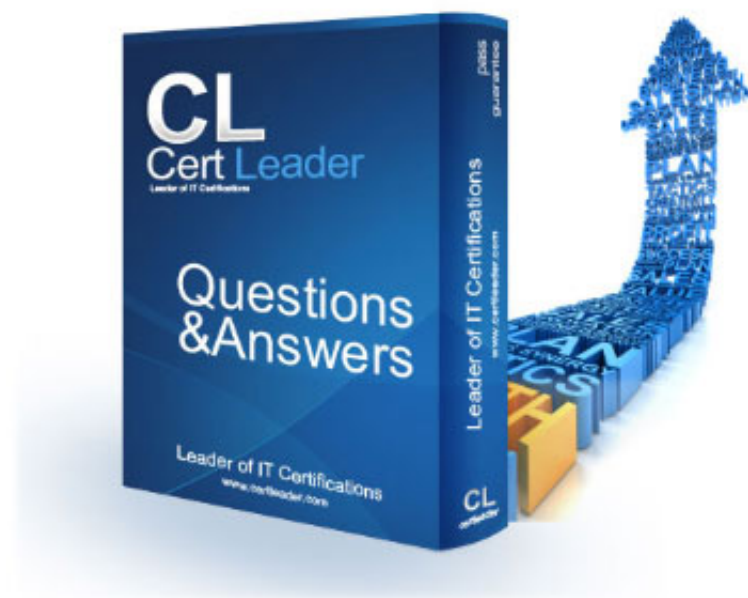


CKA Dumps

Certified Kubernetes Administrator (CKA) Program

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



NEW QUESTION 1

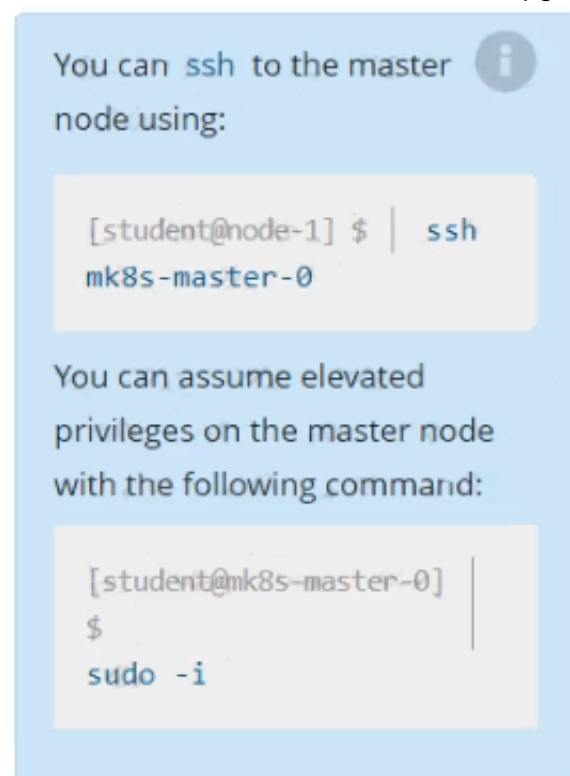
Score: 7%



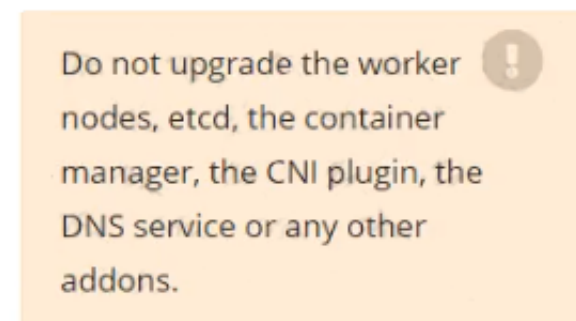
Task

Given an existing Kubernetes cluster running version 1.20.0, upgrade all of the Kubernetes control plane and node components on the master node only to version 1.20.1.

Be sure to drain the master node before upgrading it and uncordon it after the upgrade.



You are also expected to upgrade kubelet and kubectl on the master node.



- A. Mastered
- B. Not Mastered

Answer: A**Explanation:**

SOLUTION:

```
[student@node-1] > ssh ek8s
```

```
kubectl cordon k8s-master
```

```
kubectl drain k8s-master --delete-local-data --ignore-daemonsets --force
```

```
apt-get install kubeadm=1.20.1-00 kubelet=1.20.1-00 kubectl=1.20.1-00 --disableexcludes=kubernetes kubeadm upgrade apply 1.20.1 --etcd-upgrade=false
```

```
systemctl daemon-reload systemctl restart kubelet kubectl uncordon k8s-master
```

NEW QUESTION 2

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 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:

Kubectll logs frontend | grep -i “started” > /opt/error-logs

NEW QUESTION 4

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 5

Monitor the logs of pod foo and:

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



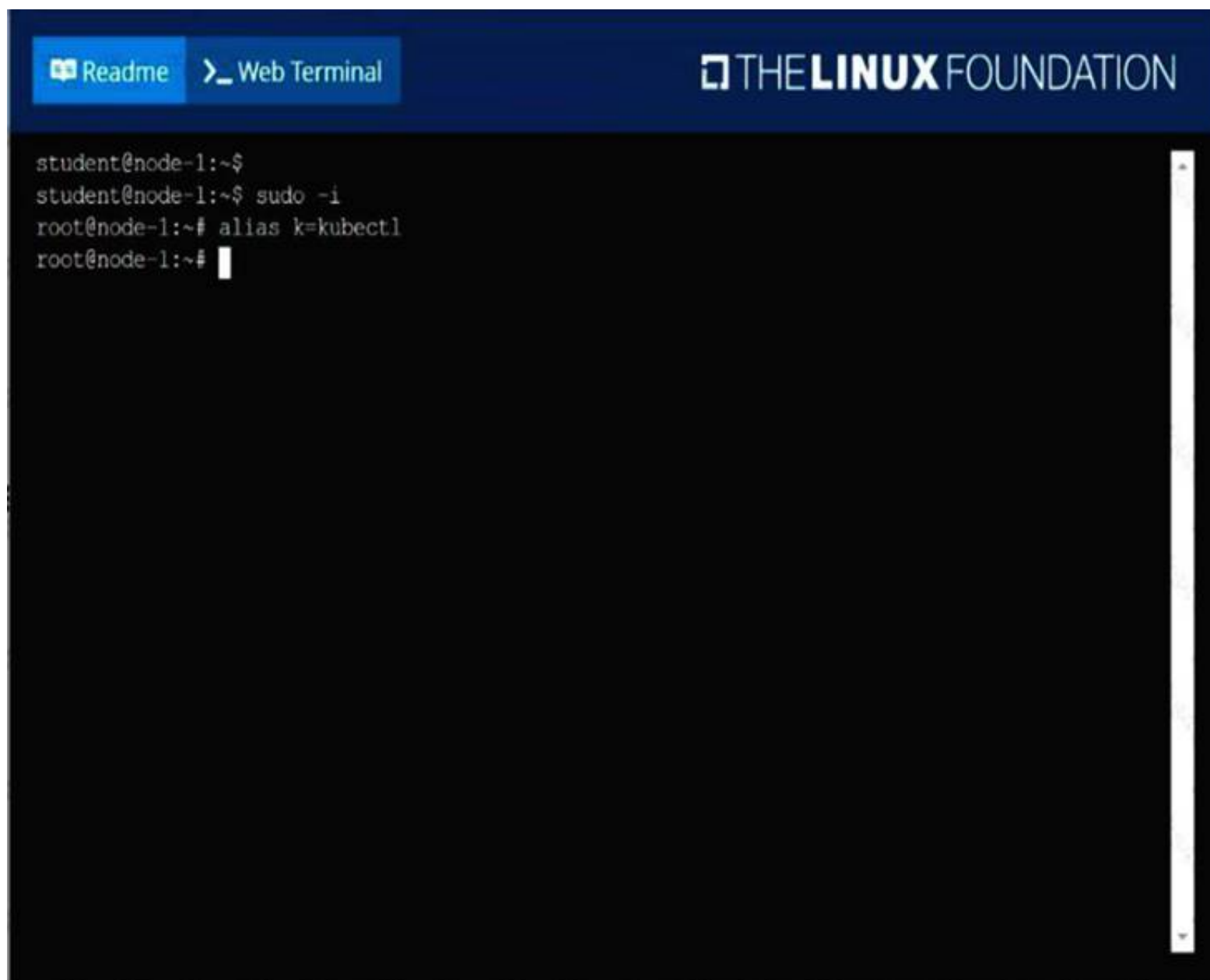
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

