

# Linux-Foundation

## Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program



**NEW QUESTION 1**

Create a deployment spec file that will:

- > Launch 7 replicas of the nginx image with the label app\_runtime\_stage=dev
- > deployment name: kual00201

Save a copy of this spec file to /opt/KUAL00201/spec\_deployment.yaml (or /opt/KUAL00201/spec\_deployment.json).  
 When you are done, clean up (delete) any new Kubernetes API object that you produced during this task.

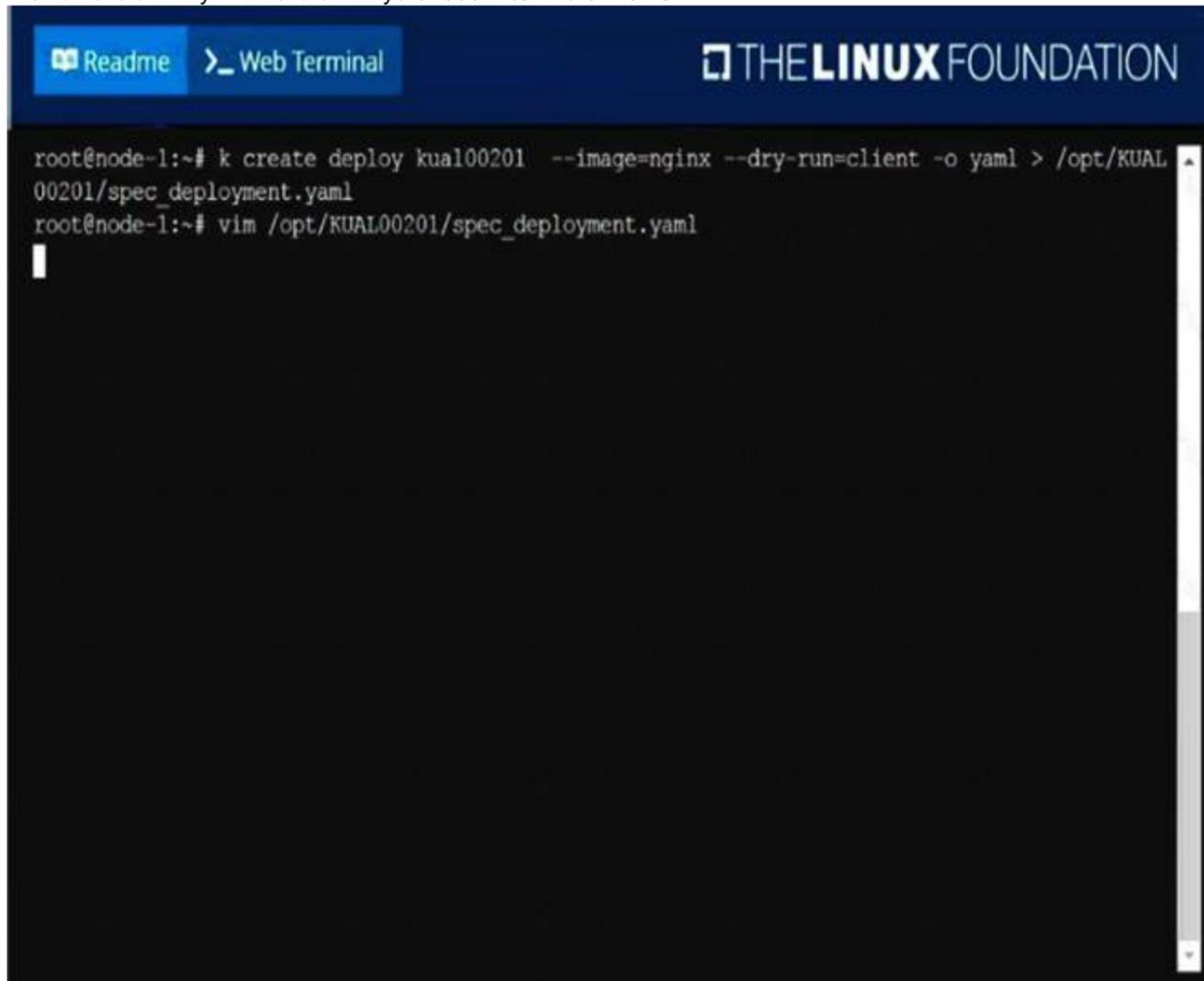
- A. Mastered
- B. Not Mastered

**Answer:** A

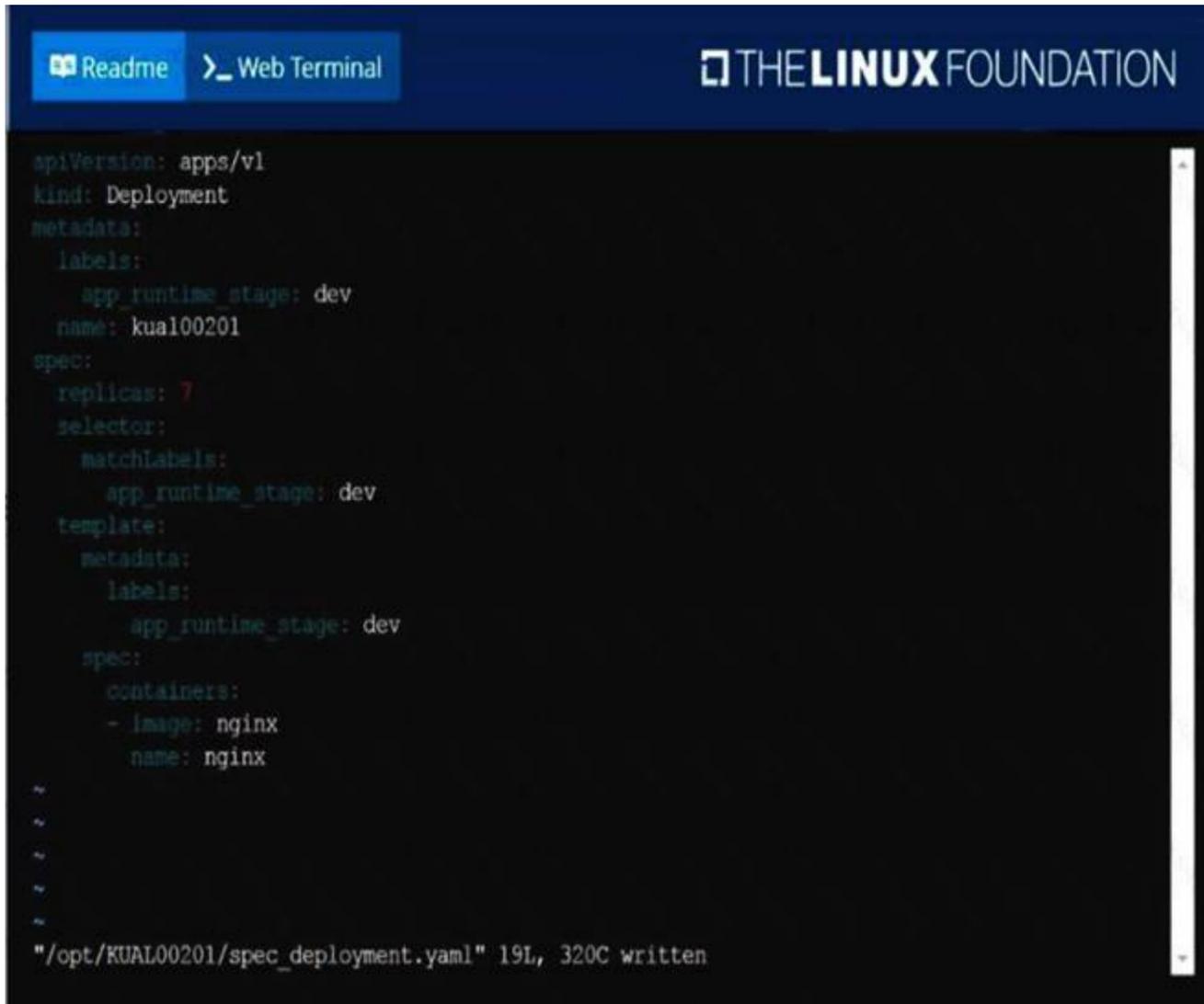
**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\10 B.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\10 C.JPG



```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app_runtime_stage: dev
  name: kua100201
spec:
  replicas: 7
  selector:
    matchLabels:
      app_runtime_stage: dev
  template:
    metadata:
      labels:
        app_runtime_stage: dev
    spec:
      containers:
      - image: nginx
        name: nginx
  ~
  ~
  ~
  ~
  ~
"/opt/KUAL00201/spec_deployment.yaml" 19L, 320C written

