

Linux-Foundation

Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program



NEW QUESTION 1

Score: 4%



Task
Create a persistent volume with name app-data , of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-data .

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
#vi pv.yaml apiVersion: v1
kind: PersistentVolume metadata:
name: app-config spec:
capacity: storage: 1Gi accessModes:
- ReadOnlyMany hostPath:
path: /srv/app-config

kubectl create -f pv.yaml

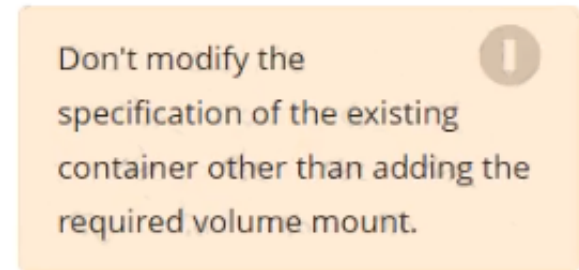
NEW QUESTION 2

Score:7%



Context
An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e. g. kubectl logs). Adding a streaming sidecar container is a good and common way to accomplish this requirement.

Task
Add a sidecar container named sidecar, using the busybox Image, to the existing Pod big-corp-app. The new sidecar container has to run the following command:
/bin/sh -c tail -n+1 -f /var/log/big-corp-app.log
Use a Volume, mounted at /var/log, to make the log file big-corp-app.log available to the sidecar container.



A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
#

```
kubectl get pod big-corp-app -o yaml
#
apiVersion: v1 kind: Pod metadata:
name: big-corp-app spec:
containers:
- name: big-corp-app image: busybox
args:
- /bin/sh
- -c
- > i=0;
while true; do
echo "$(date) INFO $i" >> /var/log/big-corp-app.log; i=$((i+1));
sleep 1; done
volumeMounts:
- name: logs mountPath: /var/log
- name: count-log-1 image: busybox
args: [/bin/sh, -c, 'tail -n+1 -f /var/log/big-corp-app.log'] volumeMounts:
- name: logs mountPath: /var/log volumes:
- name: logs emptyDir: {
}
#
kubectl logs big-corp-app -c count-log-1
```

NEW QUESTION 3

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 4

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
```

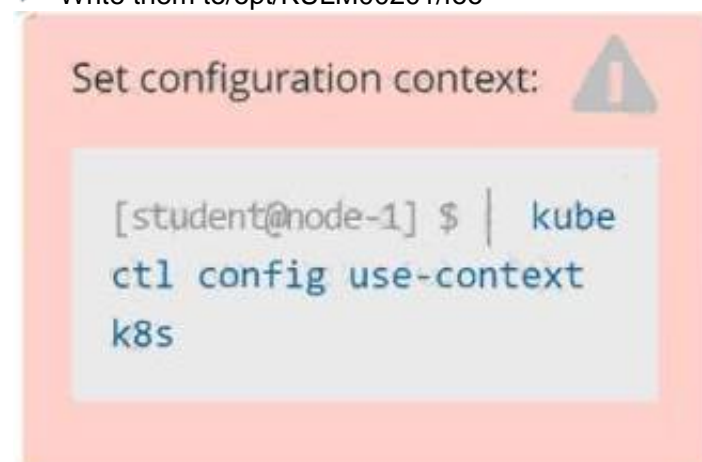
```
- image: consul
```