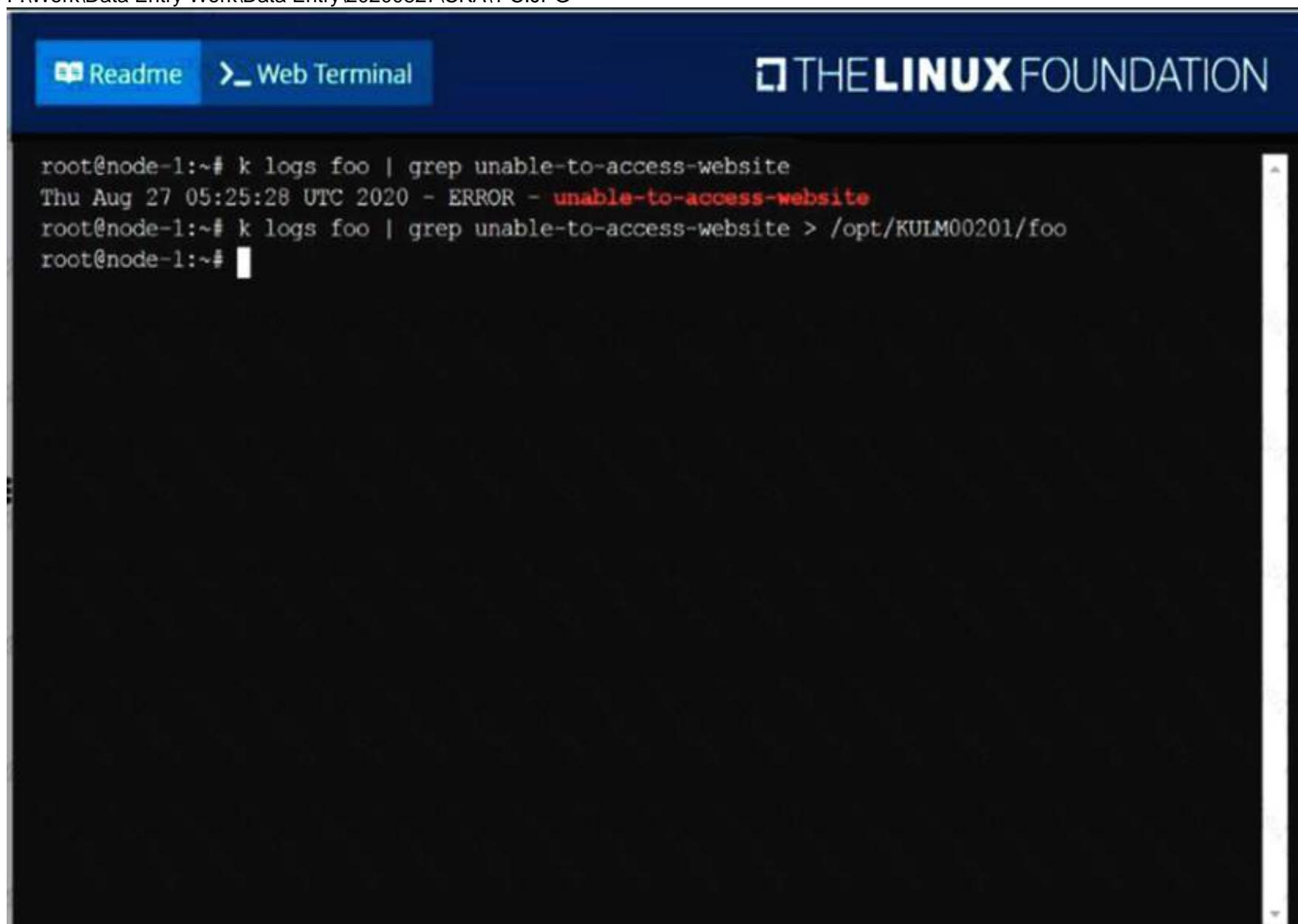
solution

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



```
student@node-1:~$  
student@node-1:~$ sudo -i  
root@node-1:~# alias k=kubect1  
root@node-1:~#
```

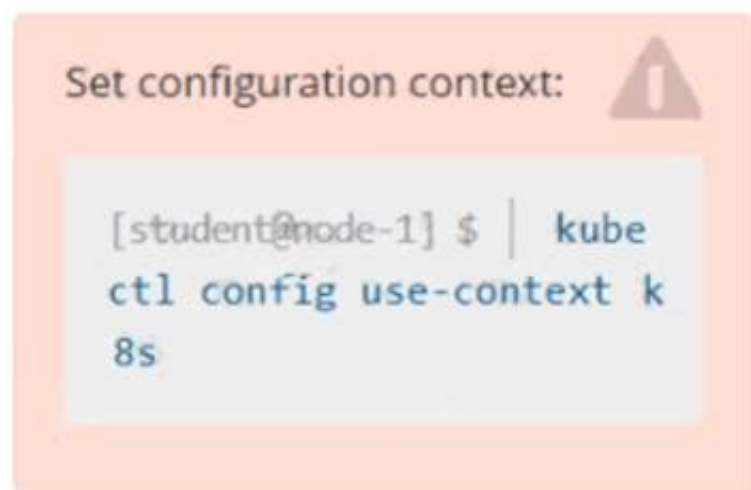
F:\Work\Data Entry Work\Data Entry\20200827\CKA\1 C.JPG



```
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:~#
```

NEW QUESTION 6

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 7

Create a deployment as follows:

- > Name: nginx-app
- > Using container nginx with version 1.11.10-alpine
- > The deployment should contain 3 replicas

Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update.

Finally, rollback that update to the previous version 1.11.10-alpine.