```

**NEW QUESTION 2**

Create a deployment as follows:

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

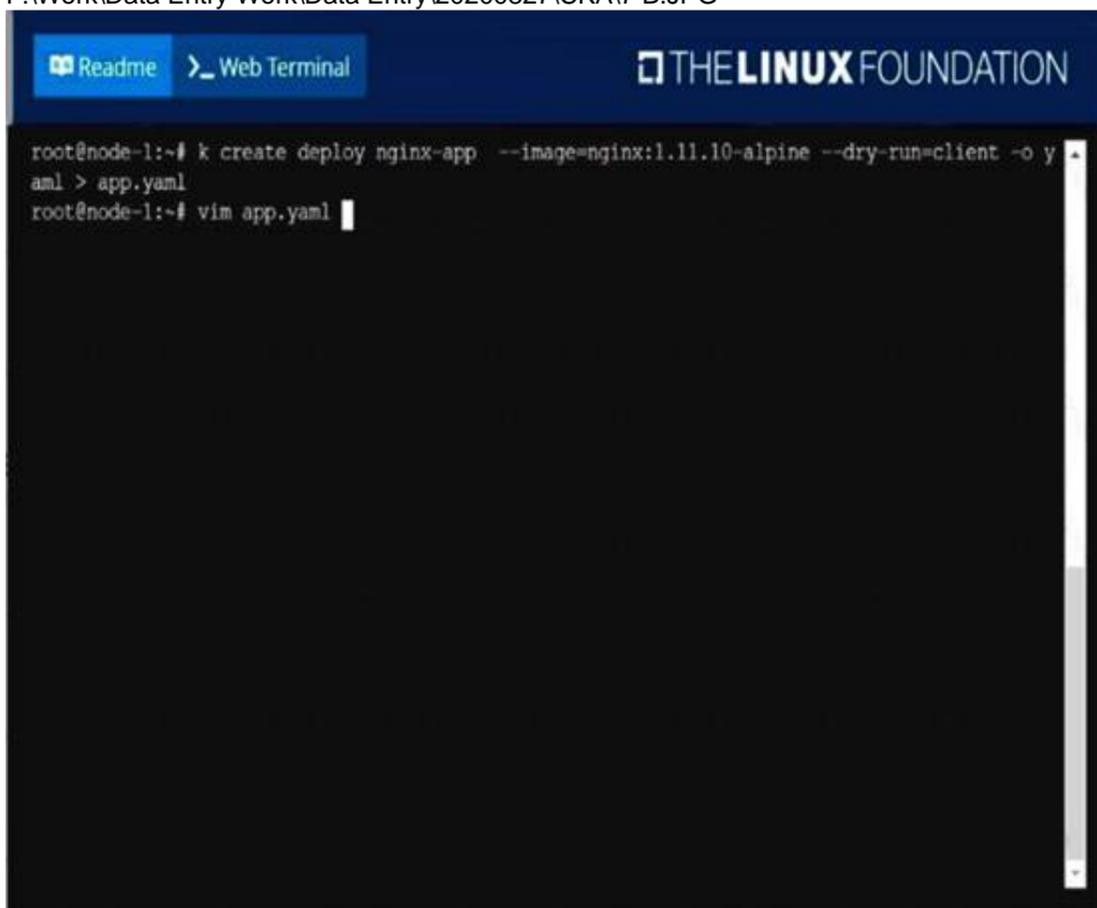
Next, deploy the application with newversion1.11.13-alpine, byperforming a rolling update.  
 Finally, rollback that update to theprevious version1.11.10-alpine.

- A. Mastered
- B. Not Mastered

**Answer: A**

**Explanation:**

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

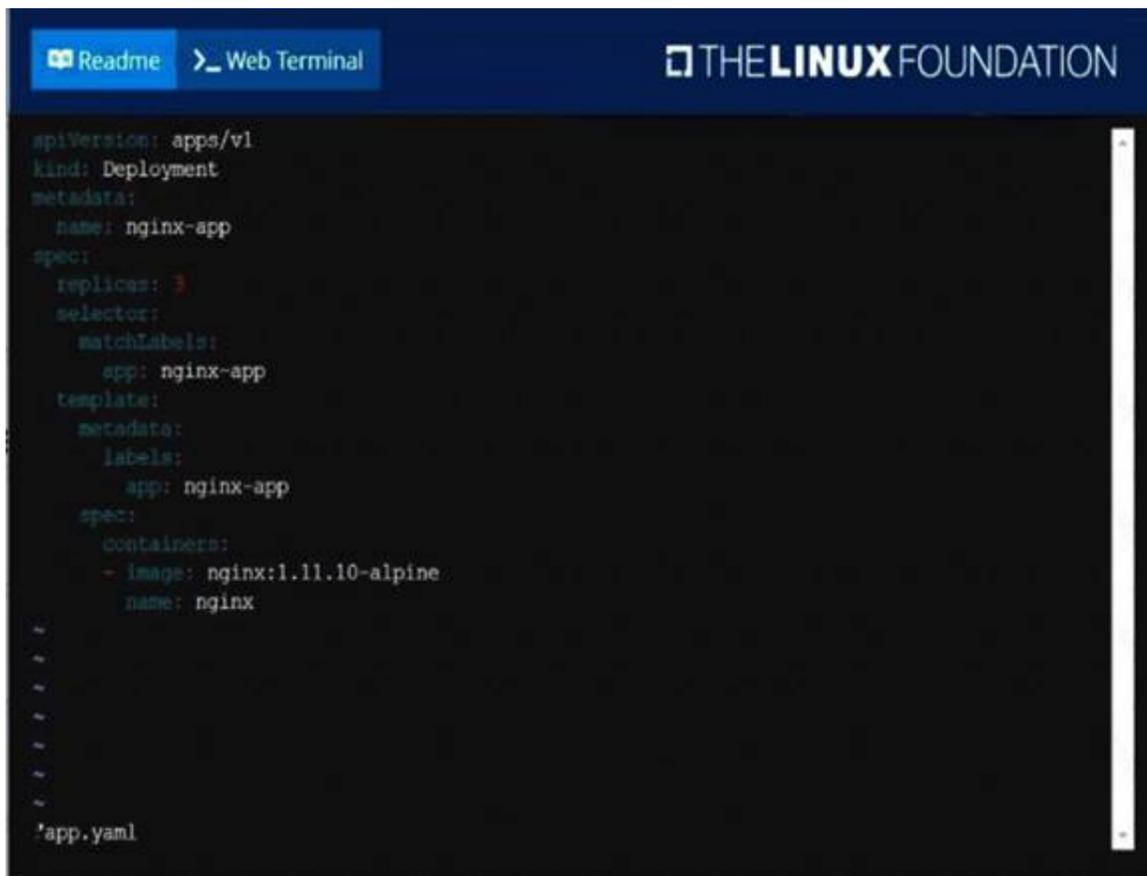


```

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