```
name: consul-container restartPolicy: Always
```

NEW QUESTION 5

Monitor the logs of pod foo and:

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



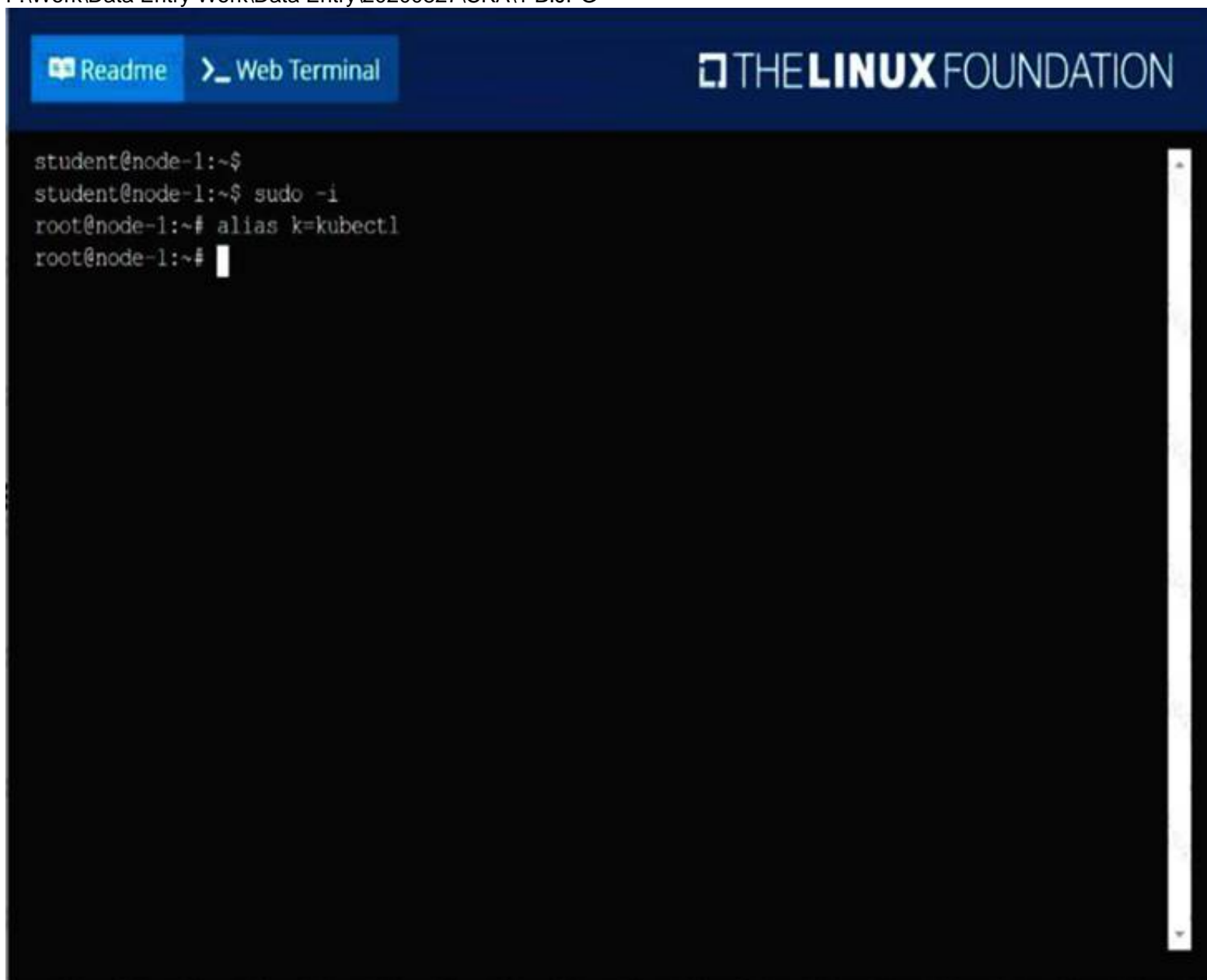
