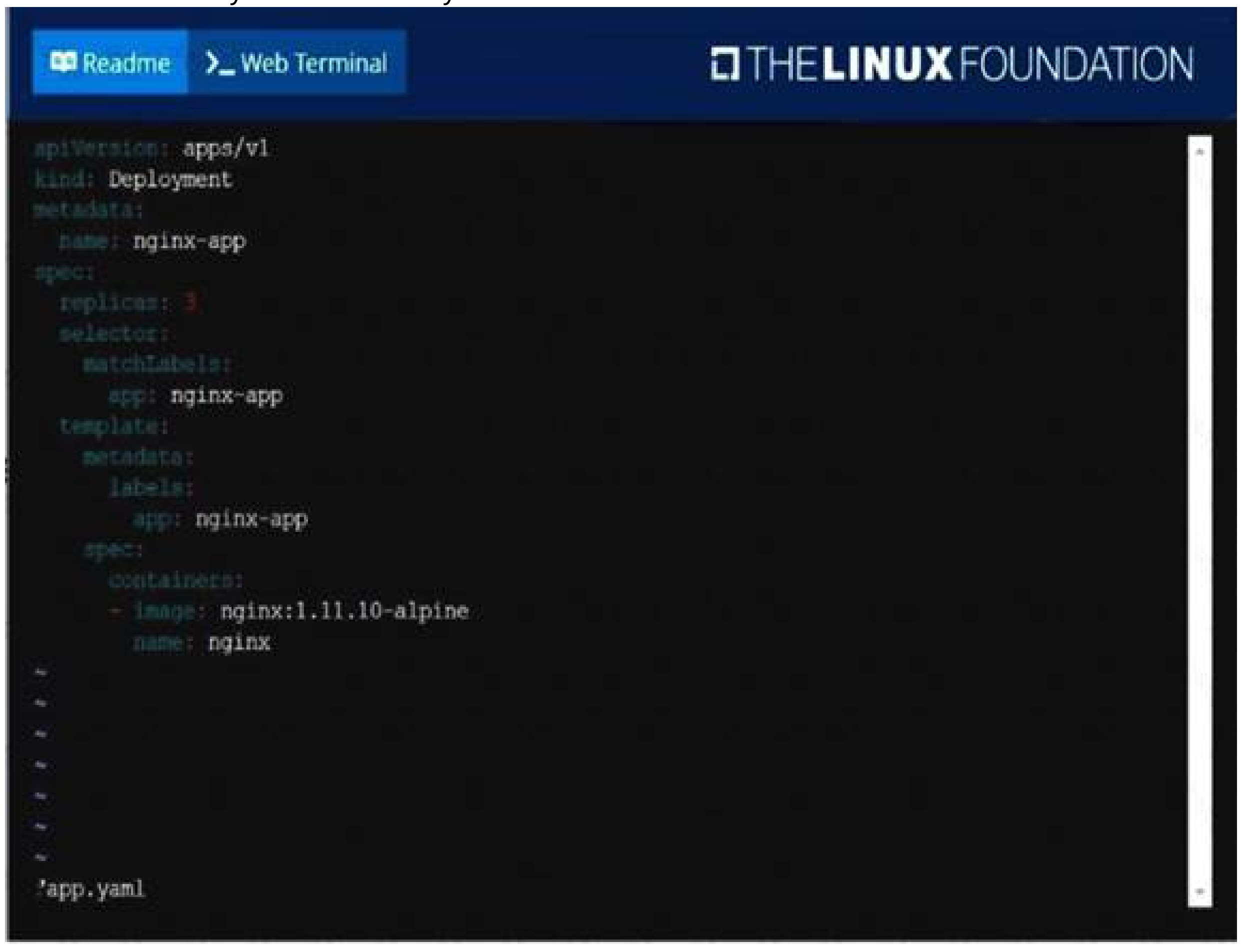
Answer: A

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



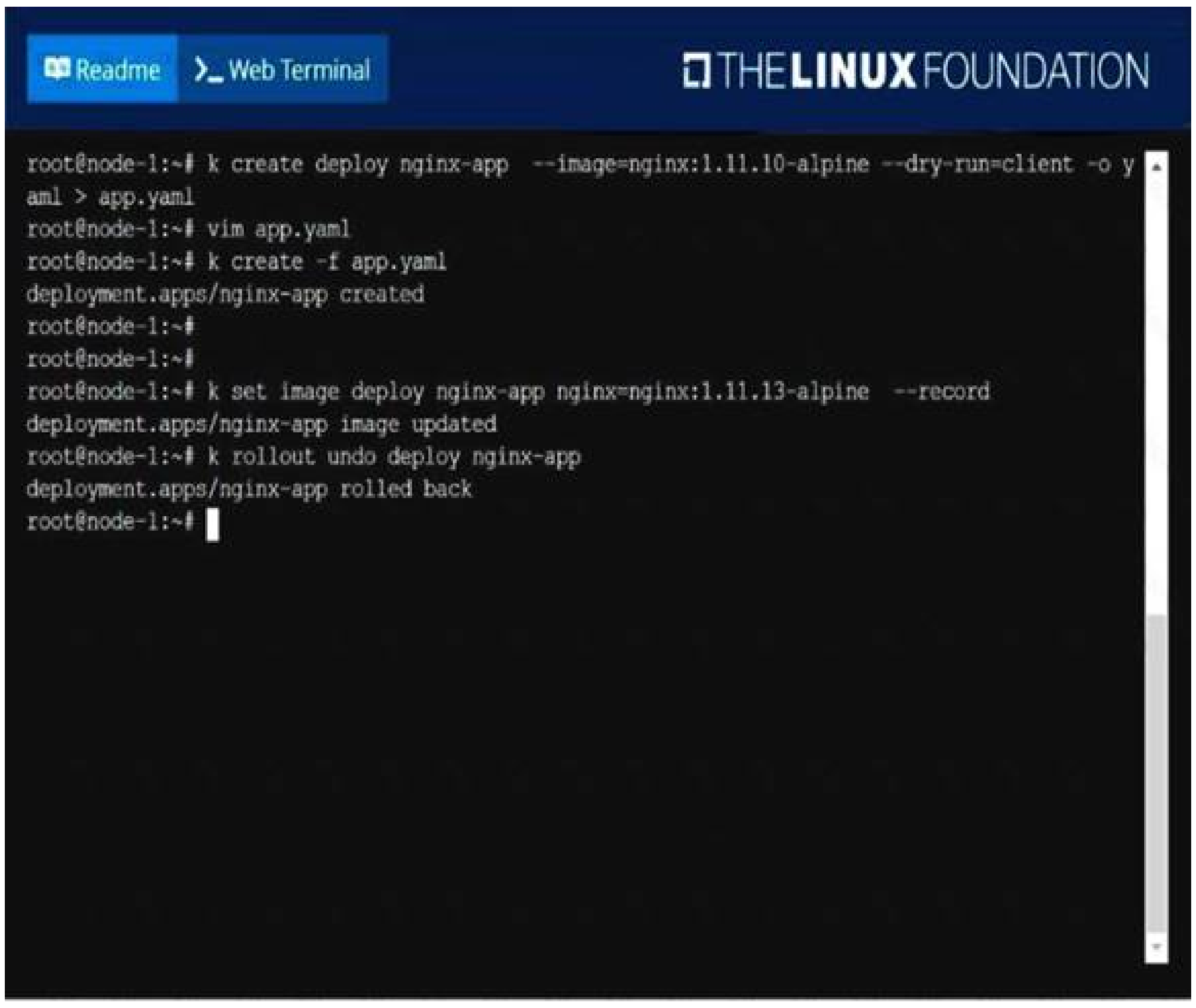
```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
```