```

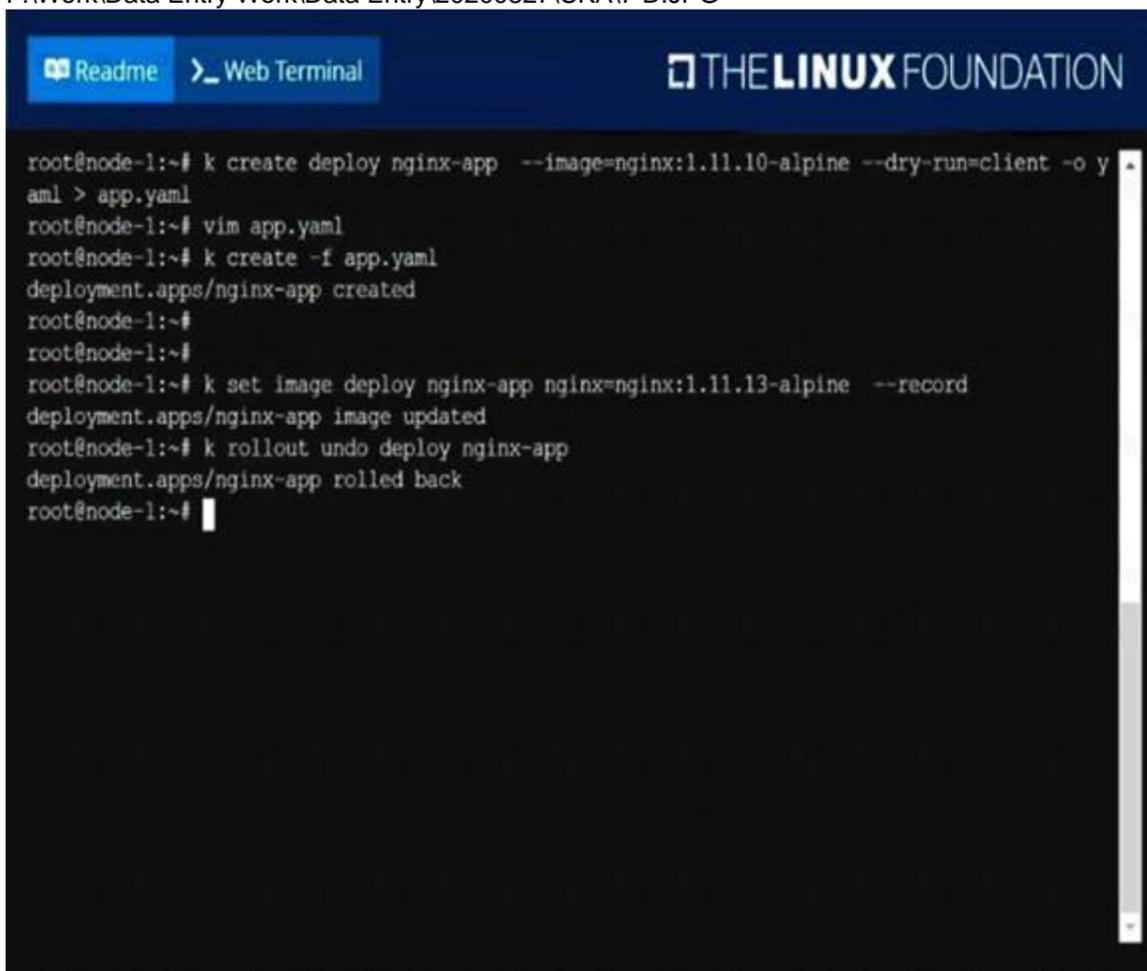
F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 C.JPG



```

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 3**

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

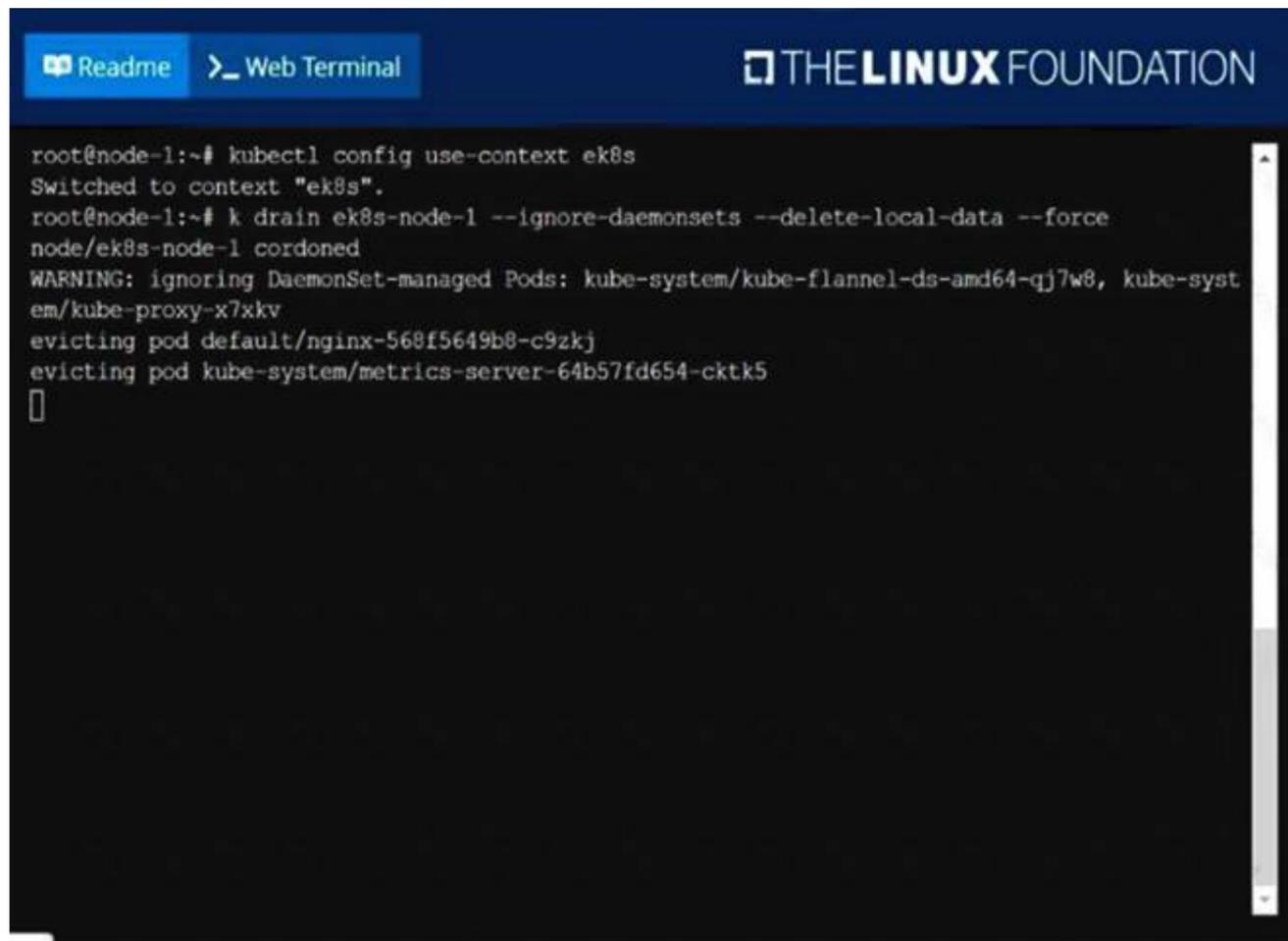
- A. Mastered
- B. Not Mastered

**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: # defines the capacity of PV we are creating
  storage: 2Gi # the amount of storage we are trying 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 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: share