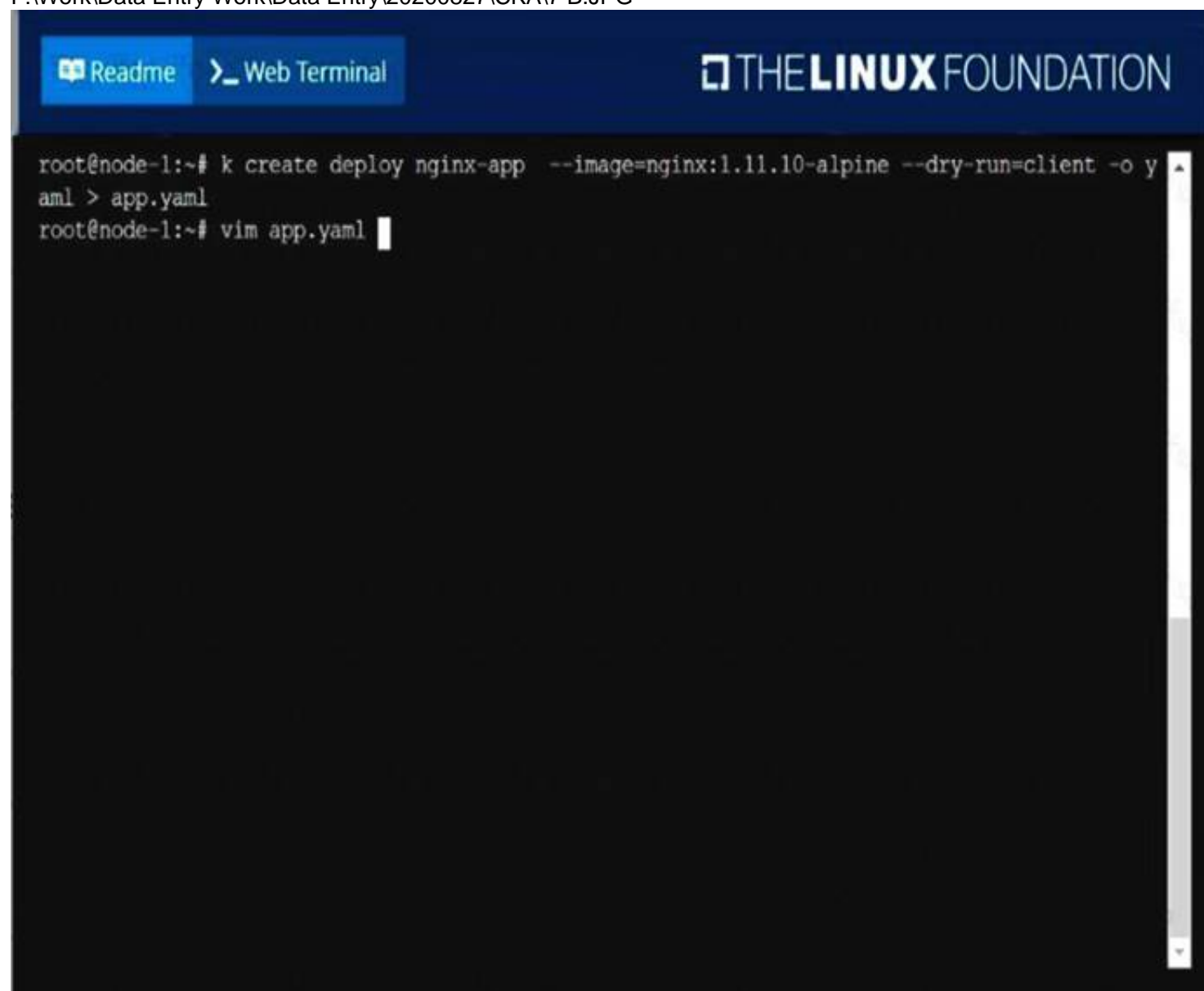
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

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



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



Create a deployment spec file that will:

- ```
> Launch 7 replicas of the nginx Image with the labelapp_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.



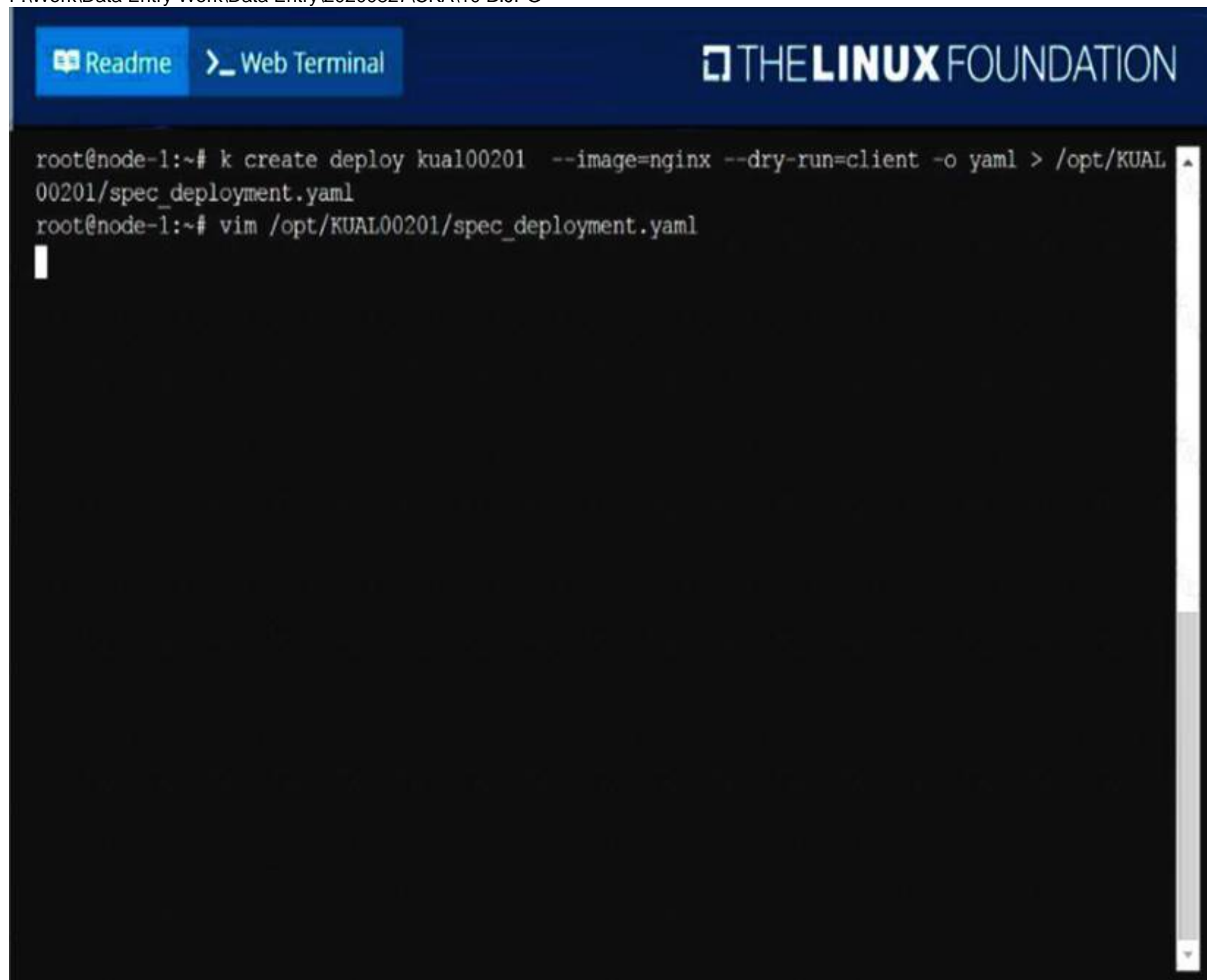
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

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



The screenshot shows a web terminal window with a dark blue header. On the left, there are two buttons: 'Readme' with a book icon and 'Web Terminal' with a terminal icon. On the right, the 'THE LINUX FOUNDATION' logo is displayed. The terminal area has a black background with white text. The commands entered are: `root@node-1:~# k create deploy kual00201 --image=nginx --dry-run=client -o yaml > /opt/KUAL00201/spec_deployment.yaml` and `root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml`. A vertical scrollbar is visible on the right side of the terminal area.

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

Readme
Web Terminal

```

apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
 app_runtime_stage: dev
 name: kual00201
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 9

Score: 4%



Task

Schedule a pod as follows:

- Name: nginx-kusc00401
- Image: nginx
- Node selector: disk=ssd

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

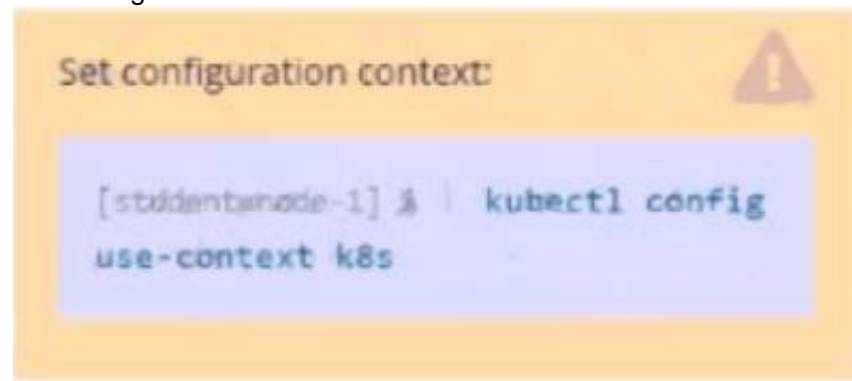
#yaml apiVersion: v1 kind: Pod metadata:
name: nginx-kusc00401 spec:
containers:
- name: nginx image: nginx
imagePullPolicy: IfNotPresent nodeSelector:
disk: spinning
#
kubectl create -f node-select.yaml

```



**NEW QUESTION 10**

Task Weight: 4%



Task

Schedule a Pod as follows:

- Name: kucc1
- App Containers: 2
- Container Name/Images: o nginx
- o consul

- A. Mastered  
B. Not Mastered

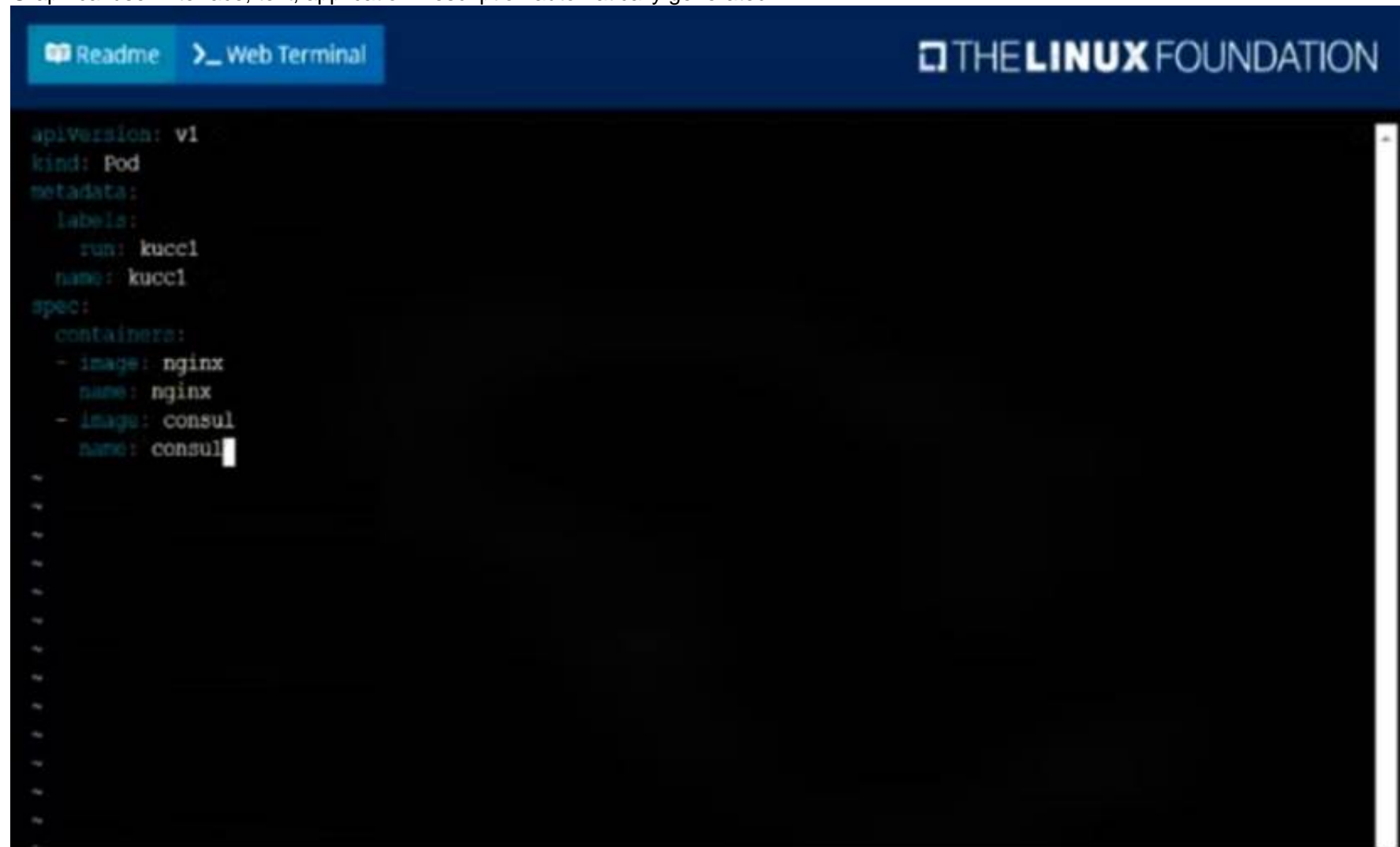
**Answer:** A

**Explanation:**

Solution:

```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl run kucc1 --image=nginx --dry-run=client -o yaml > aa.y
```

Graphical user interface, text, application Description automatically generated



Text Description automatically generated