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



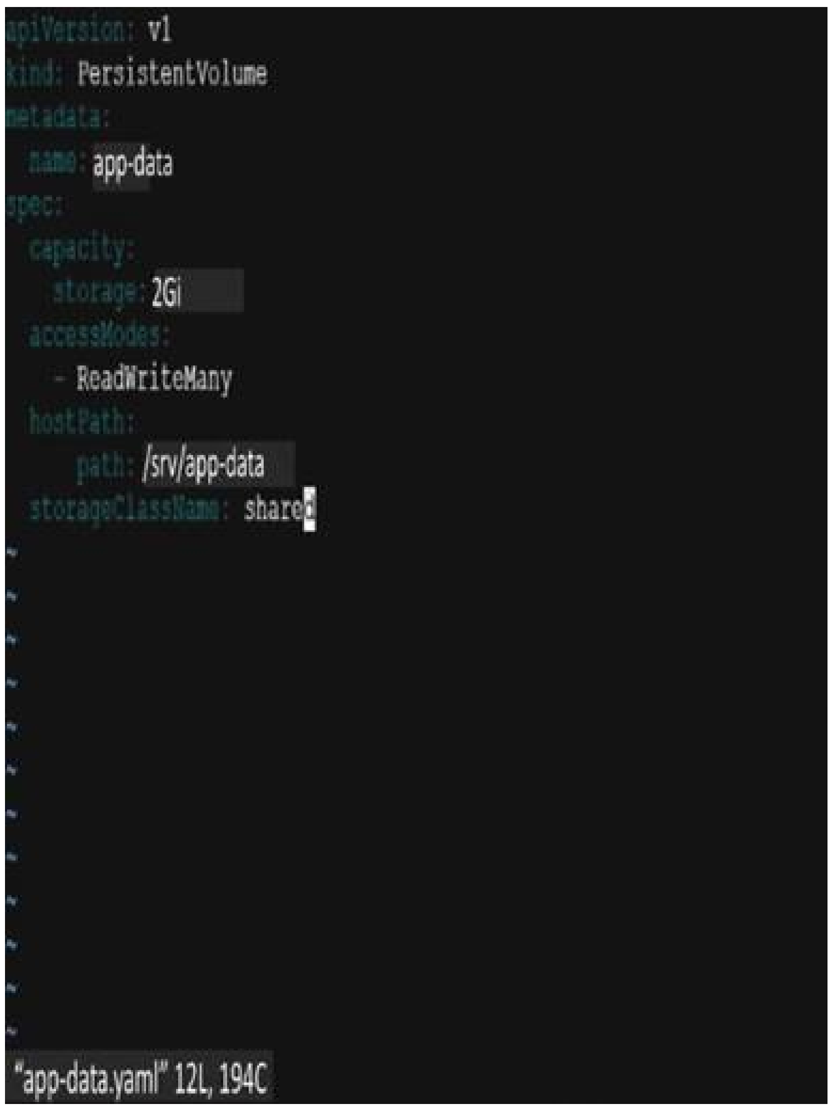
```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
root@node-1:~# k create -f app.yaml
deployment.apps/nginx-app created
root@node-1:~#
root@node-1:~#
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine  --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 2
Create a persistent volume with nameapp-data, of capacity2Giandaccess modeReadWriteMany. Thetype of volume ishostPathand itslocation is/srv/app-data.

A. Mastered
B. Not Mastered

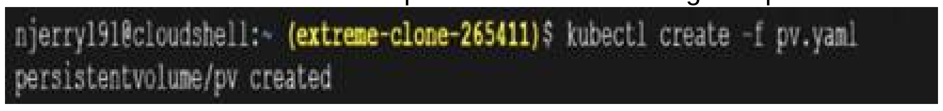
Answer: A

Explanation:
solution
Persistent Volume
A persistent volume is a piece of storage in aKubernetes 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 knowthe 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 PersistentVolume
kind: PersistentVolumeapiVersion: v1metadata:name:app-dataspec:capacity: # defines the capacity of PV we are creatingstorage:2Gi#the amount of storage we are tying to claimaccessModes: # defines the rights of the volumewe are creating-ReadWriteManyhostPath:path: "/srv/app-data" # path to which we are creating the volume
Challenge
> Create a Persistent Volume namedapp-data, with access modeReadWriteMany, storage classname shared,2Giof storage capacity and the host path/srv/app-data.



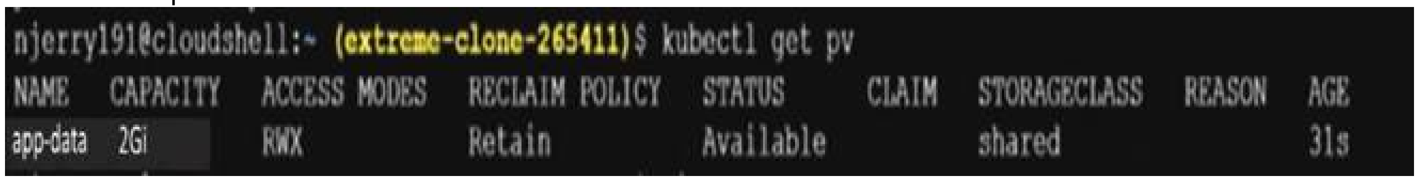
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
      path: /srv/app-data
  storageClassName: shared
~
~
~
~
~
~
~
~
~
~
~
~
"app-data.yaml" 12L, 194C
```

* 2. Save the file and create the persistent volume. Image for post



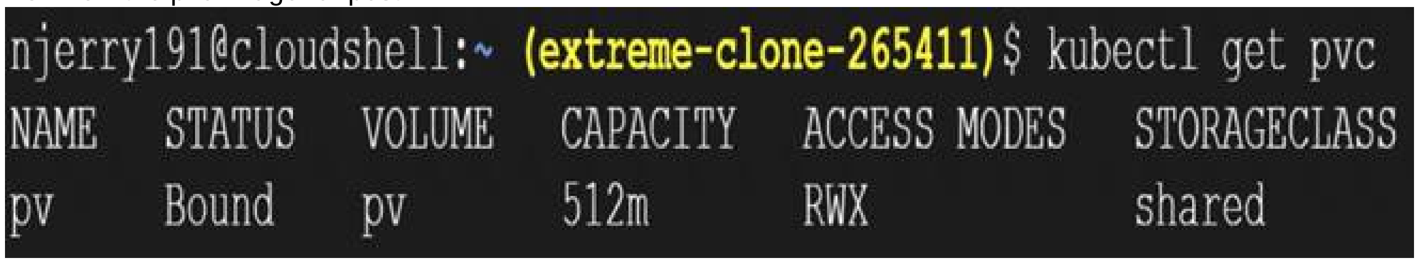
```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl create -f pv.yaml
persistentvolume/pv created
```

* 3. View the persistent volume.