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

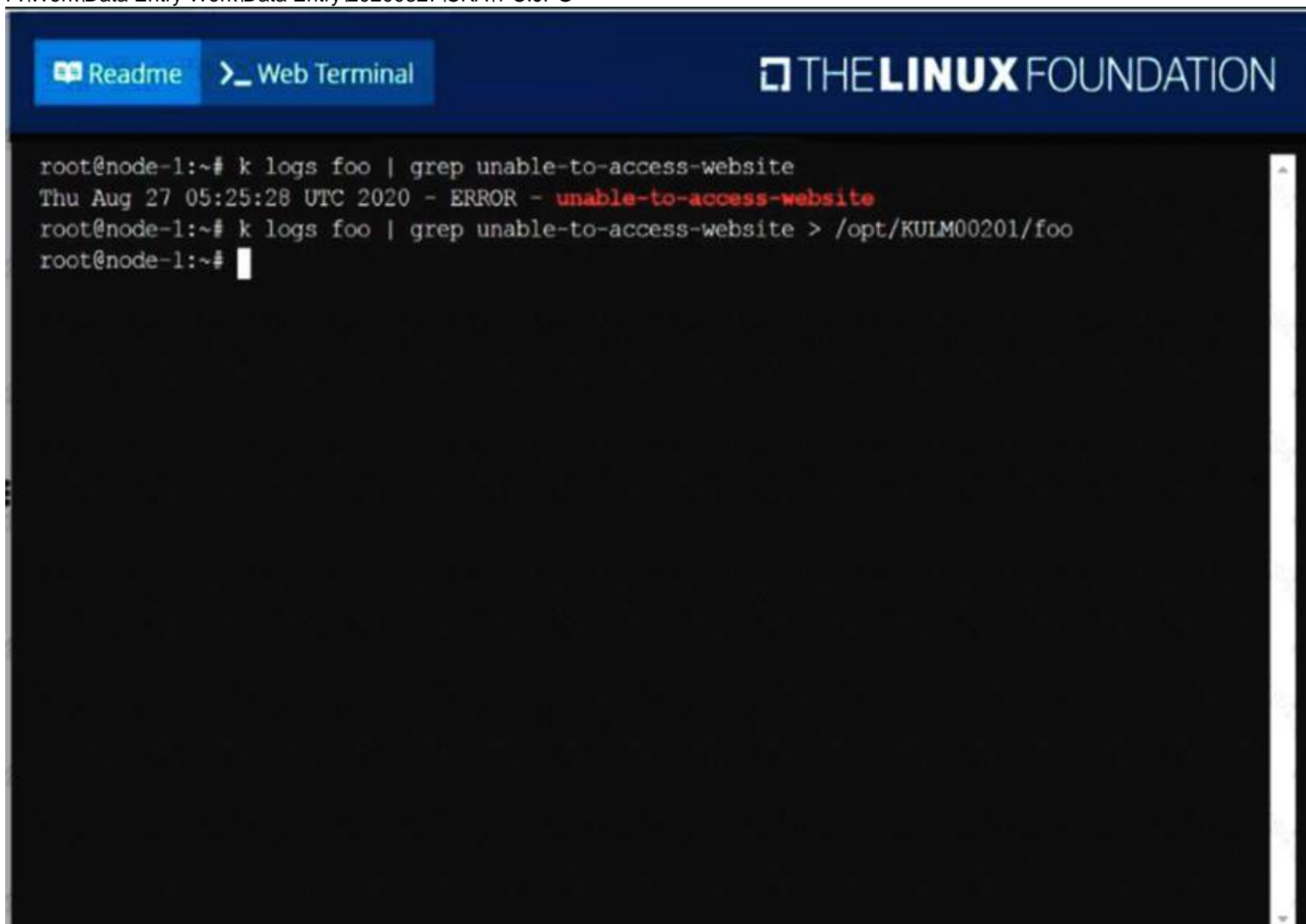
solution

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=kubectl
root@node-1:~#
```