```

"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-265411)$ 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 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: v1

kind: Pod

metadata: creationTimestamp: null name: app-data

spec: volumes: - name: config persistentVolumeClaim: claimName: app-data

containers: - image: nginx name: app volumeMounts: - mountPath: "/srv/app-data" name: config

**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**

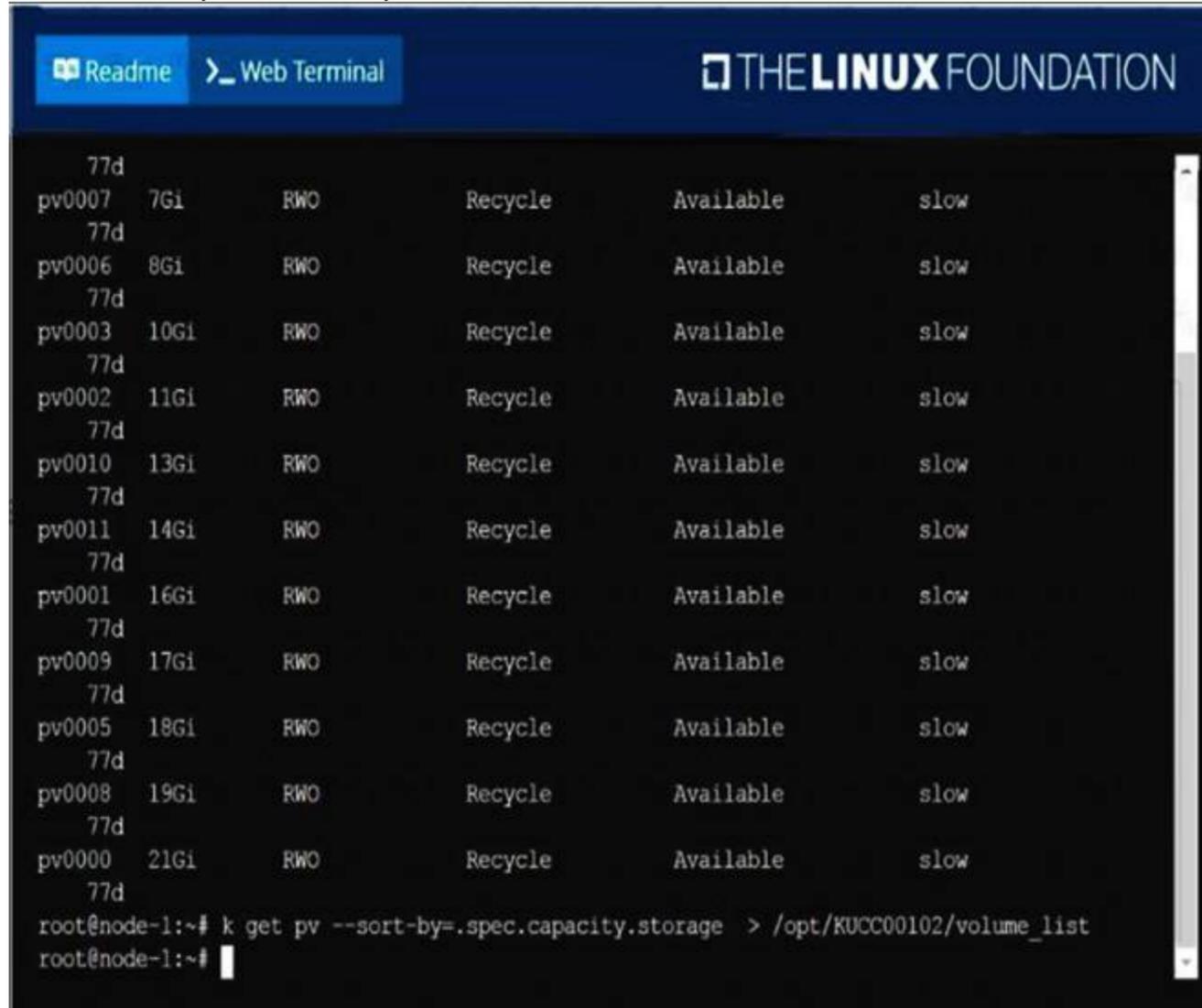
List all persistent volumes sorted by capacity, saving the fullkubectloutput to /opt/KUCC00102/volume\_list. Usekubectl 's own functionality forsorting the output, and do not manipulate it any further.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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