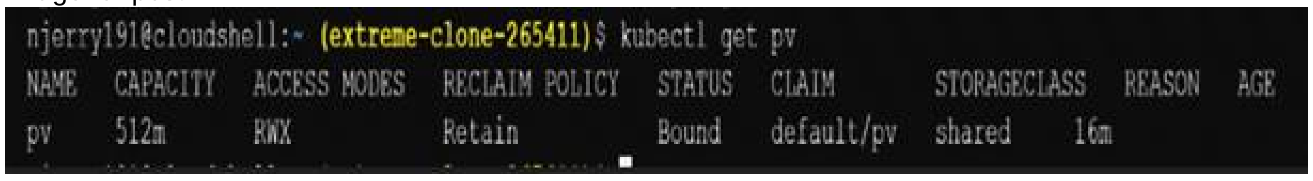
```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
app-data  2Gi        RWX            Retain           Available           shared                  31s
```

> Our persistent volume status is available meaning it is available and it has not been mounted yet. This status willchange 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. Ensurethat the Persistent Volume Claim has the same storageClassName as the persistentVolume you had previously created.
kind: PersistentVolumeapiVersion: v1metadata:name:app-data spec:
accessModes:-ReadWriteManyresources:
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 fromavailabletobound.
* 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: nullname: app-dataspec:volumes:- name:congigpvcpersistenVolumeClaim:claimName: app-datacontainers:- image: nginxname: appvolumeMounts:- mountPath: "/srv/app-data"name: configpvc

NEW QUESTION 3
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 4
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

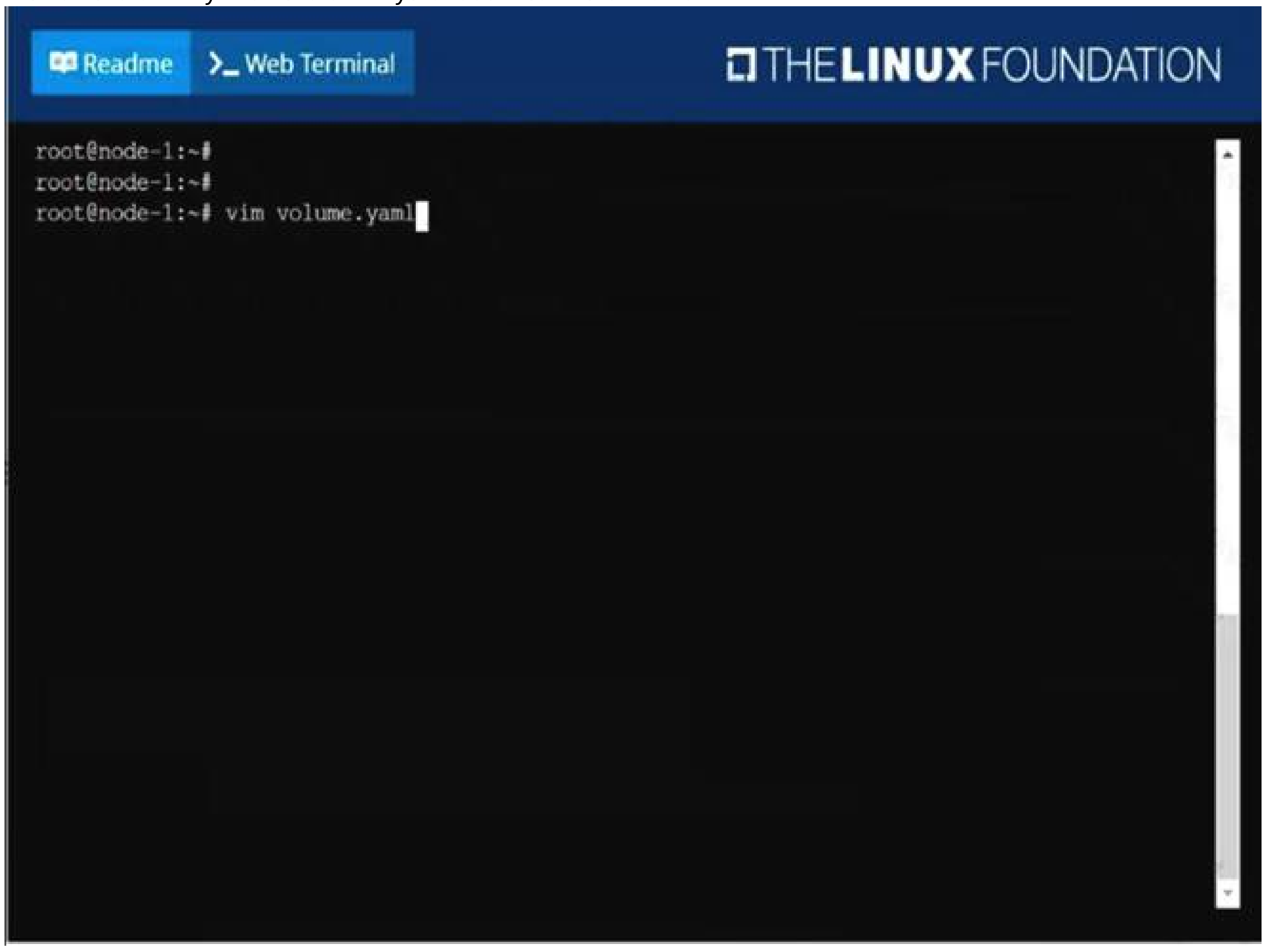
NEW QUESTION 5
Create a pod as follows:
- Name:non-persistent-redis
- container Image:redis
- Volume with name:cache-control
- Mount path:/data/redis

The pod should launch in thestagingnamespace and the volumemust notbe persistent.

