

CKA Dumps

Certified Kubernetes Administrator (CKA) Program

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



NEW QUESTION 1

List `nginx-dev` and `nginx-prod` pod and delete those pods

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
kubectl get pods -o wide
```

```
kubectl delete po nginx-dev kubectl delete po nginx-prod
```

NEW QUESTION 2

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 -nengineering -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 3

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

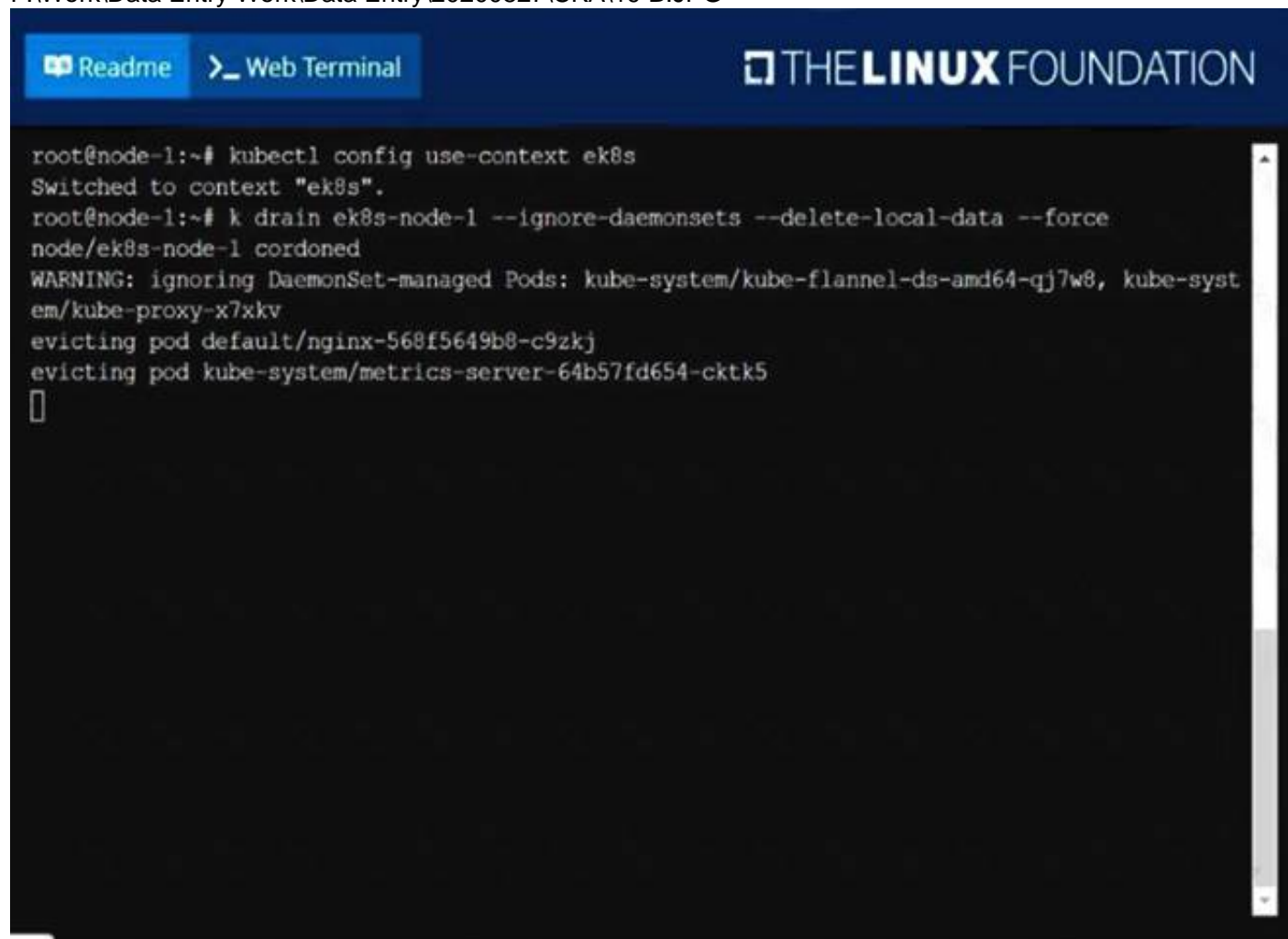
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

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