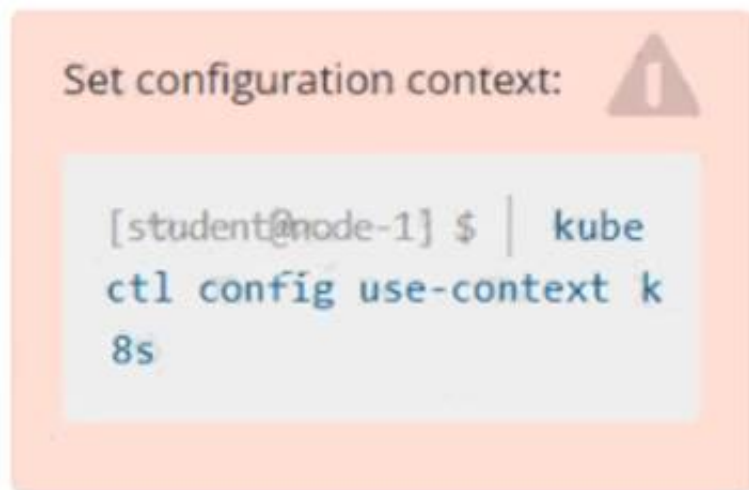
F:\Work\Data Entry Work\Data Entry\20200827\CKA\1 C.JPG



```
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/KUIM00201/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.

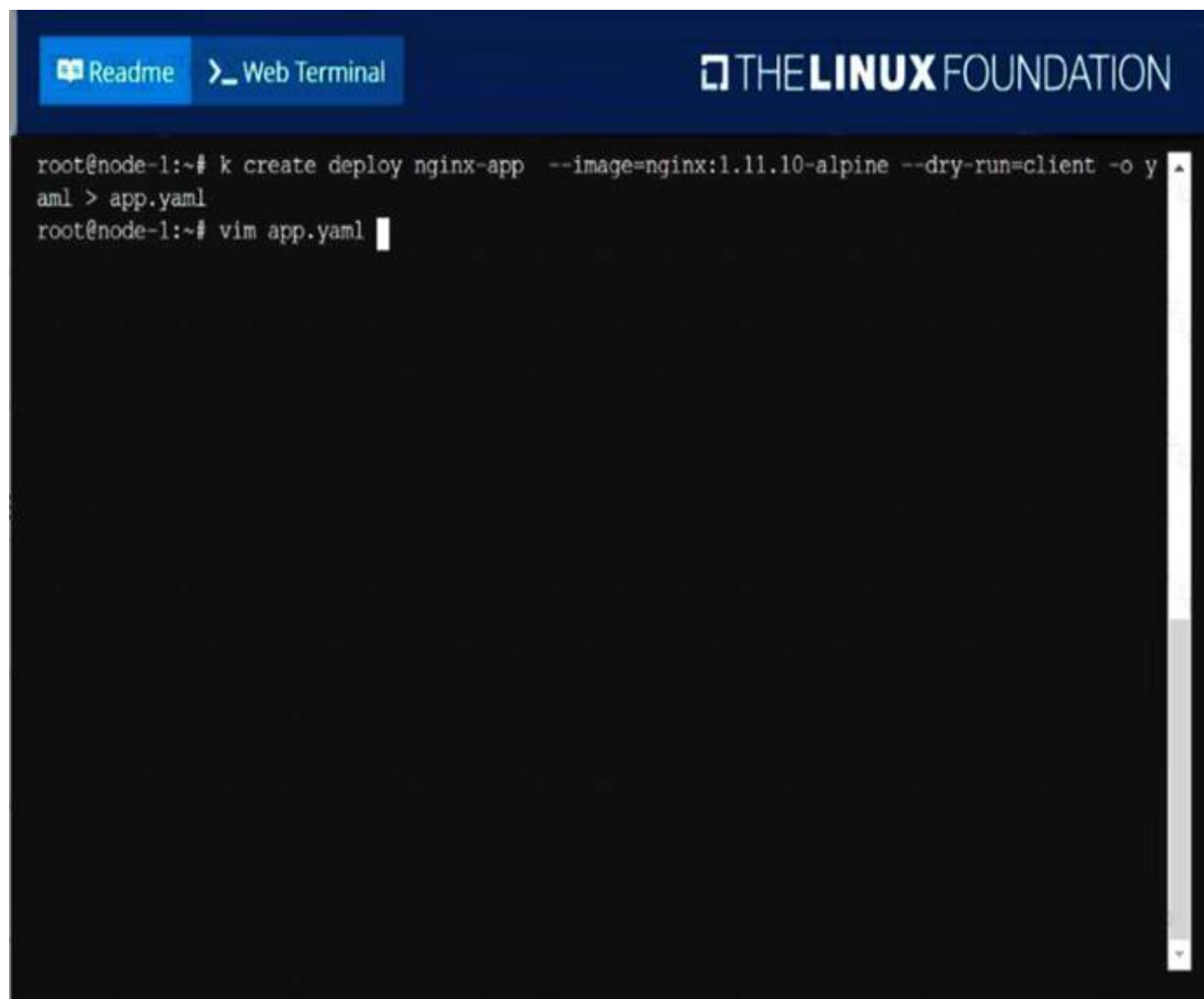
- A. Mastered
- B. Not Mastered