**NEW QUESTION 7**

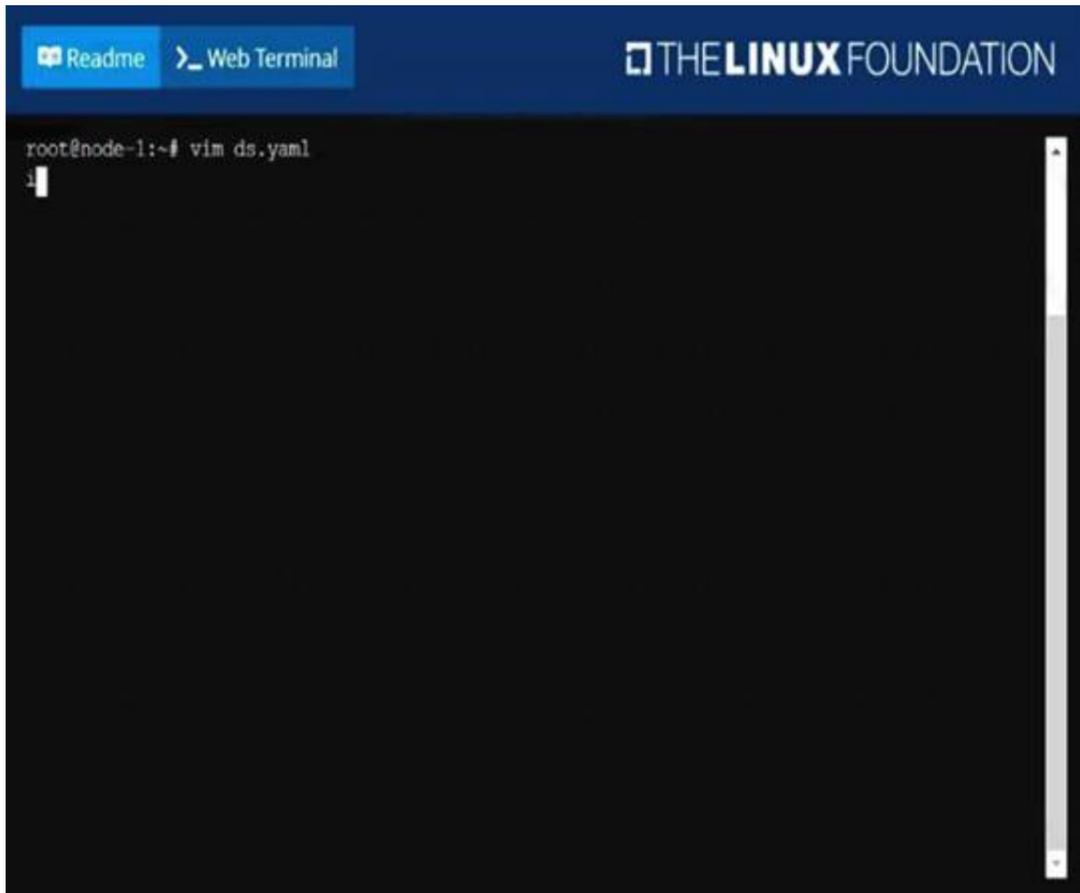
Ensure a single instance of podnginxis running on each node of theKubernetes cluster wherenginxalso represents the Image name whichhas to be used. Do not override anytaints currently in place. UseDaemonSetto complete thistask and usesd-kusc00201asDaemonSet name.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

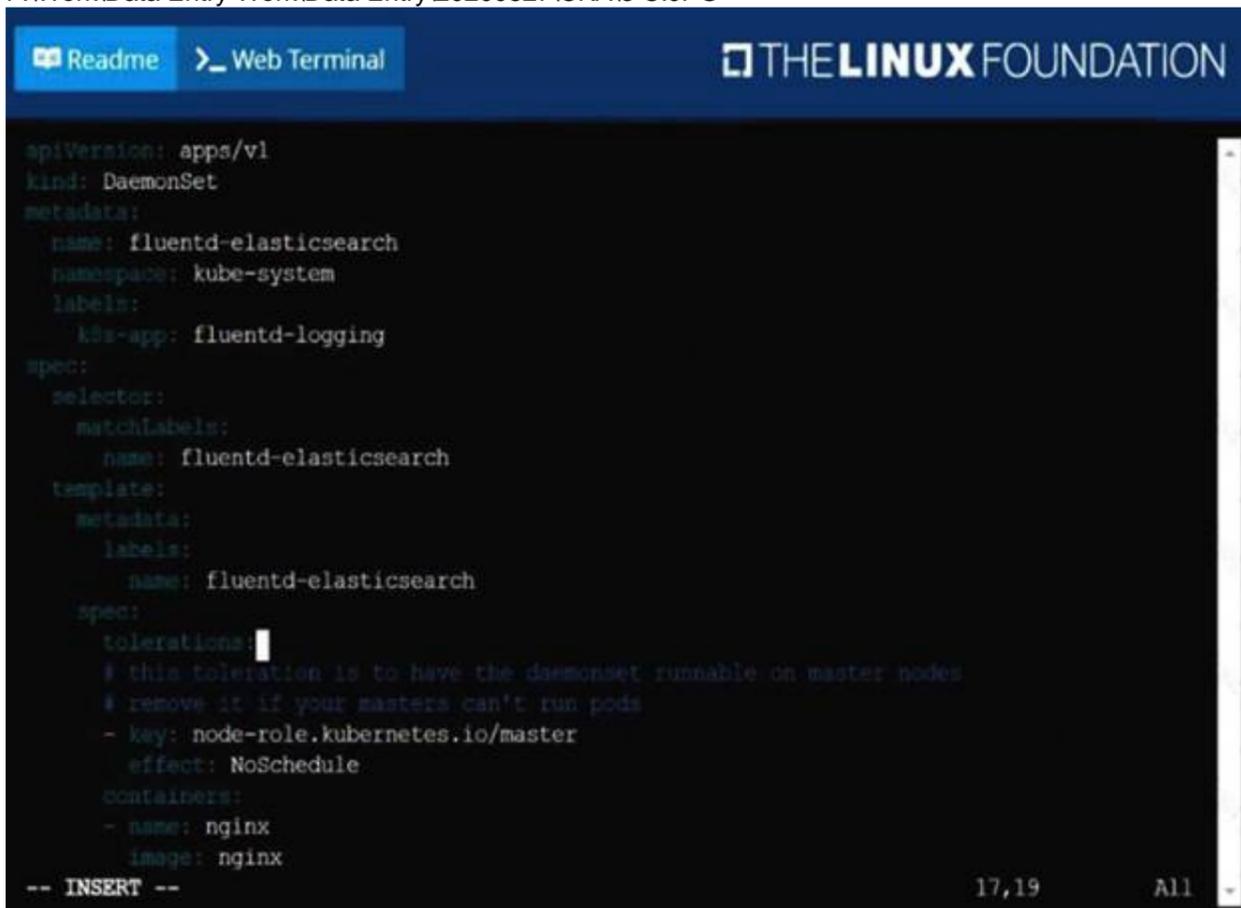
solution  
 F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 B.JPG



Readme Web Terminal THE LINUX FOUNDATION

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

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 C.JPG



Readme Web Terminal THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      tolerations:
        # this toleration is to have the daemonset runnable on master nodes
        # remove it if your masters can't run pods
        - key: node-role.kubernetes.io/master
          effect: NoSchedule
      containers:
        - name: nginx
          image: nginx
-- INSERT -- 17,19 All
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 D.JPG