A. Mastered
B. Not Mastered

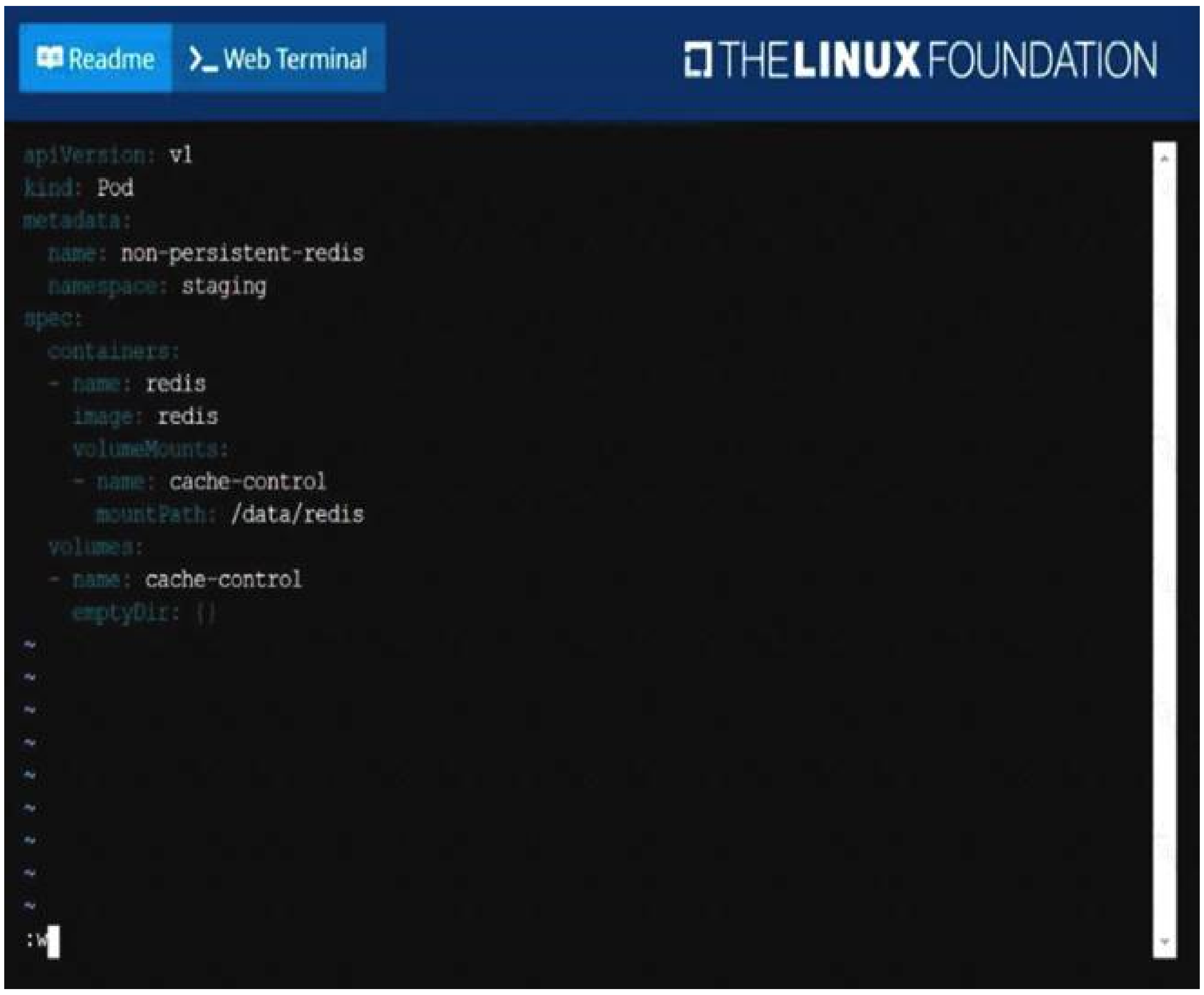
Answer: A

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



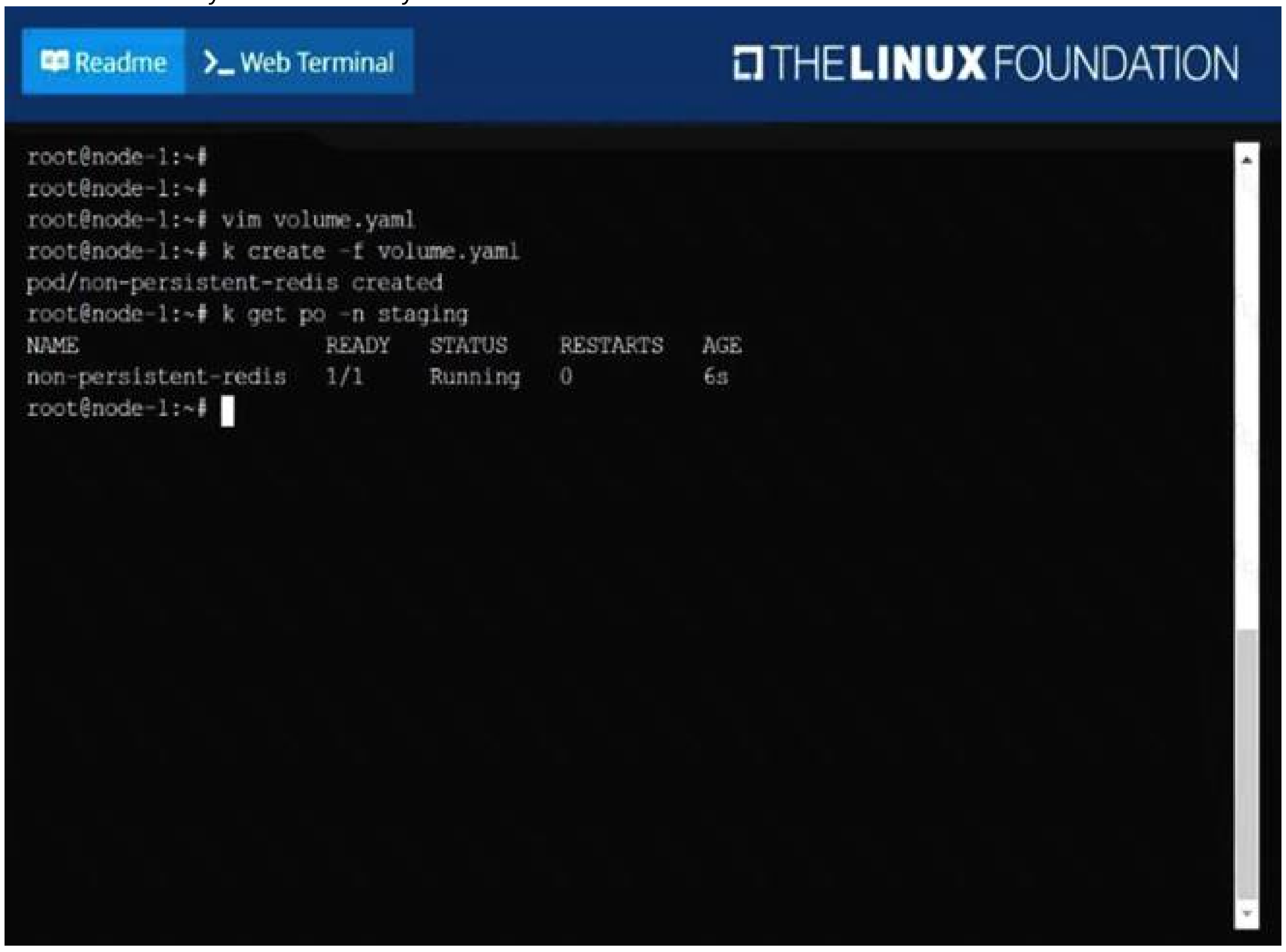
```
root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml
```