Answer: A

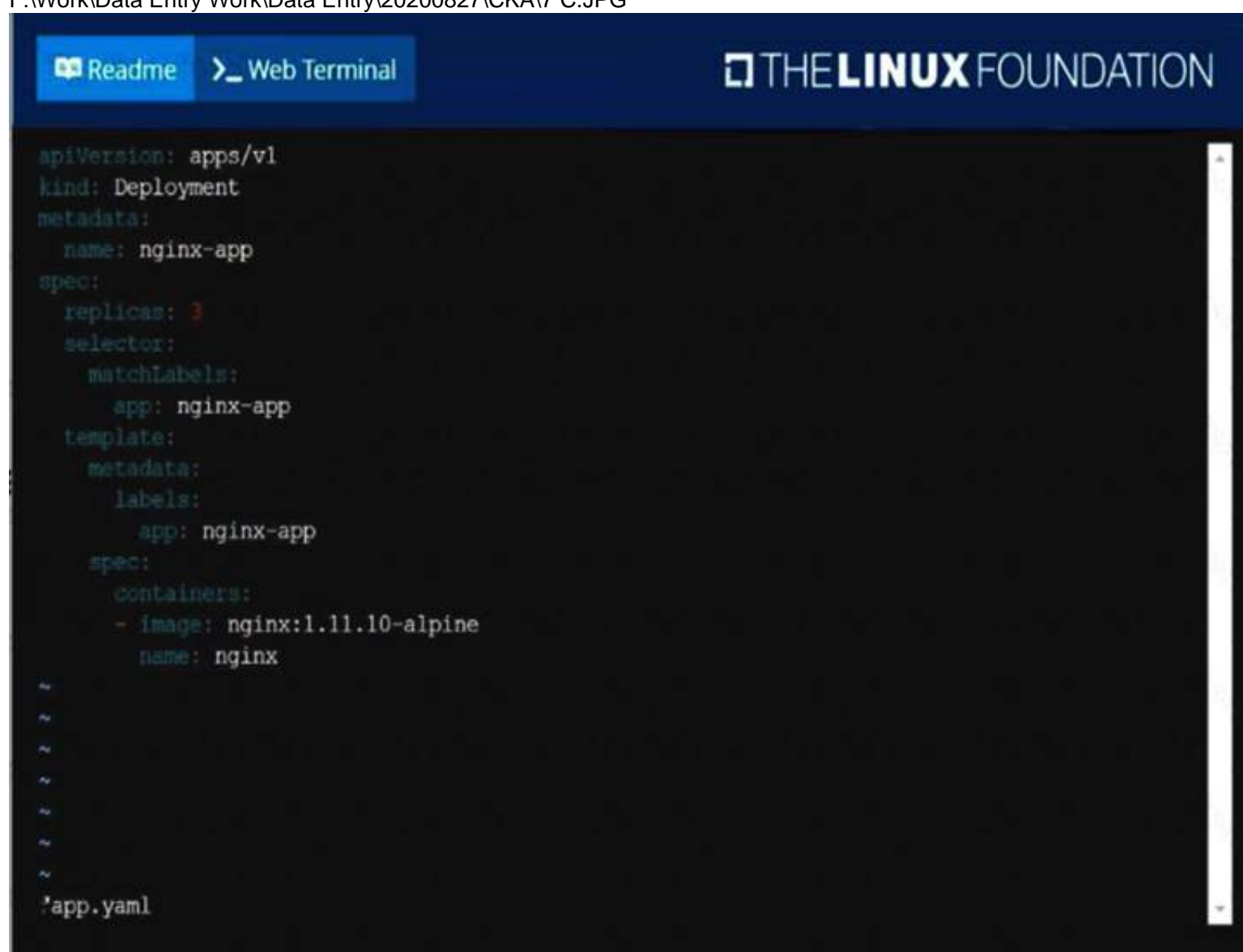
Explanation:

solution

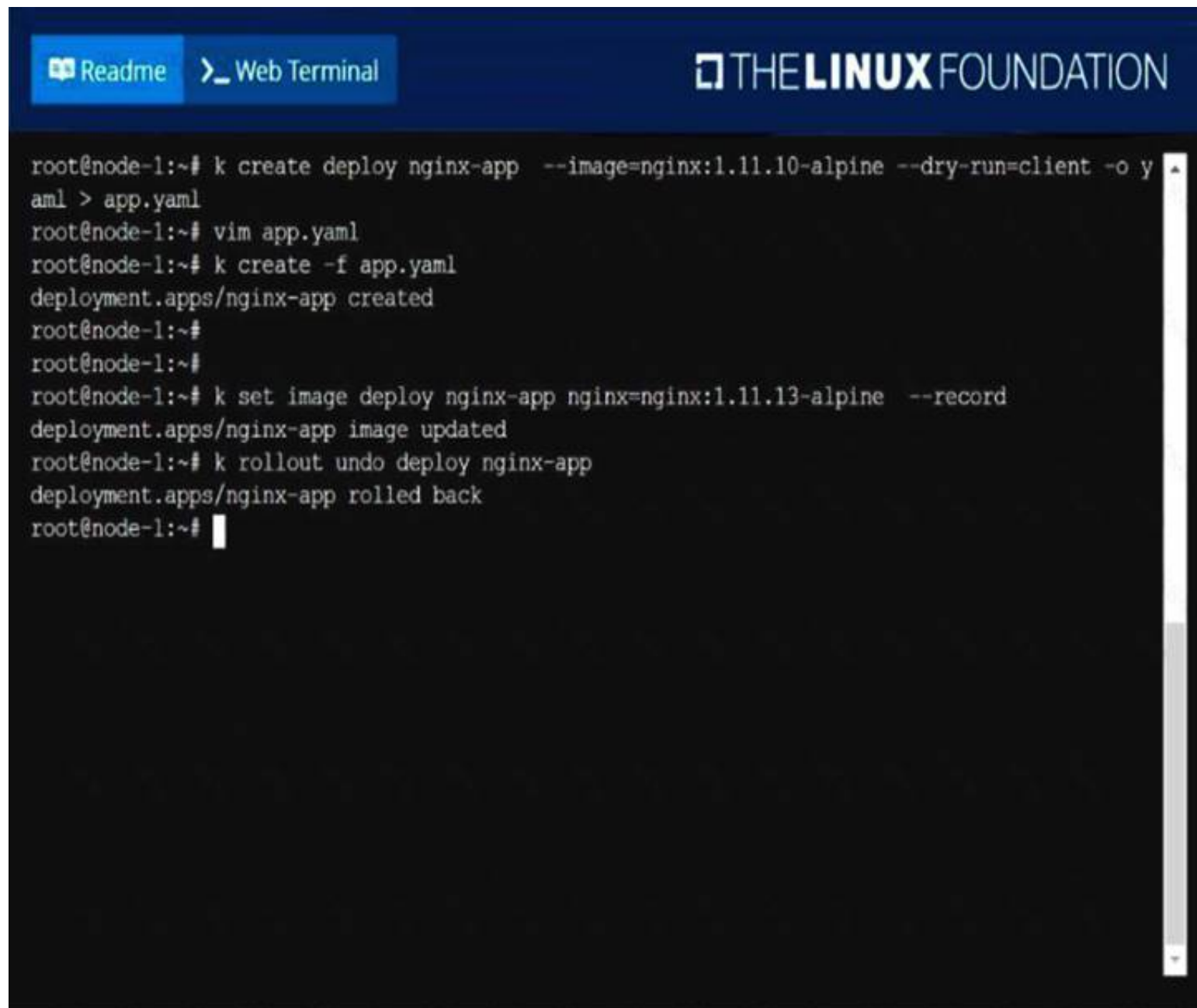
F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 B.JPG



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



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 8