```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl run kuccl --image=nginx --dry-run=client -o yaml > aa.yaml
student@node-1:~$ vim aa.yaml
student@node-1:~$ kubectl create -f aa.yaml
pod/kuccl created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
ll-factor-app 1/1 Running 0 6h34m
cpu-loader-98b9se 1/1 Running 0 6h33m
cpu-loader-ab2d3s 1/1 Running 0 6h33m
cpu-loader-kipb9a 1/1 Running 0 6h33m
foobar 1/1 Running 0 6h34m
front-end-6bc87b9748-24rcm 1/1 Running 0 5m4s
front-end-6bc87b9748-hd5wp 1/1 Running 0 5m2s
kuccl 0/2 ContainerCreating 0 3s
nginx-kusc00401 1/1 Running 0 2m28s
webserver-84c89dfd75-2dljn 1/1 Running 0 6h38m
webserver-84c89dfd75-8d8x2 1/1 Running 0 6h38m
webserver-84c89dfd75-z5zz4 1/1 Running 0 3m51s
student@node-1:~$
```

NEW QUESTION 10

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 11

Check the image version in pod without the describe command

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubectl get po nginx -o jsonpath='{.spec.containers[0].image}'

NEW QUESTION 12

Create a pod as follows:

- > Name: non-persistent-redis
- > container Image: redis
- > Volume with name: cache-control
- > Mount path: /data/redis

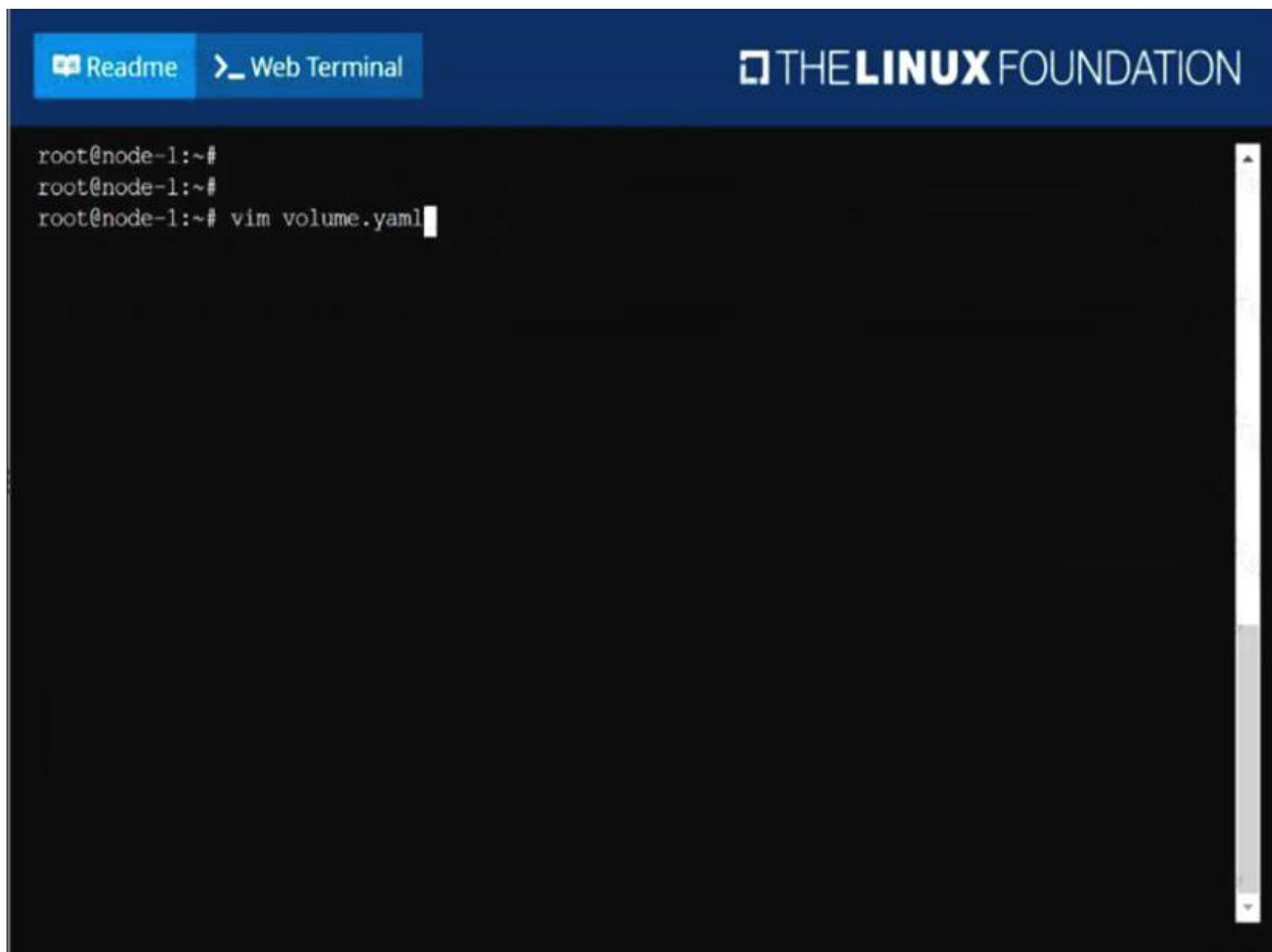
The pod should launch in the staging namespace and the volume must not be persistent.

- A. Mastered
- B. Not Mastered

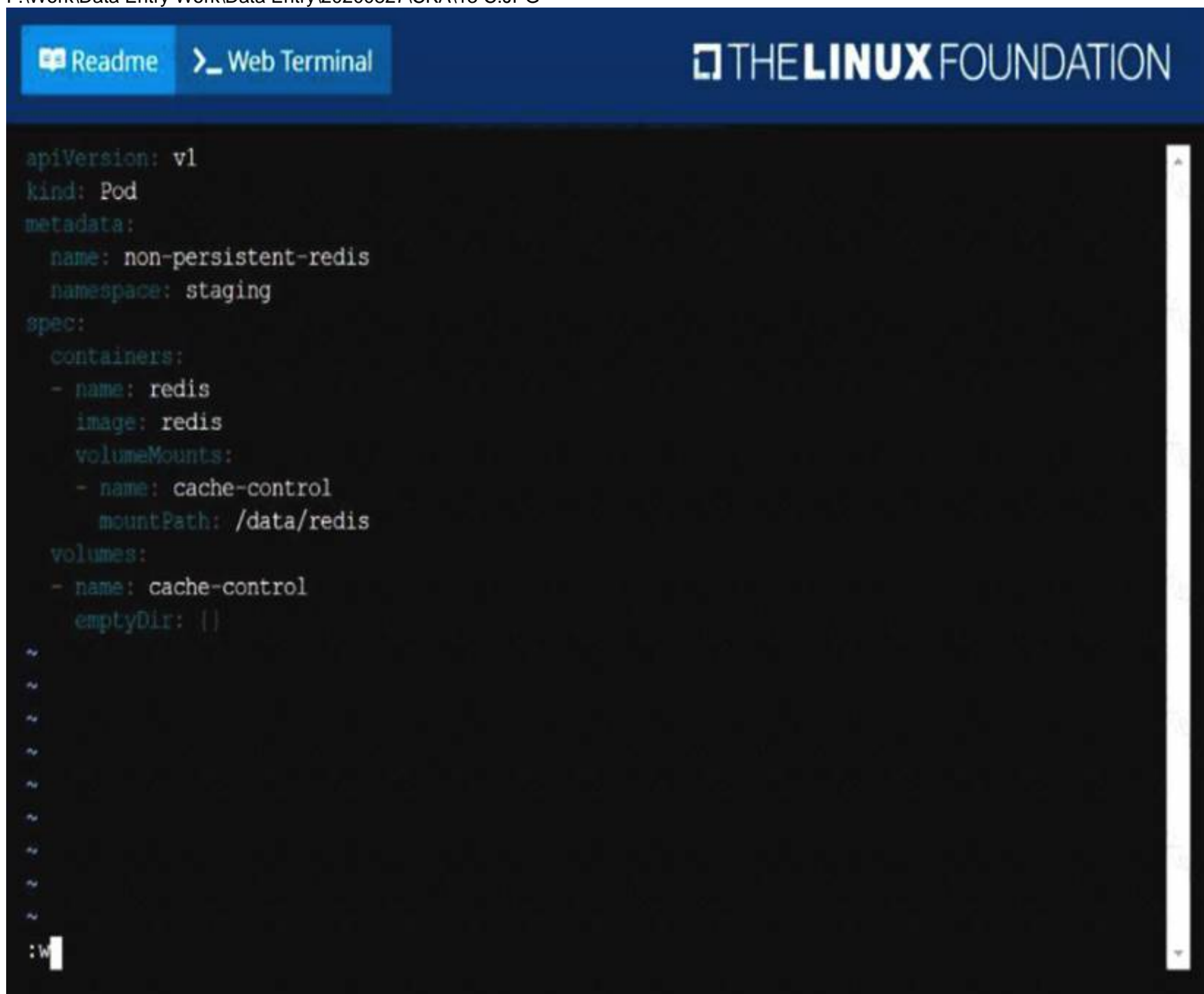
Answer: A

Explanation:

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



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 14

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
kubectl describe po nginx | grep value1

```

#### NEW QUESTION 18

For this item, you will have to ssh to the nodes ik8s-master-0 and ik8s-node-0 and complete all tasks on these nodes. Ensure that you return to the base node (hostname: node-1) when you have completed this item.

#### Context

As an administrator of a small development team, you have been asked to set up a Kubernetes cluster to test the viability of a new application.

#### Task

You must use kubeadm to perform this task. Any kubeadm invocations will require the use of the --ignore-preflight-errors=all option.

- > Configure the node ik8s-master-0 as a master node. .
- > Join the node ik8s-node-0 to the cluster.

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

#### solution

You must use the kubeadm configuration file located at /etc/kubeadm.conf when initializing your cluster.