```
root@node-1:~# kubectl config use-context ek8s
Switched to context "ek8s".
root@node-1:~# k drain ek8s-node-1 --ignore-daemonsets --delete-local-data --force
node/ek8s-node-1 cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-amd64-qj7w8, kube-system/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zkj
evicting pod kube-system/metrics-server-64b57fd654-cktk5
[]
```

NEW QUESTION 4

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 PersistentVolume

kind: PersistentVolume
apiVersion: v1
metadata:
 name: app-data
spec:
 capacity:
 storage: 2Gi
 accessModes:
 - ReadWriteMany
 hostPath:
 path: /srv/app-data
 storageClassName: shared

Challenge

> Create a Persistent Volume named app-data, with access mode ReadWriteMany, storage class name 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: PersistentVolumeClaim
apiVersion: v1
metadata:
 name: app-data
spec:

accessModes: - ReadWriteMany
resources:

requests:
 storage: 2Gi
 storageClassName: shared

* 2. Save and create the pvc

```
njerry191@cloudshell:~(extreme-clone-2654111)$ kubectl 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 5

A Kubernetes worker node, namedwk8s-node-0is in stateNotReady.Investigate why this is the case, andperform any appropriate steps tobring the node to aReadystate,ensuring that any changes are madepermanent.

You cansshto the failednode using:

```
[student@node-1] $ | sshWk8s-node-0
```

You can assume elevatedprivileges on the node with thefollowing command:

```
[student@w8ks-node-0] $ |sudo ?Ci
```