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.

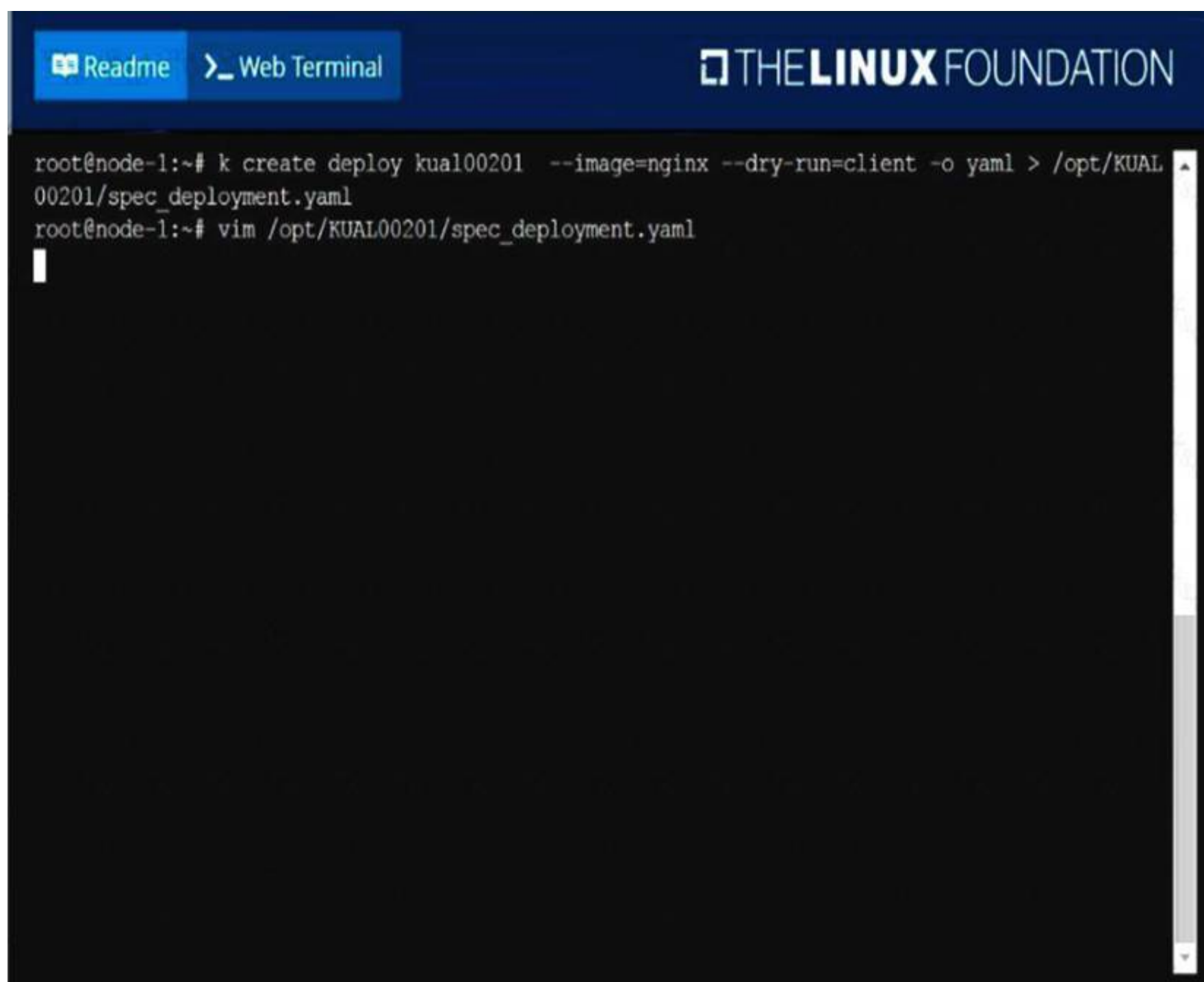
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

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

A screenshot of a web terminal interface. At the top, there is a dark blue header bar. 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 main area is a black terminal window with white text. The text shows a user at a root prompt on a node-1 machine, creating a deployment spec for nginx and then opening it in vim. A vertical scrollbar is visible on the right side of the terminal window.