**NEW QUESTION 9**

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

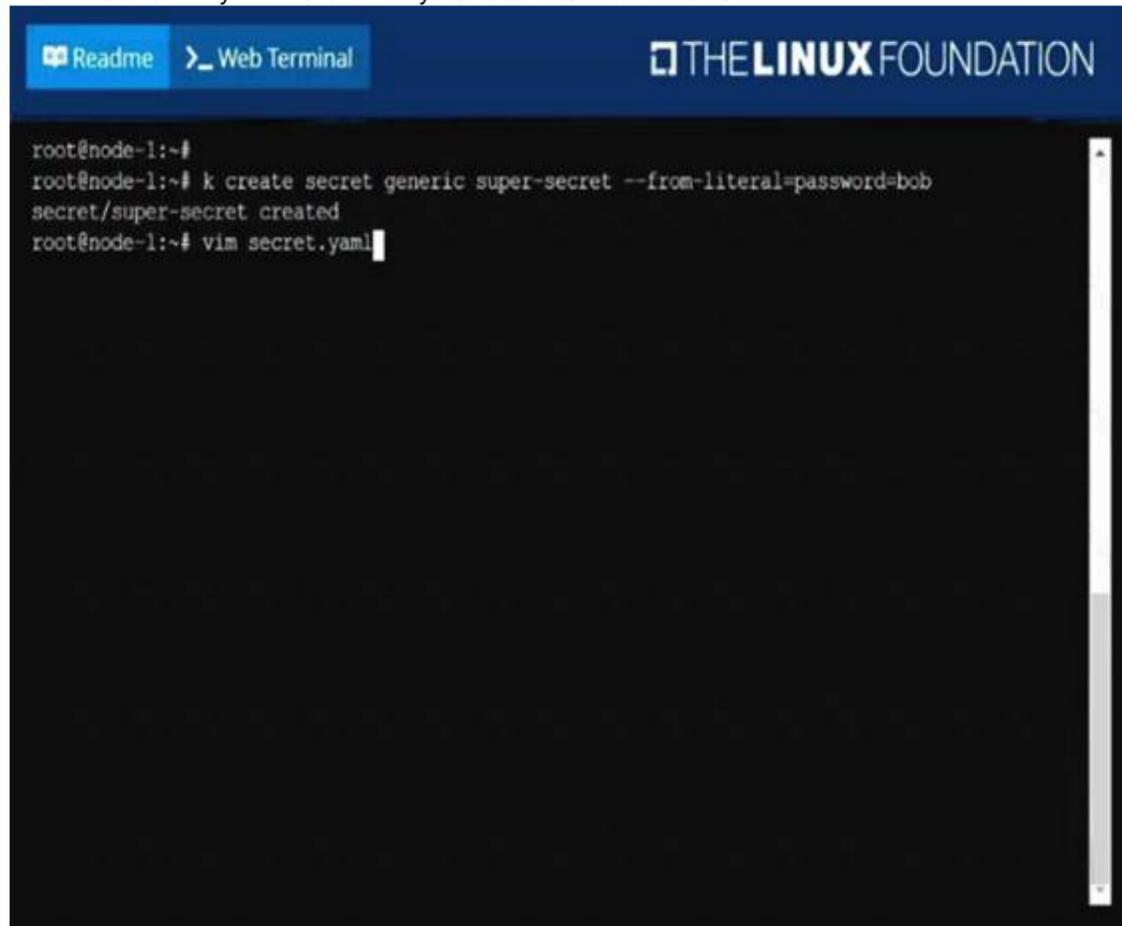
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

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

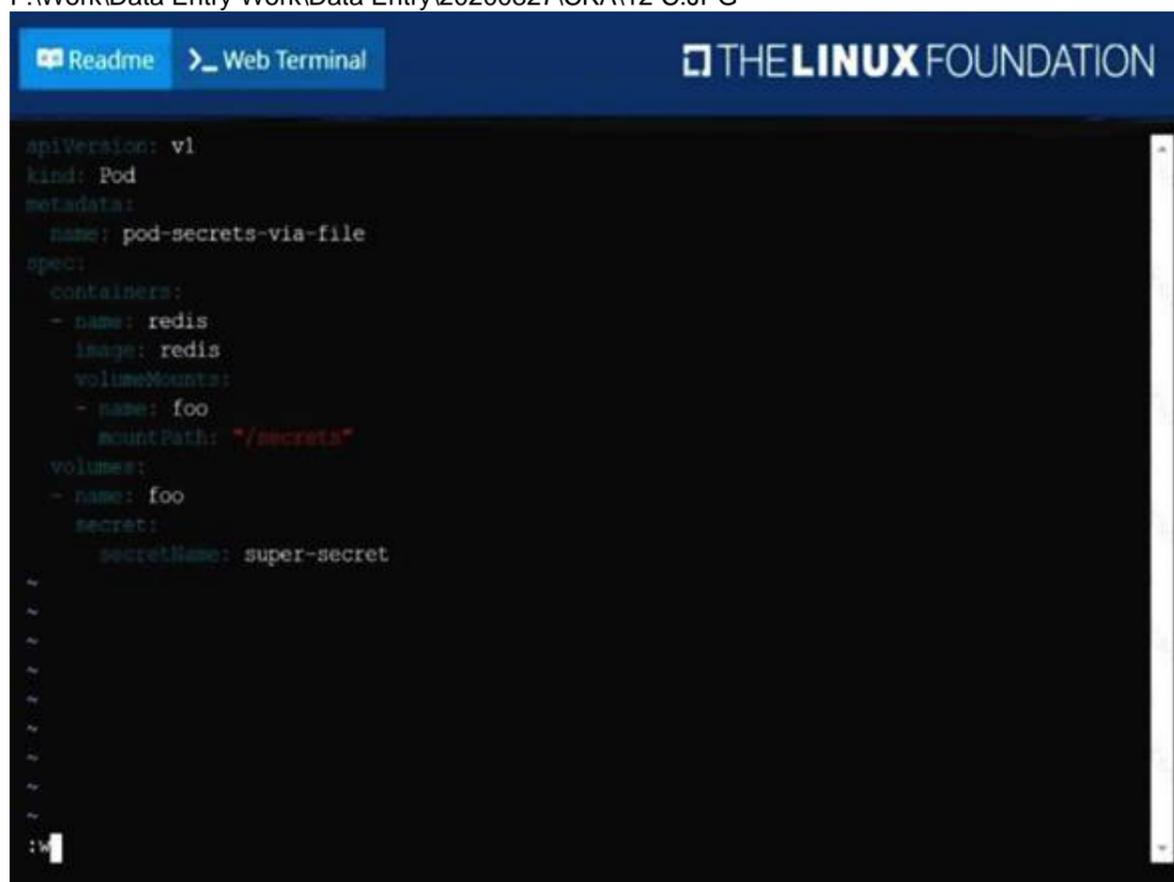


The screenshot shows a terminal window with the following commands and output:

```

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 C.JPG



The screenshot shows a terminal window displaying the contents of a pod manifest file:

```

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
  
```

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

Readme
Web Terminal
THE LINUX FOUNDATION

```

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-t4791         1/1     Running   0           6h43m
root@node-1:~#

```

**NEW QUESTION 10**

Perform the following tasks:

- > Add an init container to hungry-bear (which has been defined in spec file /opt/KUCC00108/pod-spec-KUCC00108.yaml)
- > The init container should create an empty file named /workdir/calm.txt
- > If /workdir/calm.txt is not detected, the pod should exit
- > Once the spec file has been updated with the init container definition, the pod should be created

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