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



```
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 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 6
Get list of all the pods showing name and namespace with a jsonpath expression.

A. Mastered
B. Not Mastered

Answer: A

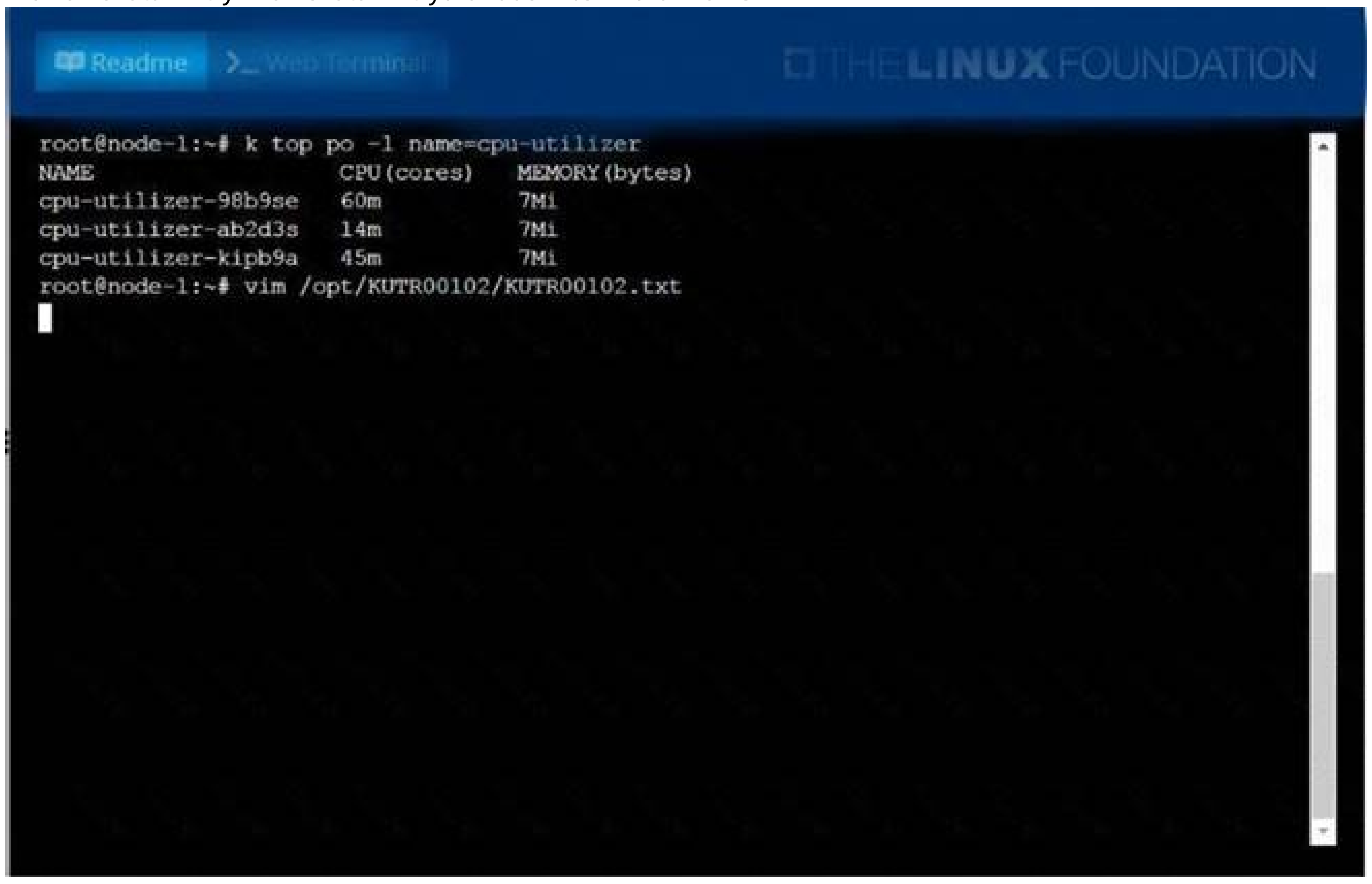
Explanation:
kubectl get pods -o=jsonpath="{.items[*]['metadata.name'
, 'metadata.namespace']}"

NEW QUESTION 7
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

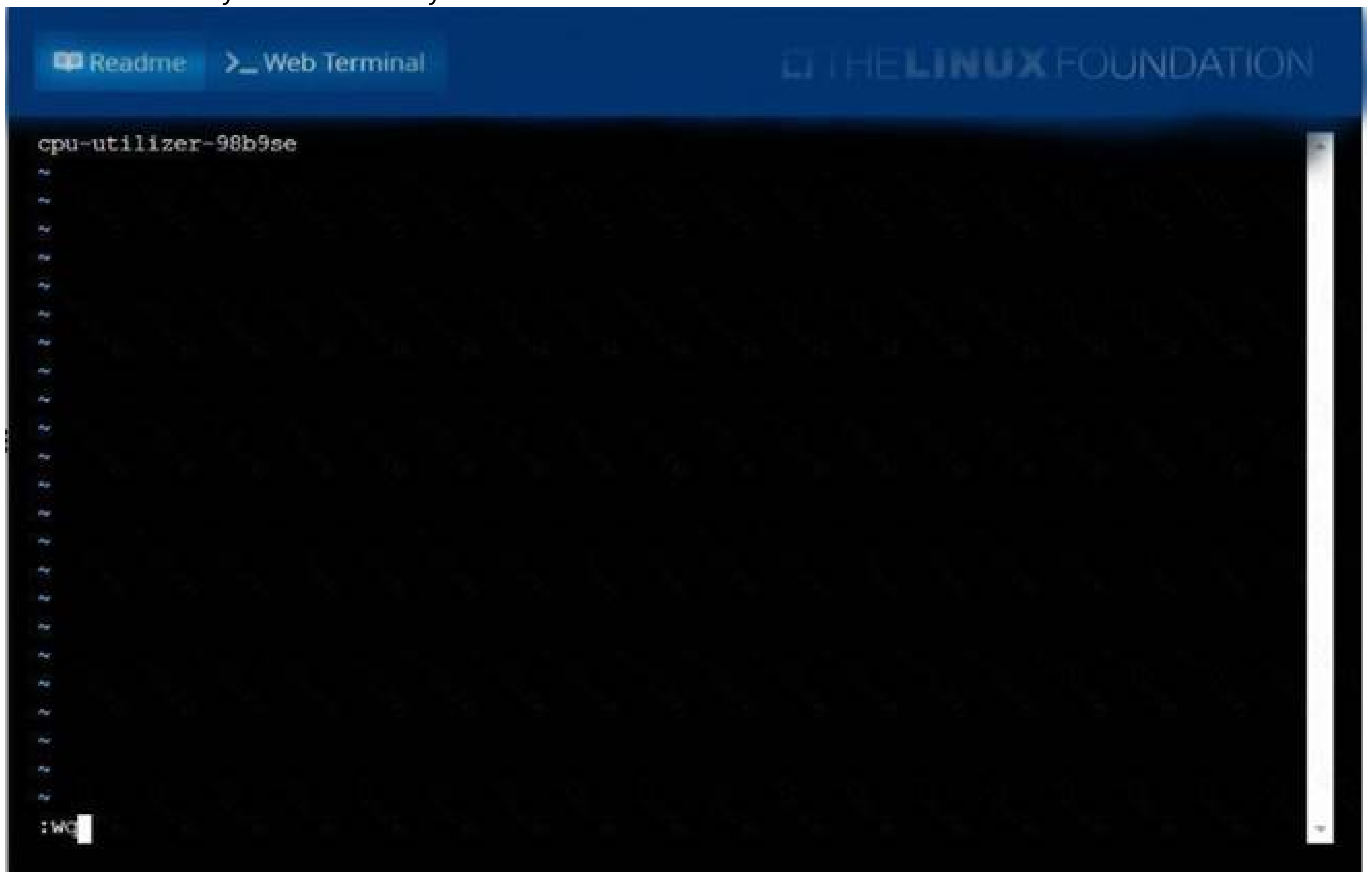
Answer: A

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