You may use any CNI plugin to complete this task, but if you don't have your favourite CNI plugin's manifest URL at hand, Calico is one popular option:

<https://docs.projectcalico.org/v3.14/manifests/calico.yaml>

Docker is already installed on both nodes and apt has been configured so that you can install the required tools.

**NEW QUESTION 23**

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:**

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml > nginx-prodpod.yaml Now, edit nginx-prod-pod.yaml file and remove entries like "creationTimestamp: null" "dnsPolicy: ClusterFirst"

vim nginx-prod-pod.yaml

apiVersion: v1 kind: Pod metadata: labels:

env: prod

name: nginx-prod spec:

containers:

- image: nginx name: nginx-prod

restartPolicy: Always

# kubectl create -f nginx-prod-pod.yaml

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=dev nginx-dev --dry-run -o yaml > nginx-dev-pod.yaml apiVersion: v1

kind: Pod metadata: labels: env: dev

name: nginx-dev spec:

containers:

- image: nginx name: nginx-dev

restartPolicy: Always

# kubectl create -f nginx-prod-dev.yaml

Verify :

kubectl get po --show-labels kubectl get po -l env=prod kubectl get po -l env=dev

**NEW QUESTION 28**

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

**NEW QUESTION 32**

Scale the deployment webserver to 6 pods.

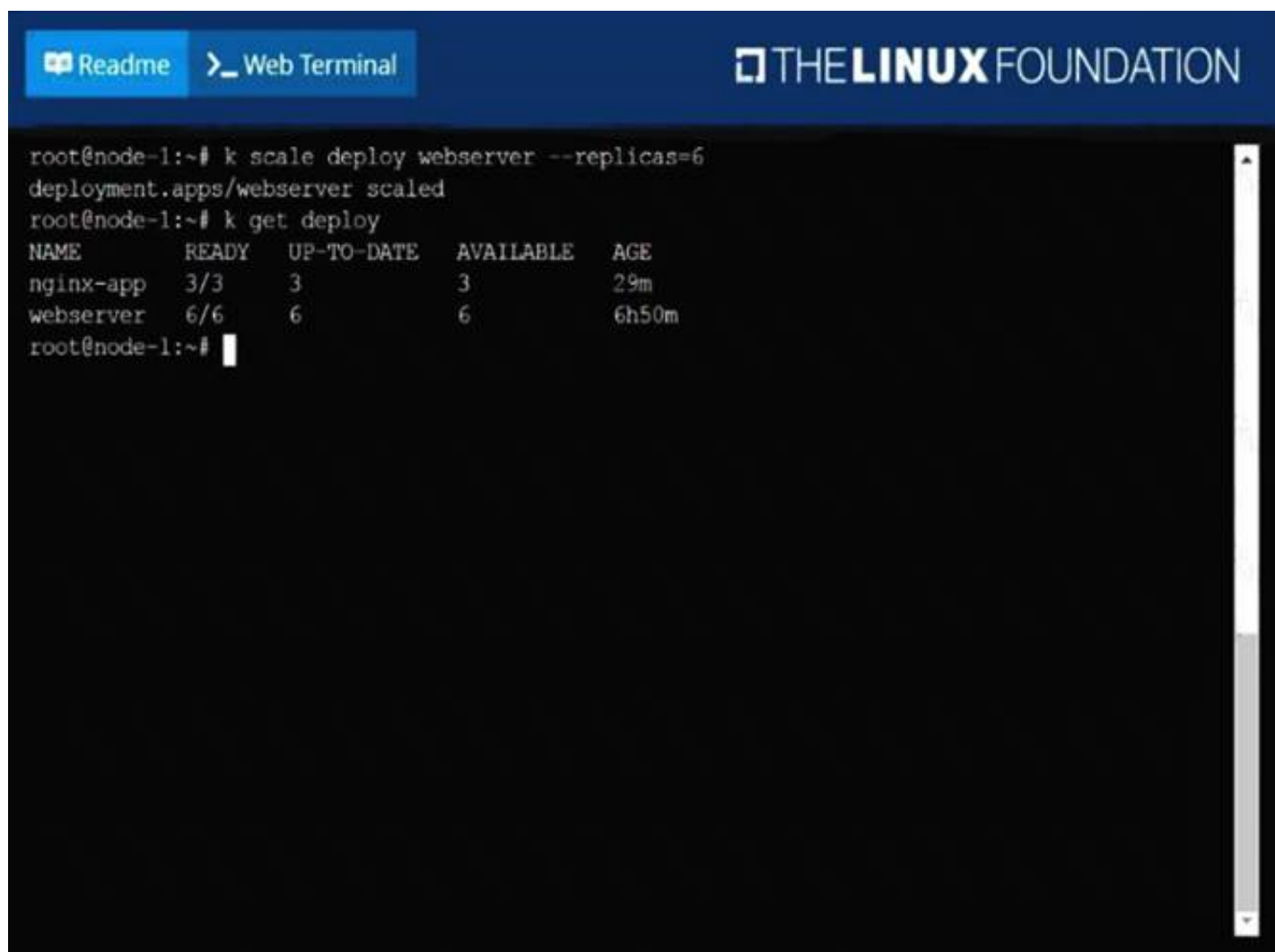
- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\14 B.JPG



The screenshot shows a terminal window titled 'Web Terminal' with the 'THE LINUX FOUNDATION' logo in the top right. The terminal output shows a user at the root of a node-1 scaling a deployment named 'webserver' to 6 replicas. The output shows the deployment was scaled successfully. Then, the user runs 'k get deploy' and the output shows a table with columns: NAME, READY, UP-TO-DATE, AVAILABLE, and AGE. The table has two rows: 'nginx-app' with values 3/3, 3, 3, and 29m; and 'webserver' with values 6/6, 6, 6, and 6h50m.

```
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:~#
```

**NEW QUESTION 36**

Create a deployment as follows:

- Name: nginx-random
- Exposed via a service nginx-random
- Ensure that the service & pod are accessible via their respective DNS records
- The container(s) within any pod(s) running as a part of this deployment should use the nginx Image

Next, use the utility nslookup to look up the DNS records of the service & pod and write the output to /opt/KUNW00601/service.dns and /opt/KUNW00601/pod.dns respectively.