```

Readme Web Terminal THE LINUX FOUNDATION

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

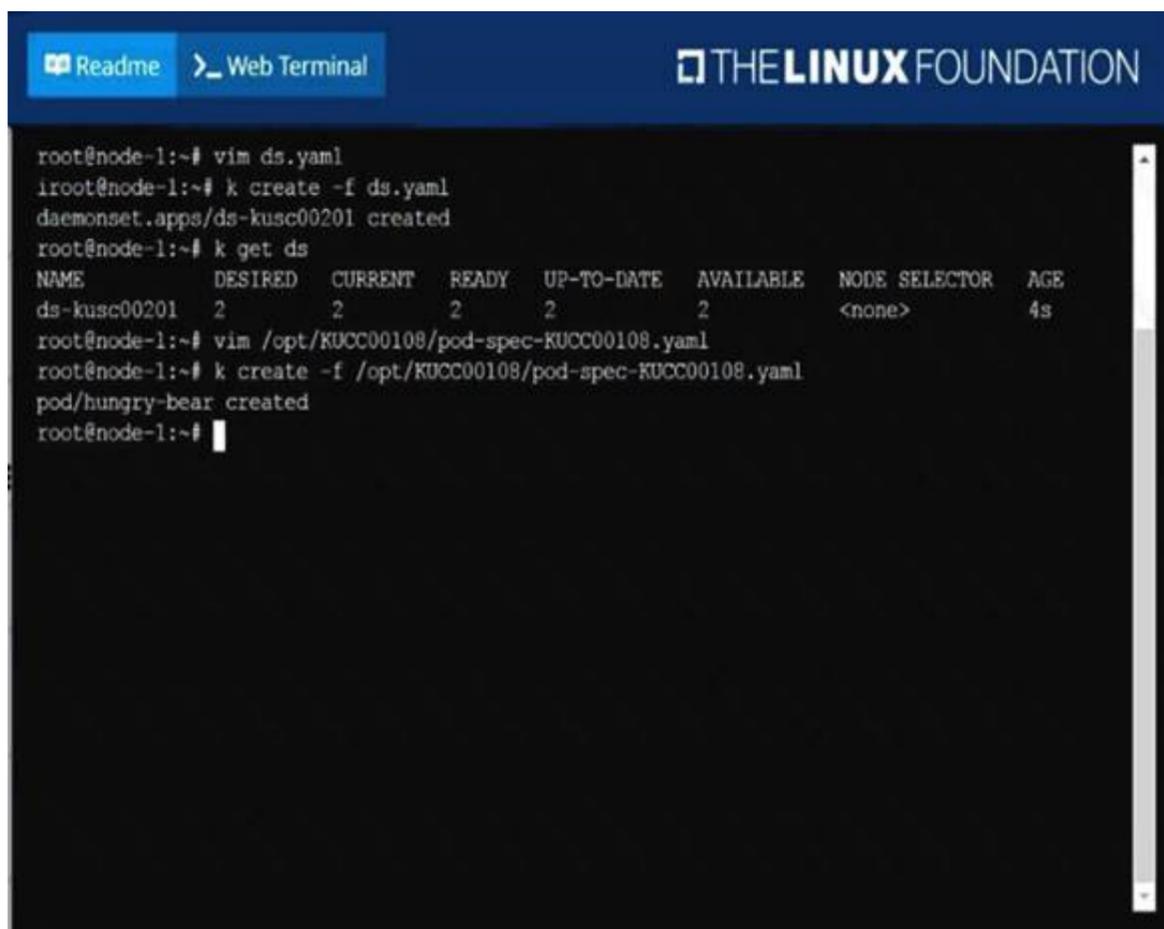
```

Readme Web Terminal THE LINUX FOUNDATION

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/caln.txt ]; then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
  initContainers:
  - name: create
    image: alpine
    command: ["/bin/sh", "-c", "touch /workdir/caln.txt"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
:~#

```

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



```

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
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~#