```
root@node-1:~# k top po -l name=cpu-utilizer
NAME                  CPU(cores)   MEMORY(bytes)
cpu-utilizer-98b9se   60m          7Mi
cpu-utilizer-ab2d3s   14m          7Mi
cpu-utilizer-kipb9a   45m          7Mi
root@node-1:~# vim /opt/KUTR00102/KUTR00102.txt
```

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



```
cpu-utilizer-98b9se
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

NEW QUESTION 8
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"

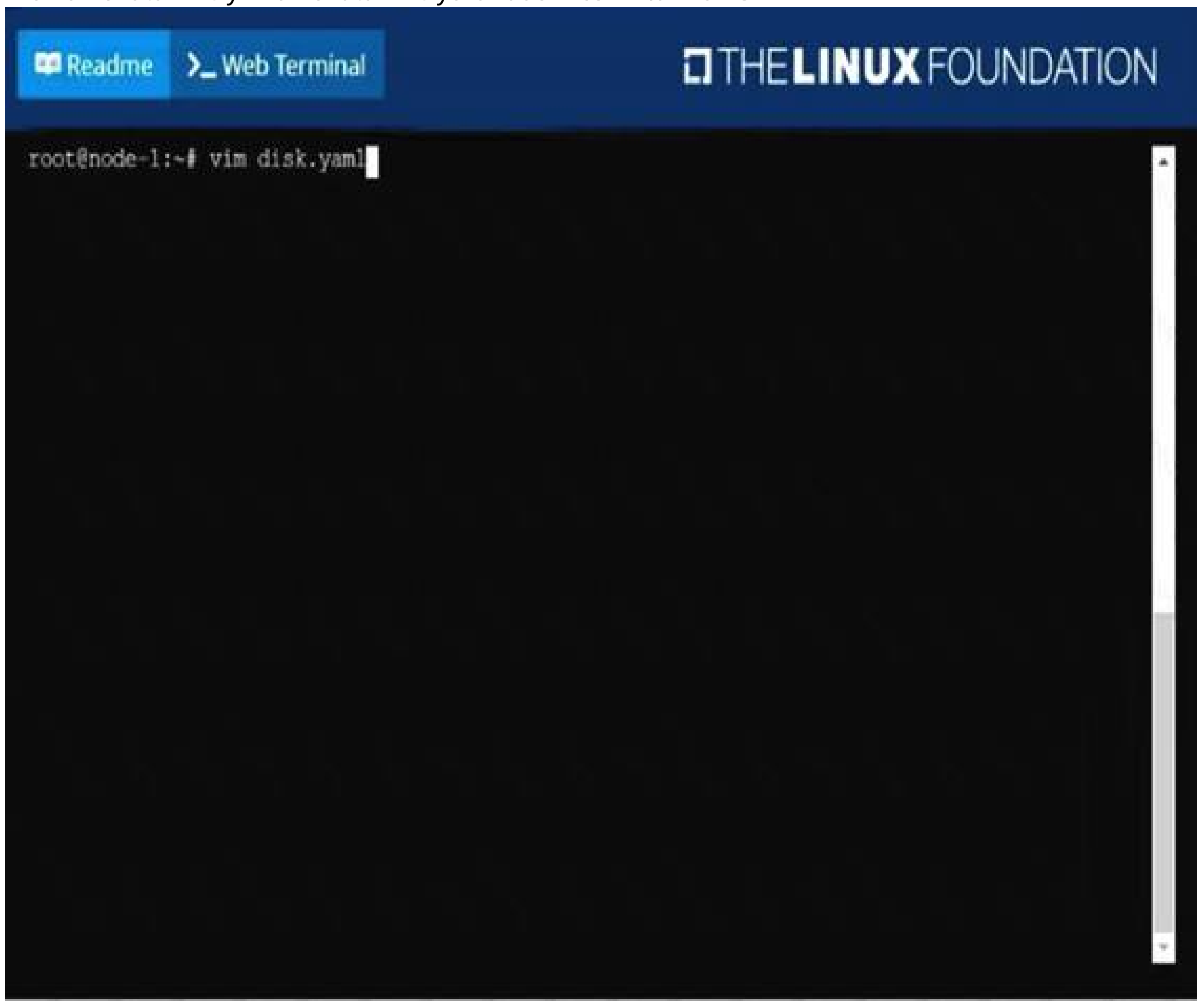
NEW QUESTION 9
Schedule a pod as follows:
- Name: nginx-kusc00101
- Image: nginx
- Node selector: disk=ssd