- A. Mastered
- B. Not Mastered

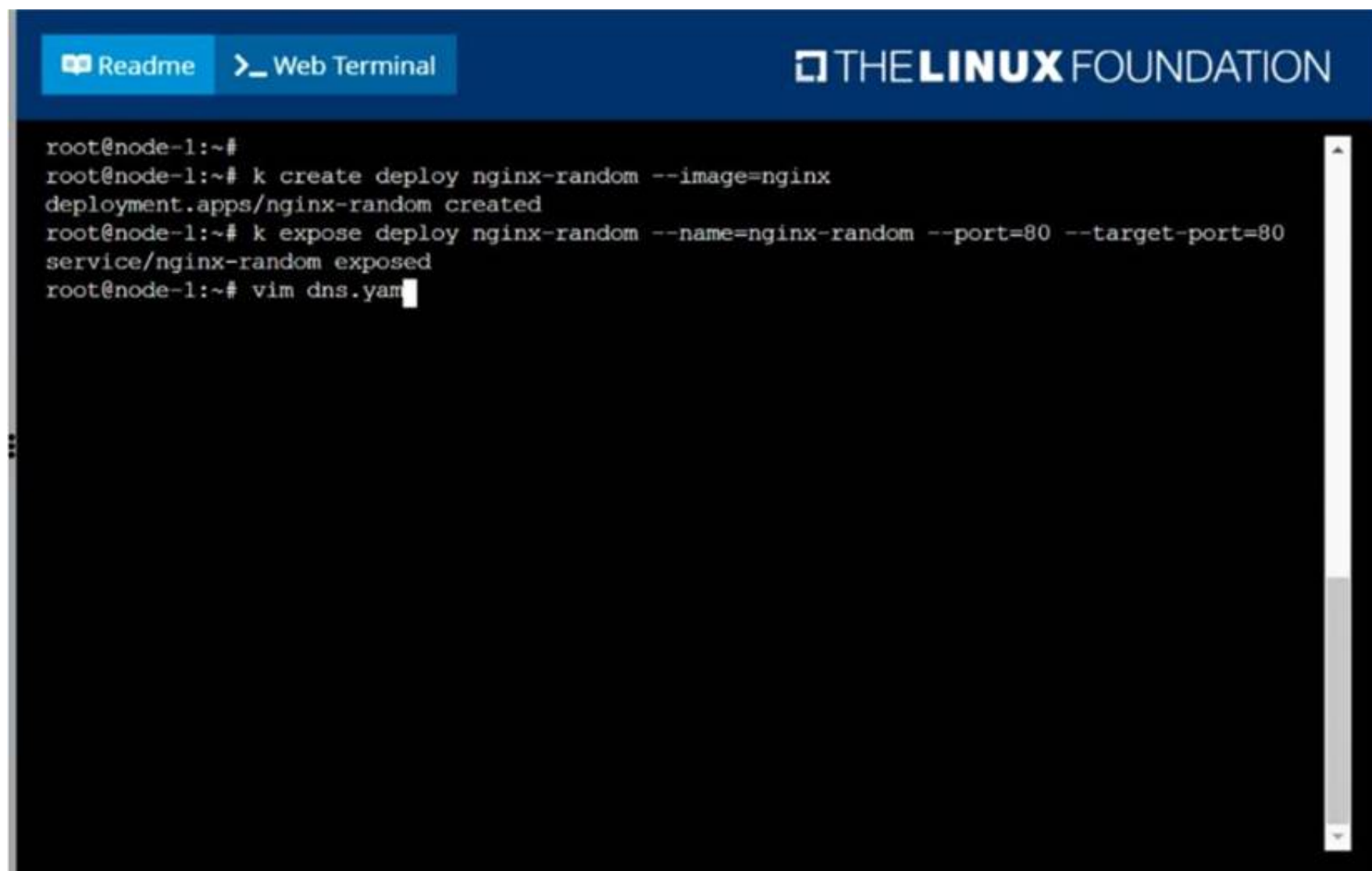
**Answer:** A

**Explanation:**

Solution:

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 C.JPG

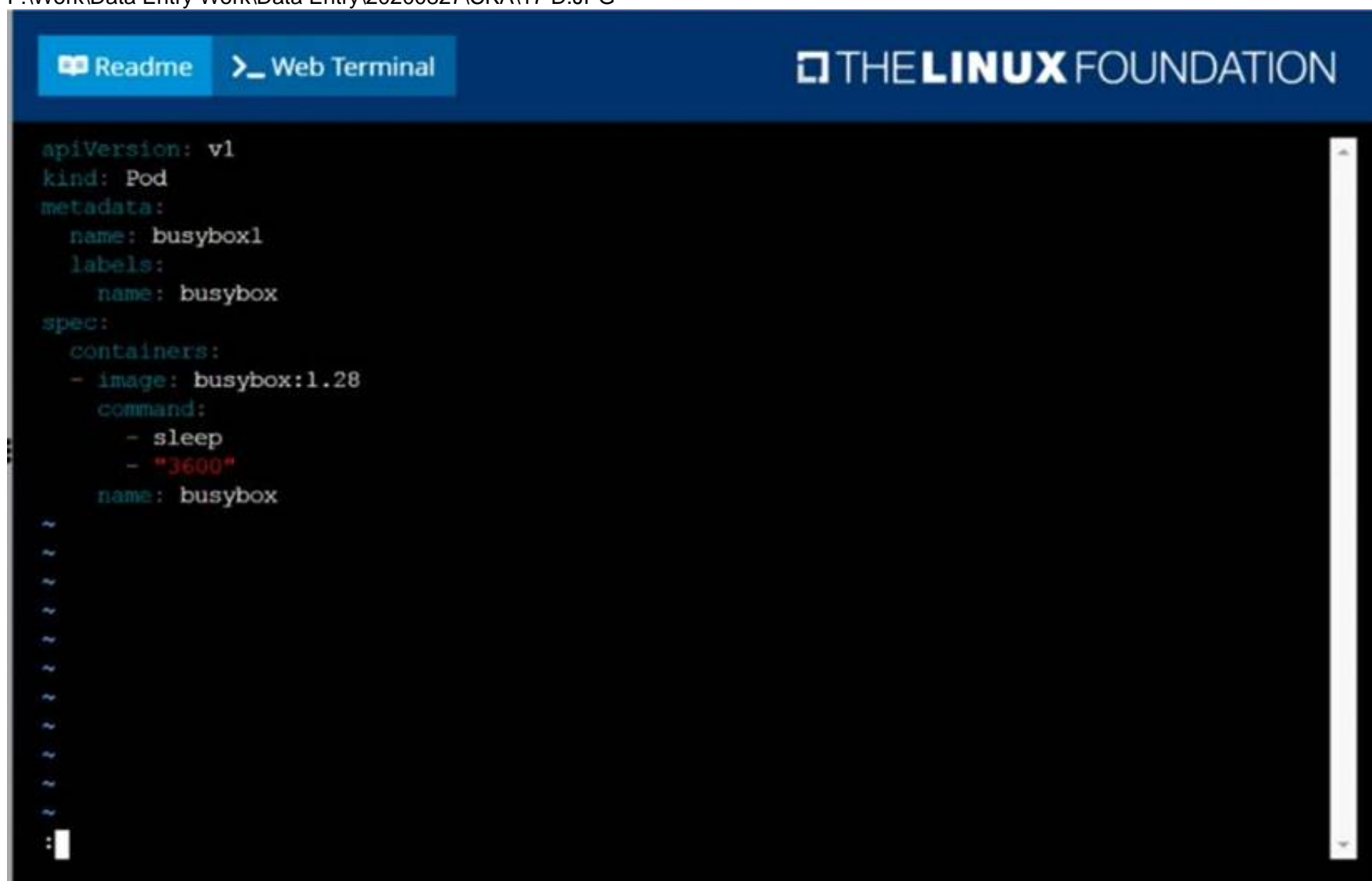




The screenshot shows a web terminal interface with a dark blue header. On the left, there are two buttons: 'Readme' and 'Web Terminal'. On the right, the 'THE LINUX FOUNDATION' logo is displayed. The terminal window shows a series of commands and their outputs:

```
root@node-1:~#
root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 D.JPG



The screenshot shows a web terminal interface with a dark blue header. On the left, there are two buttons: 'Readme' and 'Web Terminal'. On the right, the 'THE LINUX FOUNDATION' logo is displayed. The terminal window shows a Kubernetes manifest file:

```
apiVersion: v1
kind: Pod
metadata:
 name: busybox1
 labels:
 name: busybox
spec:
 containers:
 - image: busybox:1.28
 command:
 - sleep
 - "3600"
 name: busybox
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 E.JPG

```

root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yaml
root@node-1:~# k create -f dns.yaml
pod/busybox1 created
root@node-1:~# k get po -o wide | grep nginx-random
nginx-random-6d5766bbdc-ptzv2 1/1 Running 0 103s 10.244.2.16 k8s-node-1
 <none> <none>
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name: nginx-random
Address 1: 10.111.37.132 nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random > /opt/KUNW00601/service.dns
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name: 10-244-2-16.default.pod
Address 1: 10.244.2.16 10-244-2-16.nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod > /opt/KUNW00601/pod.dns

```

#### NEW QUESTION 37

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 41

Score: 4%



Task

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:

kubectl run kucc8 --image=nginx --dry-run -o yaml > kucc8.yaml

# vi kucc8.yaml apiVersion: v1 kind: Pod metadata:

creationTimestamp: null name: kucc8

spec: containers:

- image: nginx name: nginx

- image: redis name: redis

- image: memcached

name: memcached

- image: consul name: consul

#

kubectl create -f kucc8.yaml

#12.07

**NEW QUESTION 43**

Ensure a single instance of pod nginx is running on each node of the Kubernetes cluster where nginx also represents the Image name which has to be used. Do not override any taints currently in place.