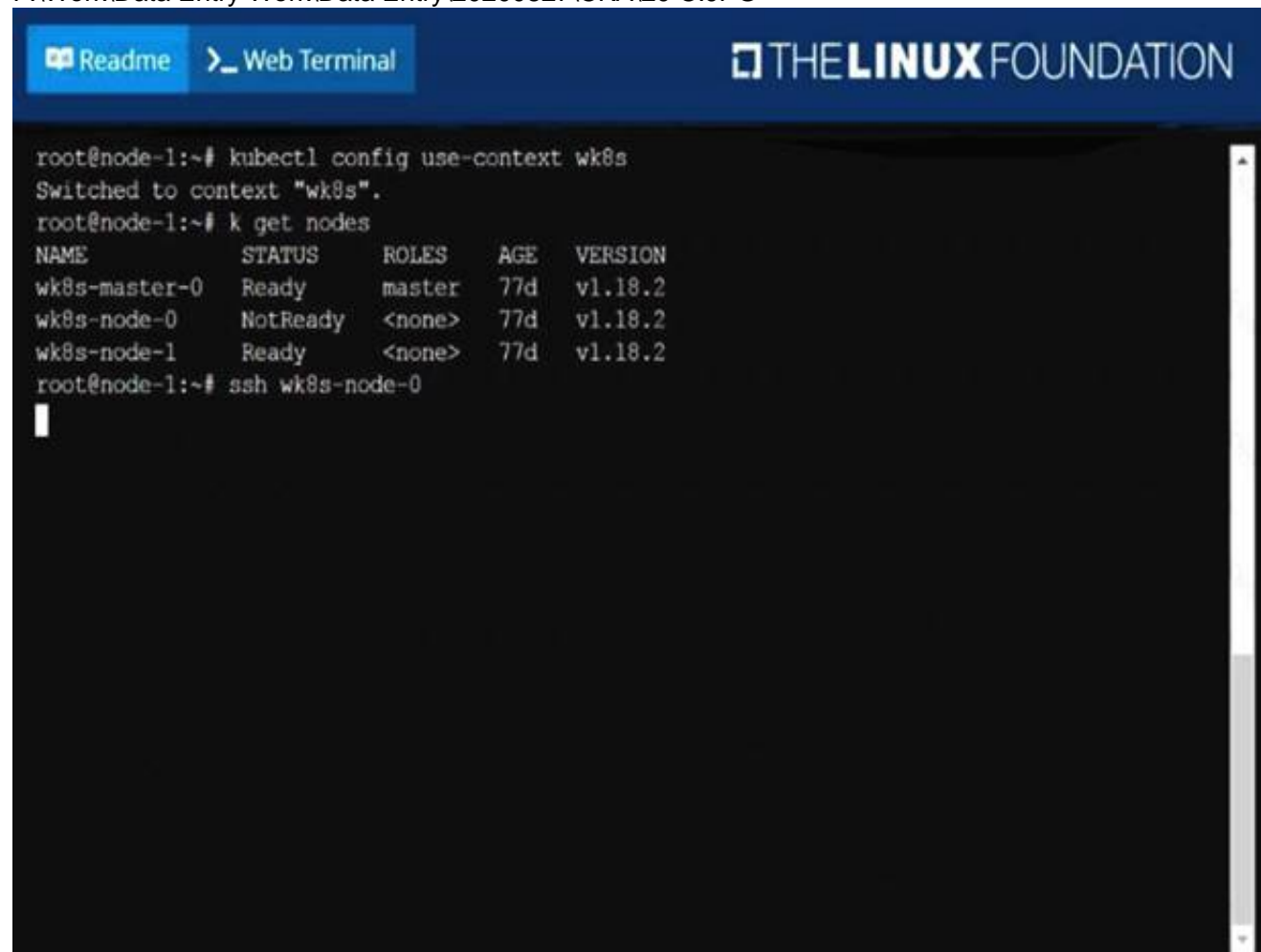
- A. Mastered
- B. Not Mastered

Answer: A

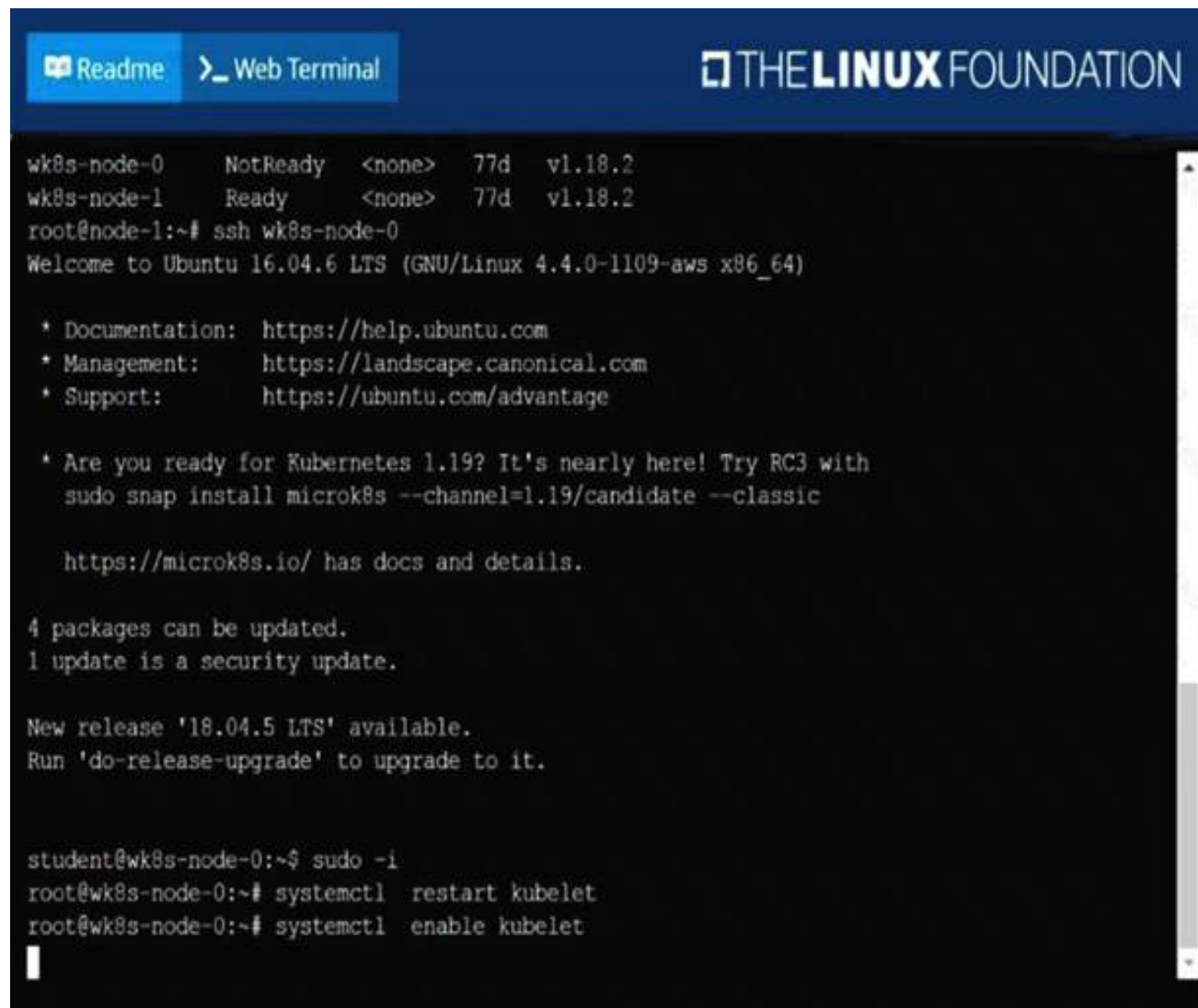
Explanation:

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



```

wk8s-node-0    NotReady  <none>  77d   v1.18.2
wk8s-node-1    Ready     <none>  77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
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-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet

```

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



```

   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-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/sys
temd/system/kubelet.service.
root@wk8s-node-0:~# exit
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.5.34 closed.
root@node-1:~# k get nodes
NAME          STATUS    ROLES    AGE   VERSION
wk8s-master-0 Ready     master   77d   v1.18.2
wk8s-node-0   Ready     <none>   77d   v1.18.2
wk8s-node-1   Ready     <none>   77d   v1.18.2
root@node-1:~#

```

NEW QUESTION 6

Check the image version in pod without the describe command

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubect! get po nginx -o jsonpath='{.spec.containers[].image}'

NEW QUESTION 7

Perform the following tasks:

- > Add an init container tohungry-bear(which has beendefined 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

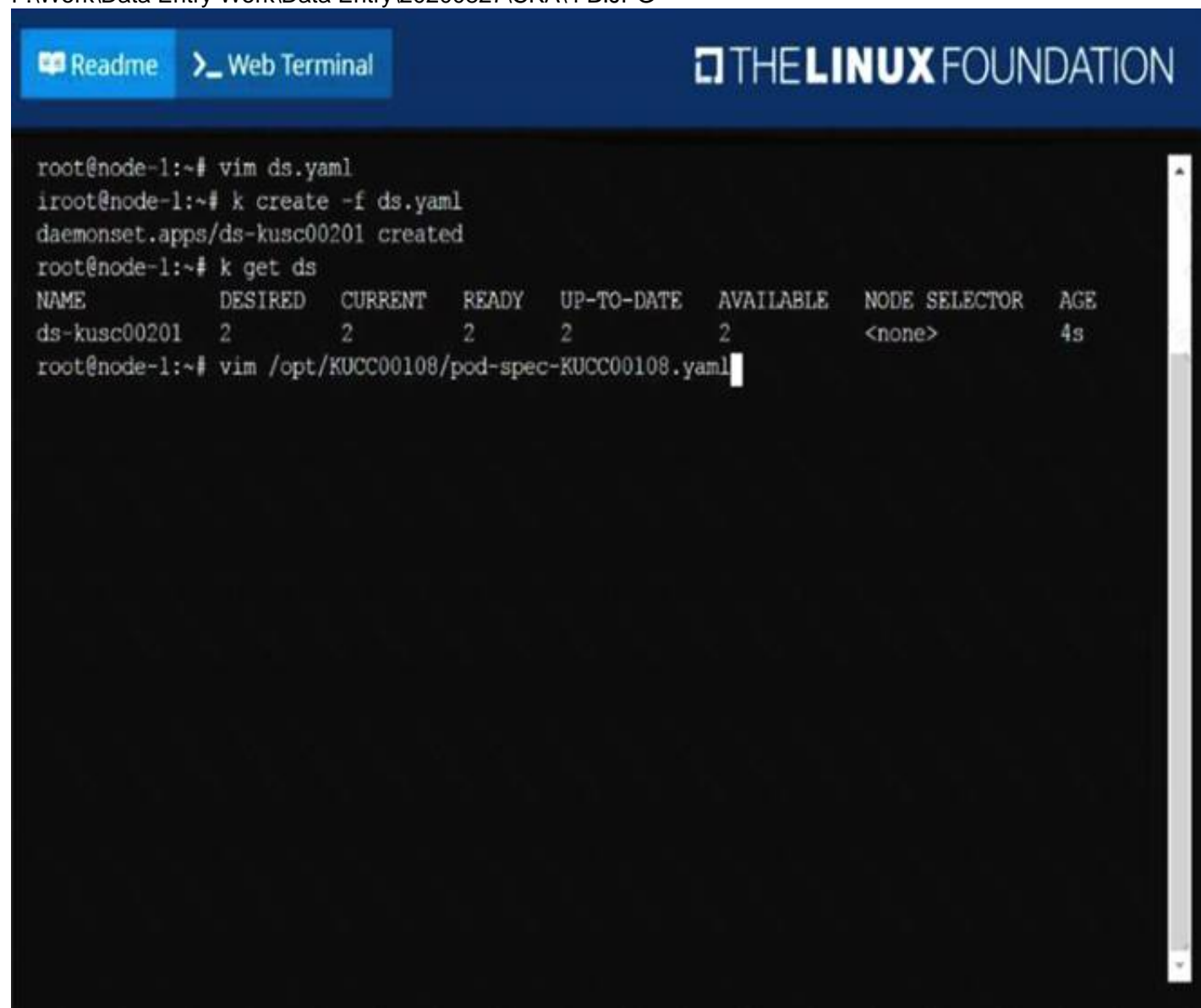
- A. Mastered
- B. Not Mastered

Answer: A

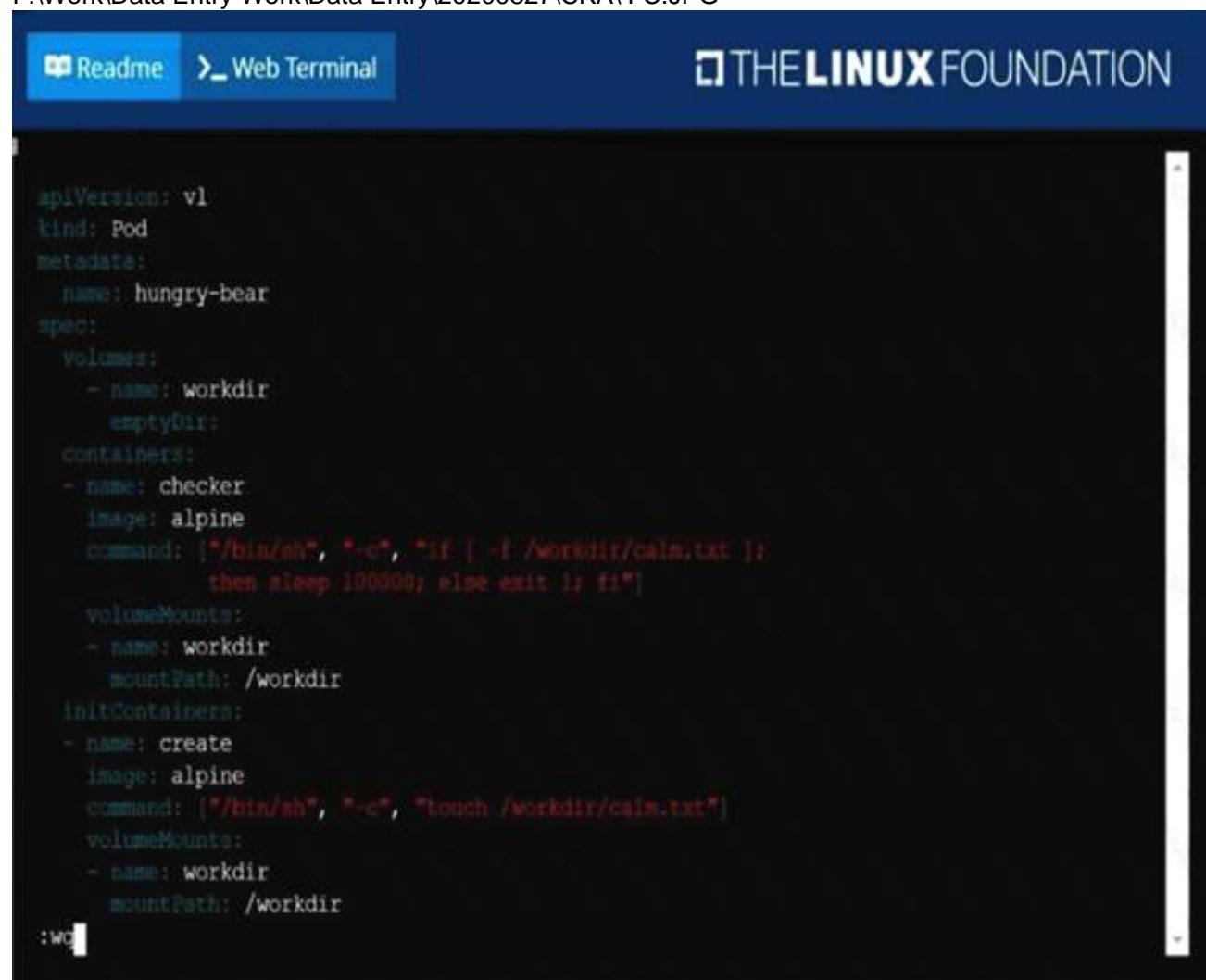
Explanation:

solution

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



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



F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 D.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:~#
```

NEW QUESTION 8

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 9

Create a snapshot of theetcdinstance running athttps://127.0.0.1:2379, saving thesnapshot to the file path /srv/data/etcd-snapshot.db.
The following TLScertificates/key are suppliedfor connecting to the server withetcdctl:

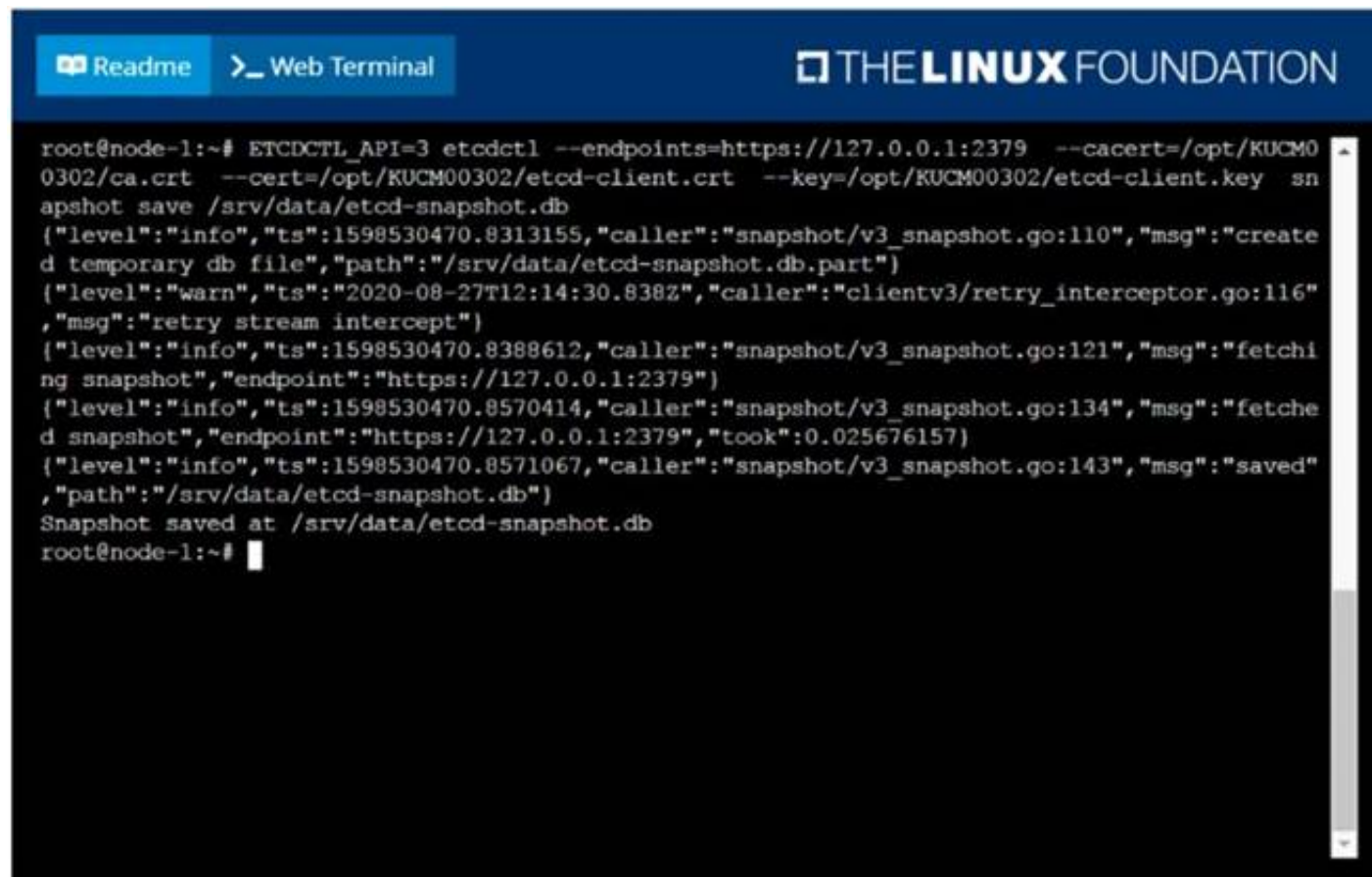
- > CA certificate:/opt/KUCM00302/ca.crt
- > Client certificate:/opt/KUCM00302/etcd-client.crt
- > Client key:Topt/KUCM00302/etcd-client.key

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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