A. Mastered

B. Not Mastered

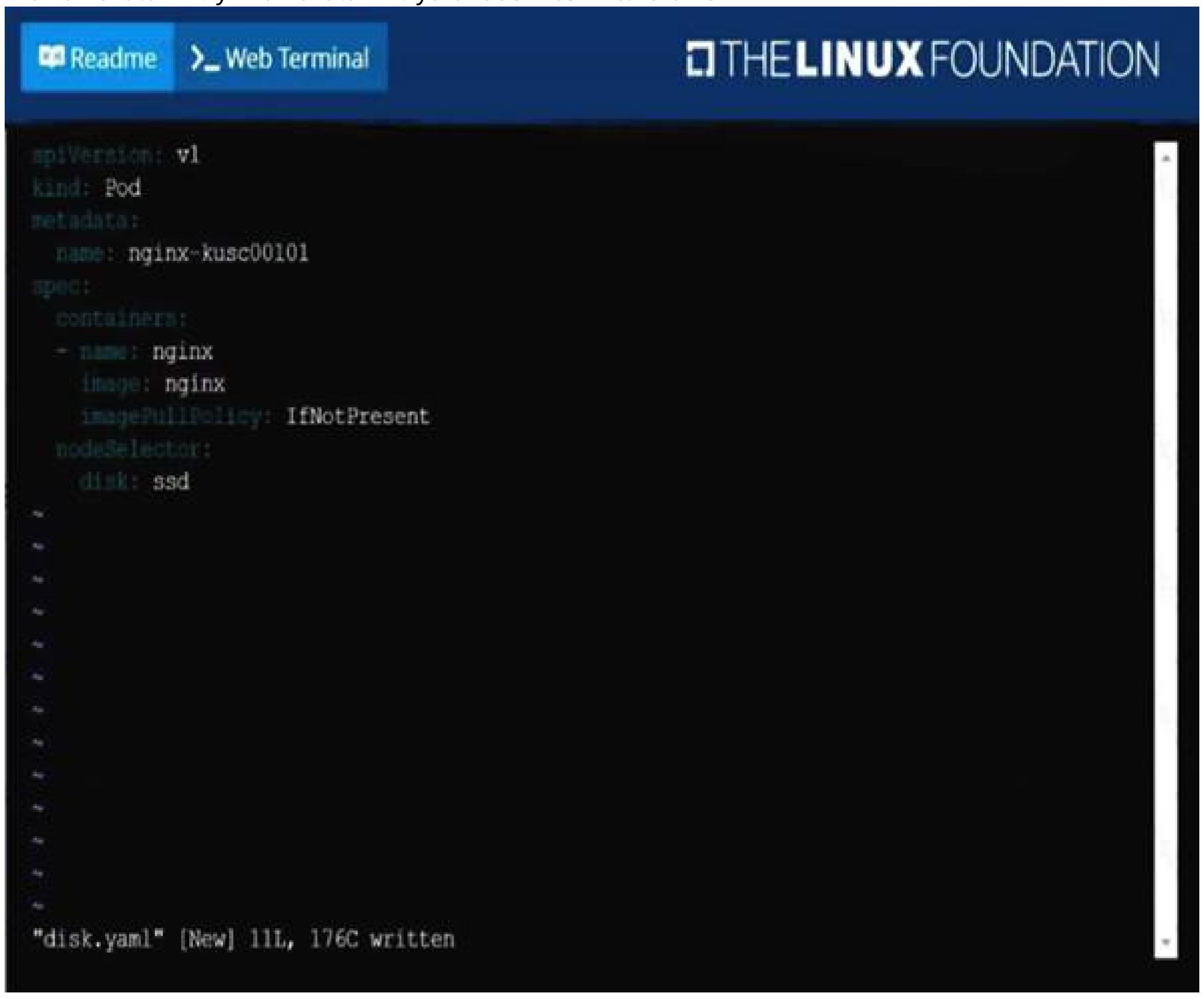
Answer: A

Explanation:
solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 B.JPG



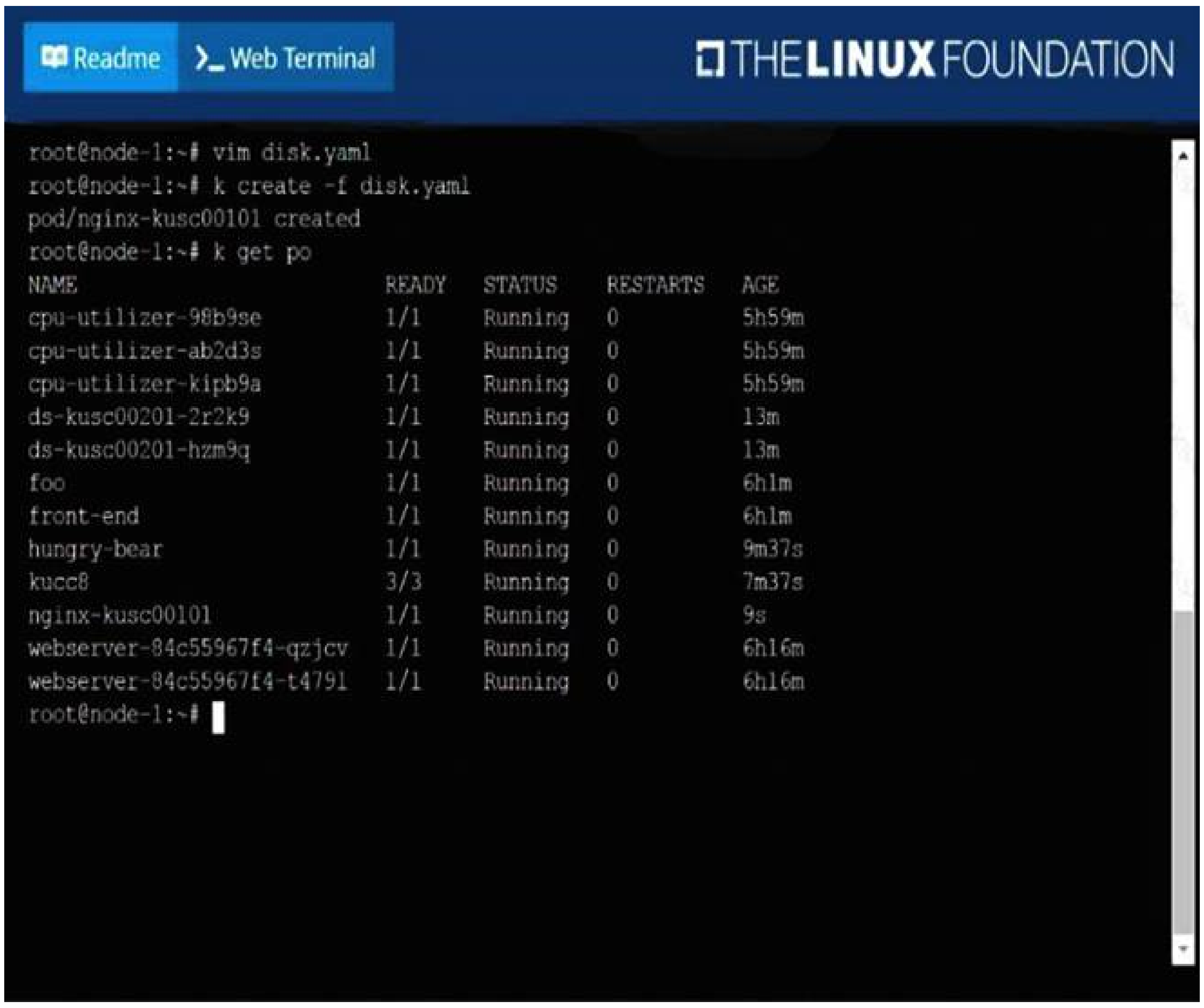
```
root@node-1:~# vim disk.yaml
```

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



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kusc00101
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    disk: ssd
~
~
~
~
~
~
~
~
~
~
~
~
~
"disk.yaml" [New] 11L, 176C written
```

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