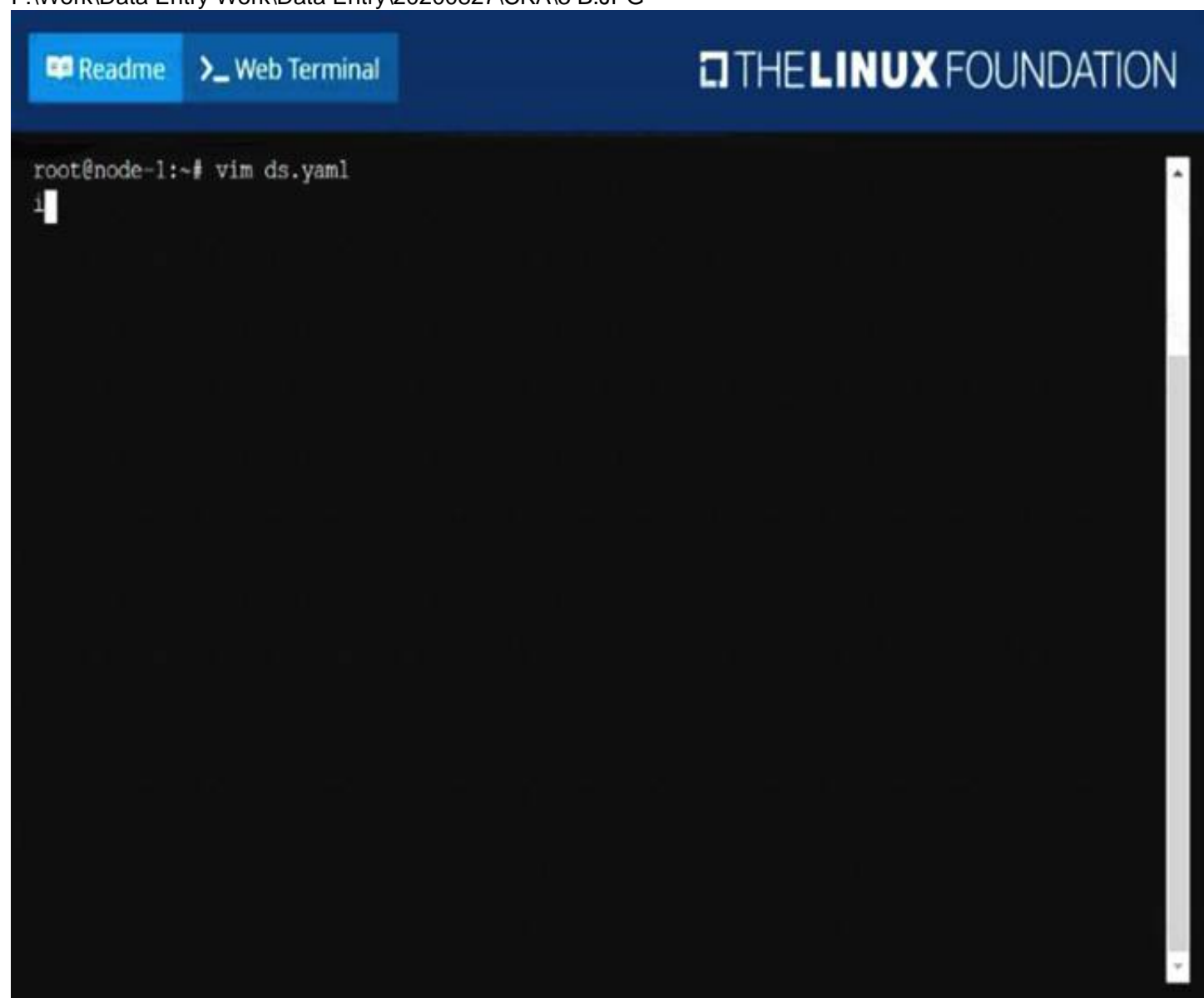
Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

- A. Mastered
- B. Not Mastered

**Answer:** A**Explanation:**

solution

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



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

ReadmeWeb 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 --17,19All
```

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

ReadmeWeb Terminal

THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
 name: ds-kusc00201
spec:
 selector:
 matchLabels:
 name: fluentd-elasticsearch
 template:
 metadata:
 labels:
 name: fluentd-elasticsearch
 spec:
 containers:
 - name: nginx
 image: nginx
~
~
~
~
~
~
~
~
~
:wc
```

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

ReadmeWeb 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:~#
```

NEW QUESTION 47

List “nginx-dev” and “nginx-prod” pod and delete those pods

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubect1 get pods -o wide  
kubectl delete po “nginx-dev”kubectl delete po “nginx-prod”

NEW QUESTION 50

Create a nginx pod with label env=test in engineering namespace

- A. Mastered
- B. Not Mastered

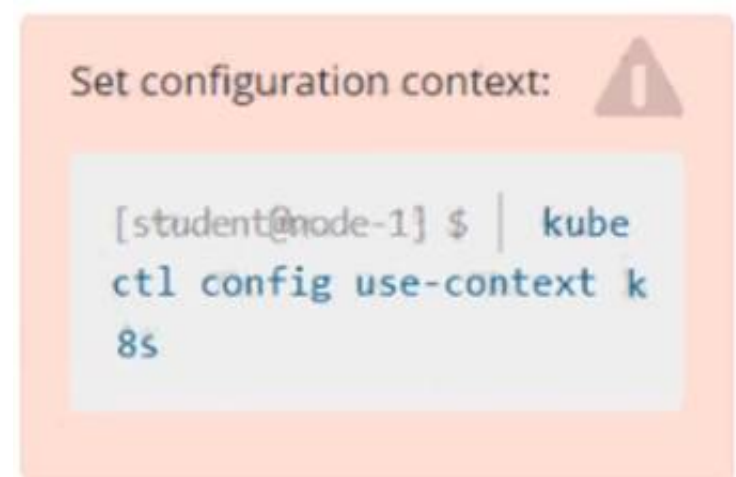
Answer: A

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 -n engineering -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 52

Score: 4%





Context

You have been asked to create a new ClusterRole for a deployment pipeline and bind it to a specific ServiceAccount scoped to a specific namespace.

Task

Create a new ClusterRole named deployment-clusterrole, which only allows to create the following resource types:

- Deployment
- StatefulSet
- DaemonSet

Create a new ServiceAccount named cicd-token in the existing namespace app-team1.

Bind the new ClusterRole deployment-clusterrole to the new ServiceAccount cicd-token , limited to the namespace app-team1.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

Task should be complete on node k8s -1 master, 2 worker for this connect use command

```
[student@node-1] > ssh k8s
```

```
kubectl create clusterrole deployment-clusterrole --verb=create
```

```
--resource=deployments,statefulsets,daemonsets
```

```
kubectl create serviceaccount cicd-token --namespace=app-team1
```

```
kubectl create rolebinding deployment-clusterrole --clusterrole=deployment-clusterrole
```

```
--serviceaccount=default:cicd-token --namespace=app-team1
```

## NEW QUESTION 54

.....



## 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>