```

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:~#

```

NEW QUESTION 10

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 10

Print pod name and start time to ??/opt/pod-status?? file

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubectl get pods -o=jsonpath='{range items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}}{end}'

NEW QUESTION 13

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

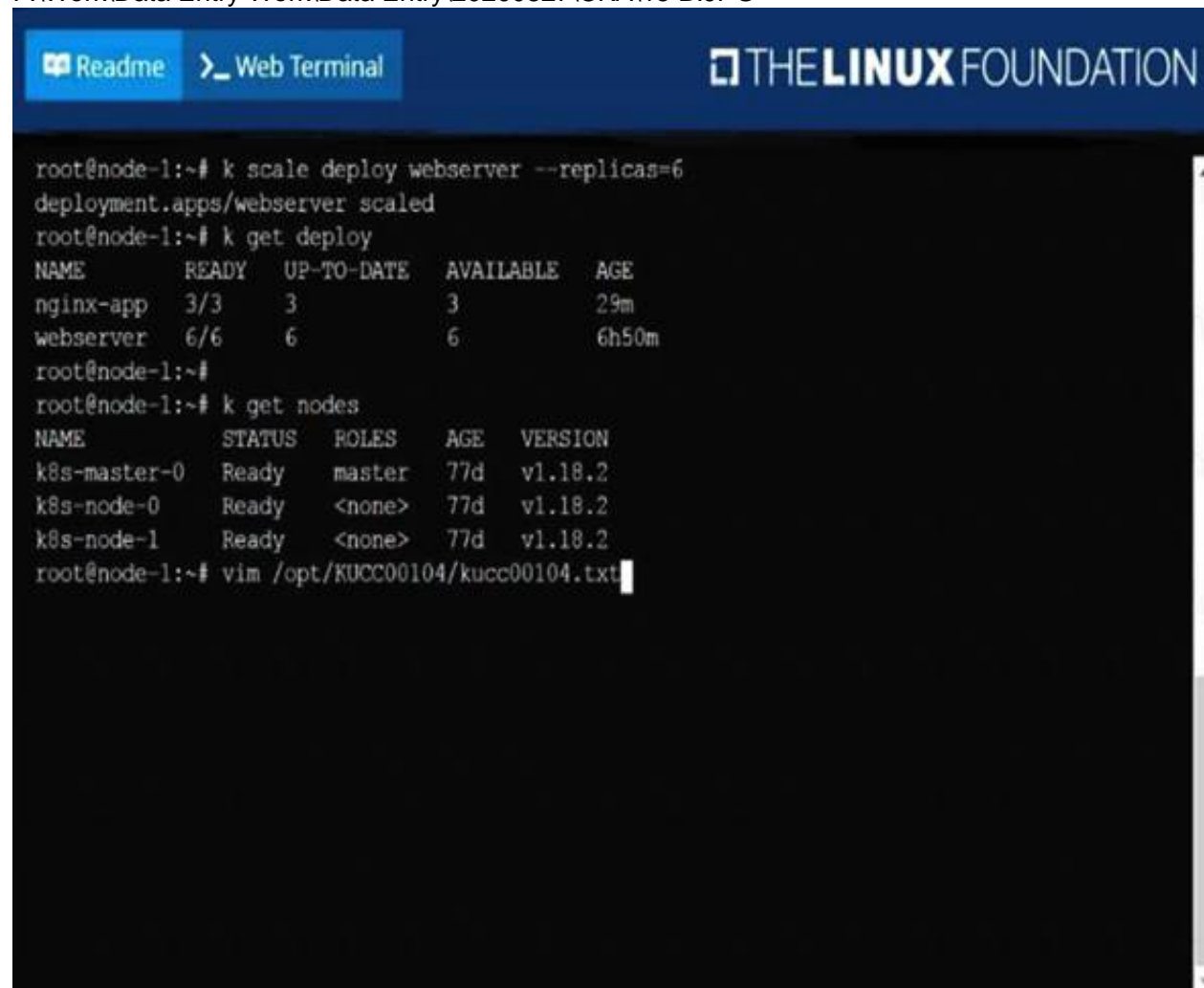
Check to see how many worker nodes are ready (not including nodes taintedNoSchedule) and write the number to `to/opt/KUCC00104/kucc00104.txt`.

- A. Mastered
B. Not Mastered

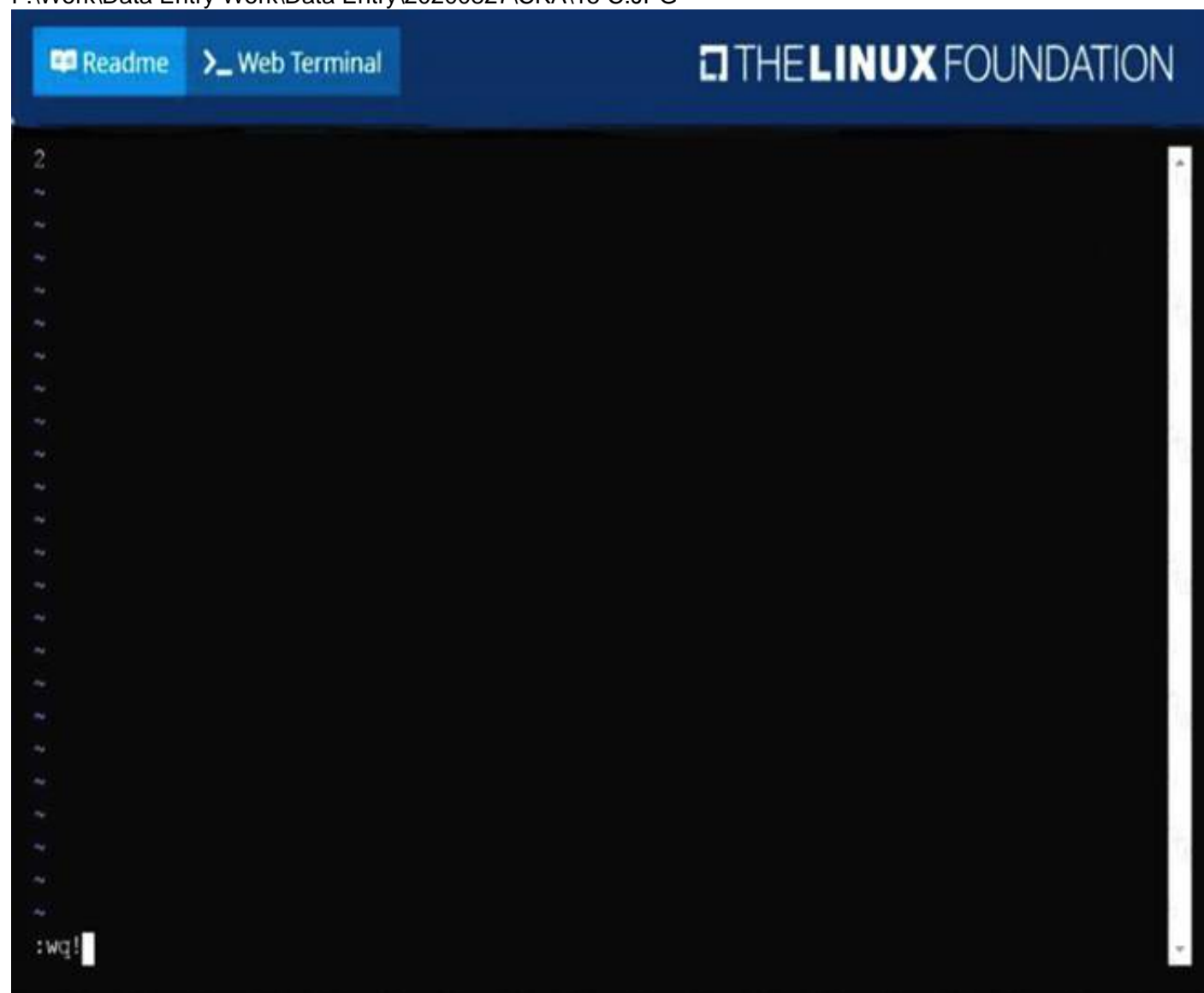
Answer: A

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



• • • • •

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>