```

**NEW QUESTION 10**

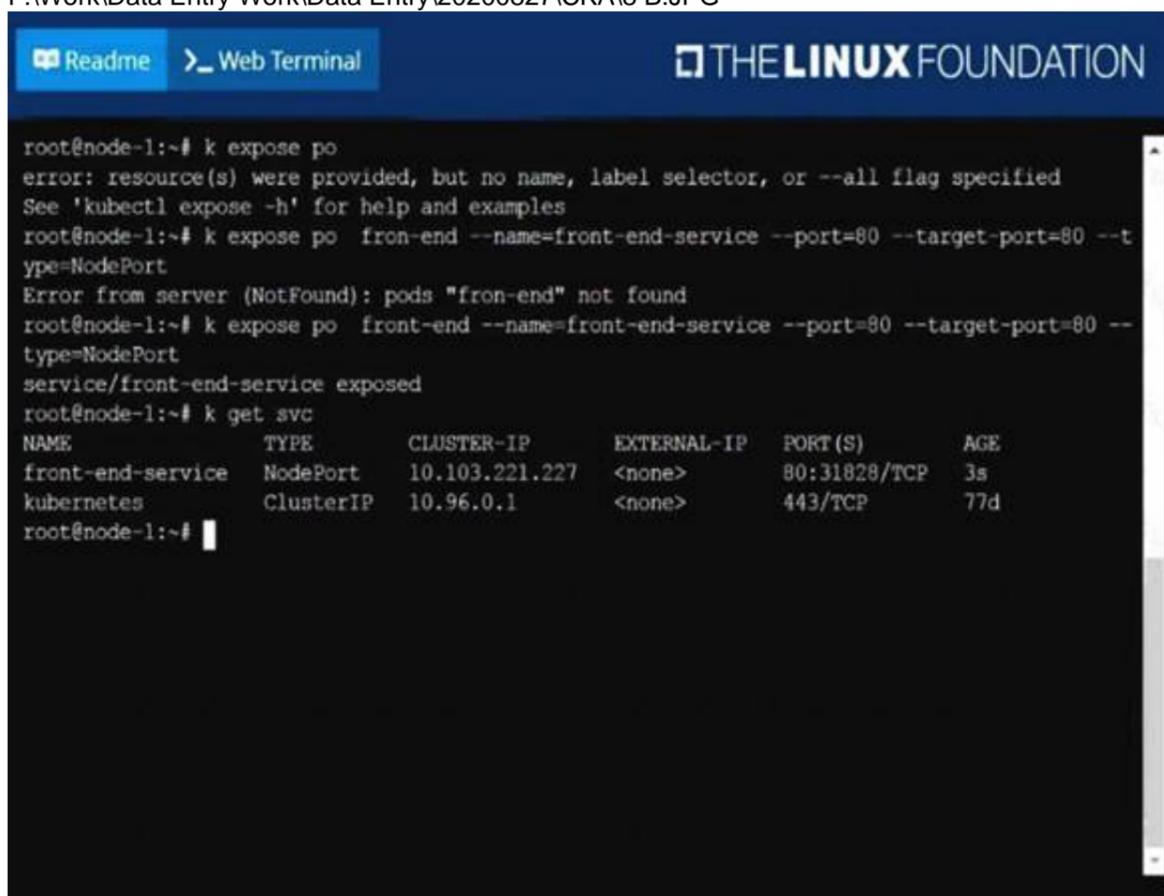
Create and configure the service front-end-services so it's accessible through NodePort and routes to the existing pod named front-end.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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



```

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 11**

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 16**

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 17**

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

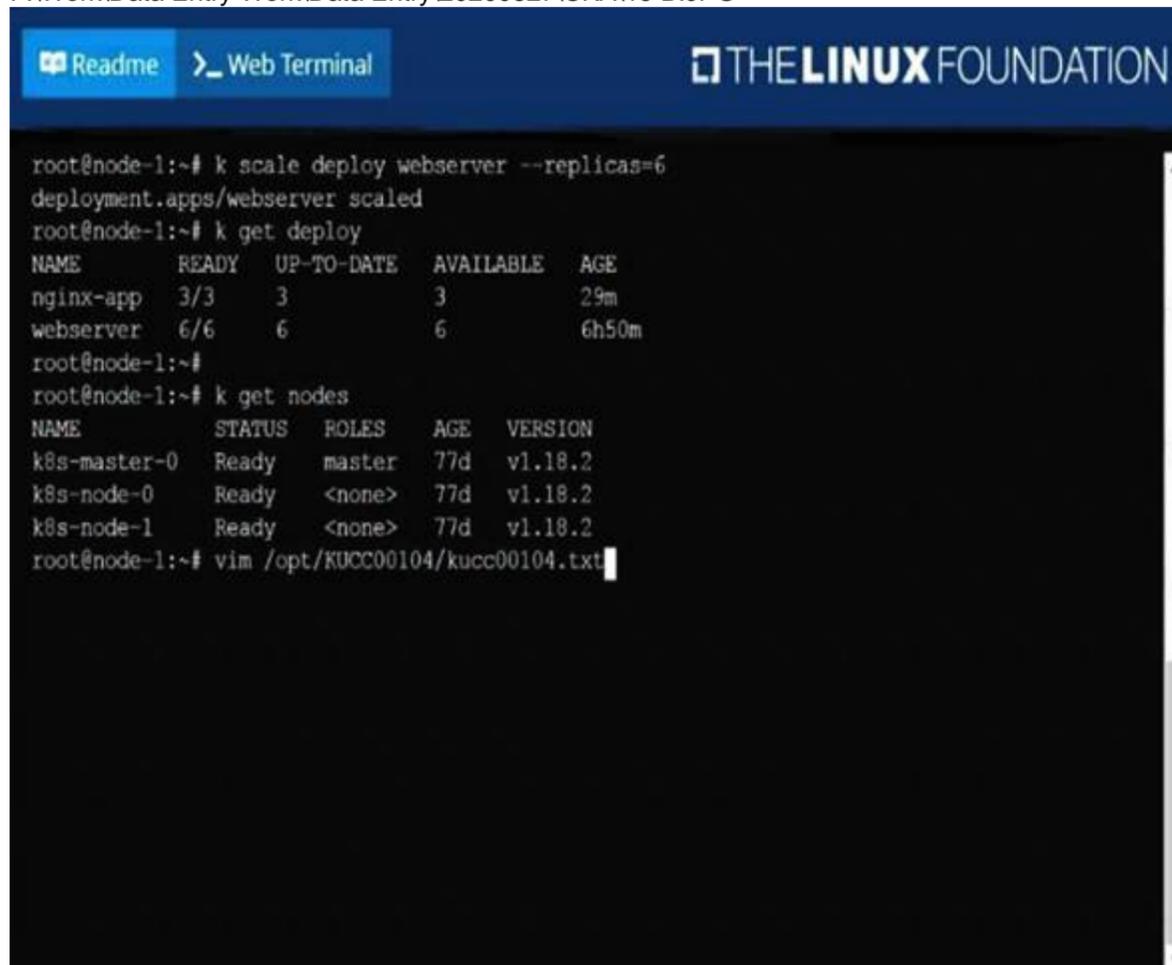
- A. Mastered
- B. Not Mastered

**Answer:** A

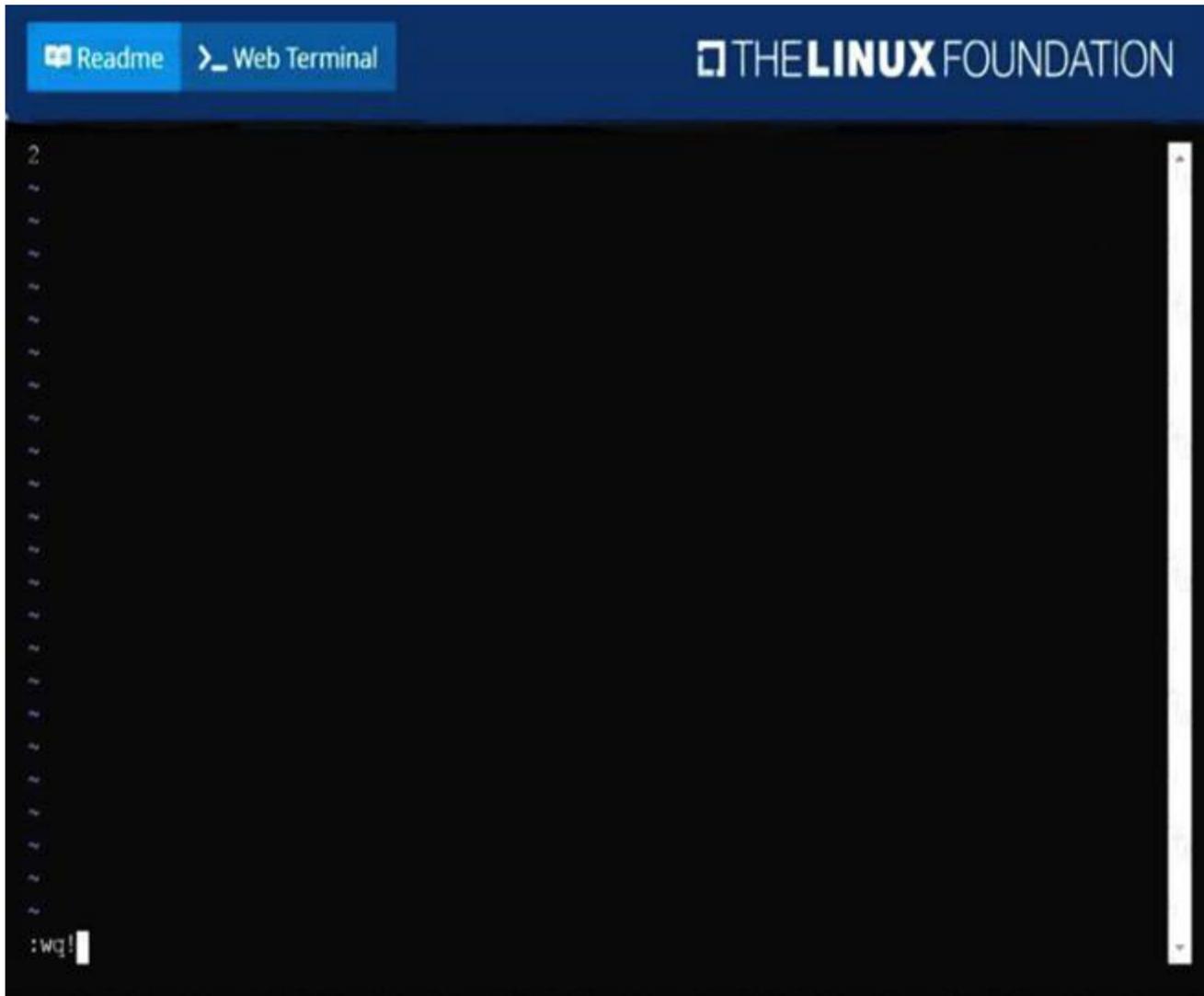
**Explanation:**

solution

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



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



**NEW QUESTION 18**

.....

## **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](#)**