```
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-t4791    1/1     Running   0          6h16m
root@node-1:~#
```

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

A. Mastered
B. Not Mastered

Answer: A

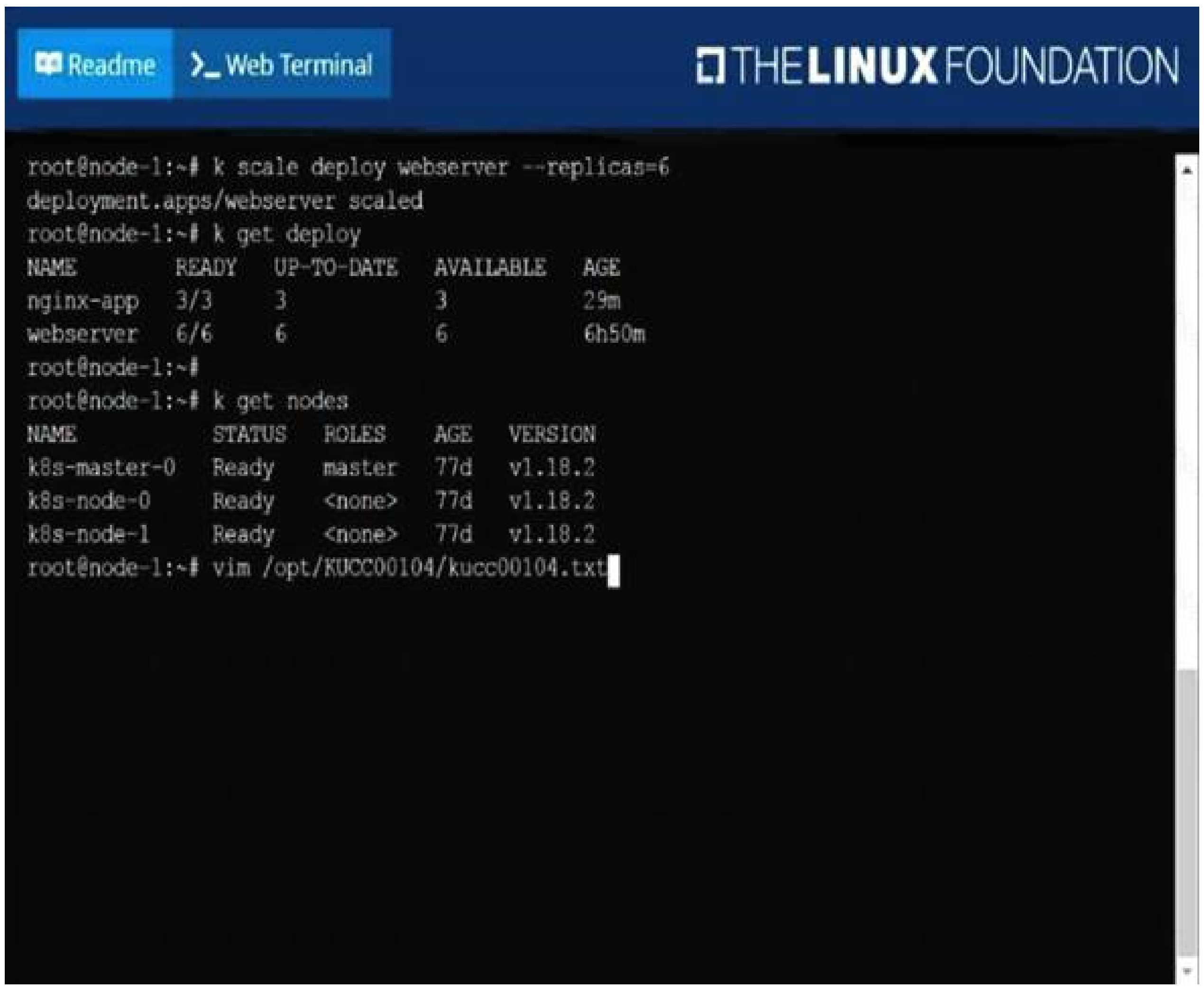
Explanation:
// create a pod
kubectl run nginx --image=nginx --restart=Never --port=80
// List the pod with different verbosity kubectl get po nginx --v=7
kubectl get po nginx --v=8 kubectl get po nginx --v=9

NEW QUESTION 11
Check to see how many worker nodes are ready (not including nodes taintedNoSchedule) and write the number to/opt/KUCC00104/kucc00104.txt.

A. Mastered
B. Not Mastered

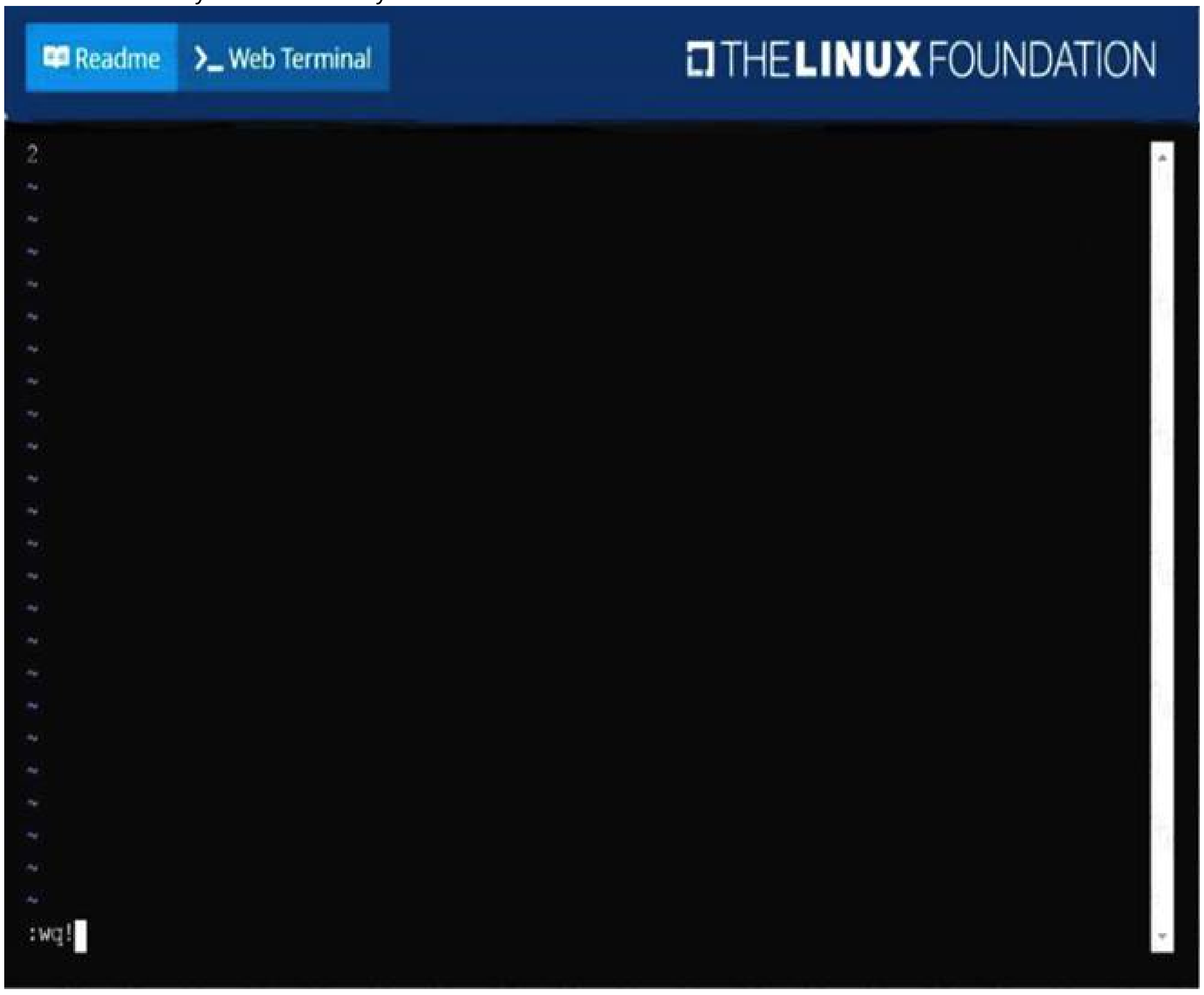
Answer: A

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



```
root@node-1:~# k scale deploy webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app   3/3     3            3           29m
webserver   6/6     6            6           6h50m
root@node-1:~#
root@node-1:~# k get nodes
NAME           STATUS   ROLES    AGE   VERSION
k8s-master-0   Ready    master   77d   v1.18.2
k8s-node-0     Ready    <none>   77d   v1.18.2
k8s-node-1     Ready    <none>   77d   v1.18.2
root@node-1:~# vim /opt/KUCC00104/kucc00104.txt
```

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



```
2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

NEW QUESTION 12

......