```
root@node-1:~# k create deploy kual00201 --image=nginx --dry-run=client -o yaml > /opt/KUAL
00201/spec_deployment.yaml
root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml
```

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



```
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

Create a pod as follows:

- > Name: mongo
- > Using Image: mongo
- > In a new Kubernetes namespace named: my-website

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

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

Readme

Web Terminal

THE **LINUX** FOUNDATION

```
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 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 10

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.

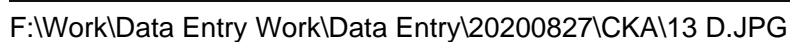
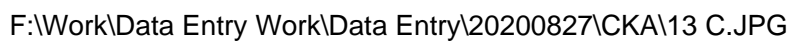
- A. Mastered
- B. Not Mastered

Answer: A


Explanation:

solution

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



Readme
Web Terminal



```

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 15

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 17

Create a pod that echo "hello world" and then exists. Have the pod deleted automatically when it's completed

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubectl run busybox --image=busybox -it --rm --restart=Never -
 /bin/sh -c 'echo hello world'
 kubectl get po # You shouldn't see pod with the name "busybox"

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.

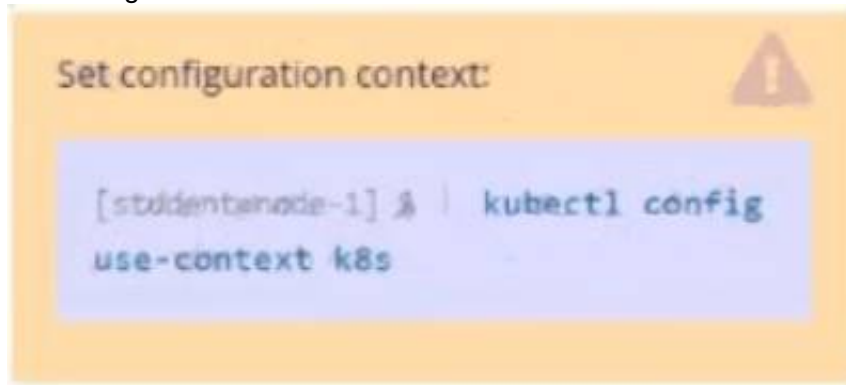
You may use any CNI plugin to complete this task, but if you don't have your favourite CNI plugin's manifest URL at hand, Calico is one popular option:

<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 22

Task Weight: 4%



Task

Scale the deployment webserver to 3 pods.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
student@node-1:~$ kubectl scale deploy webserver --replicas=3
deployment.apps/webserver scaled
student@node-1:~$ kubectl scale deploy webserver --replicas=3
```

NEW QUESTION 25

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

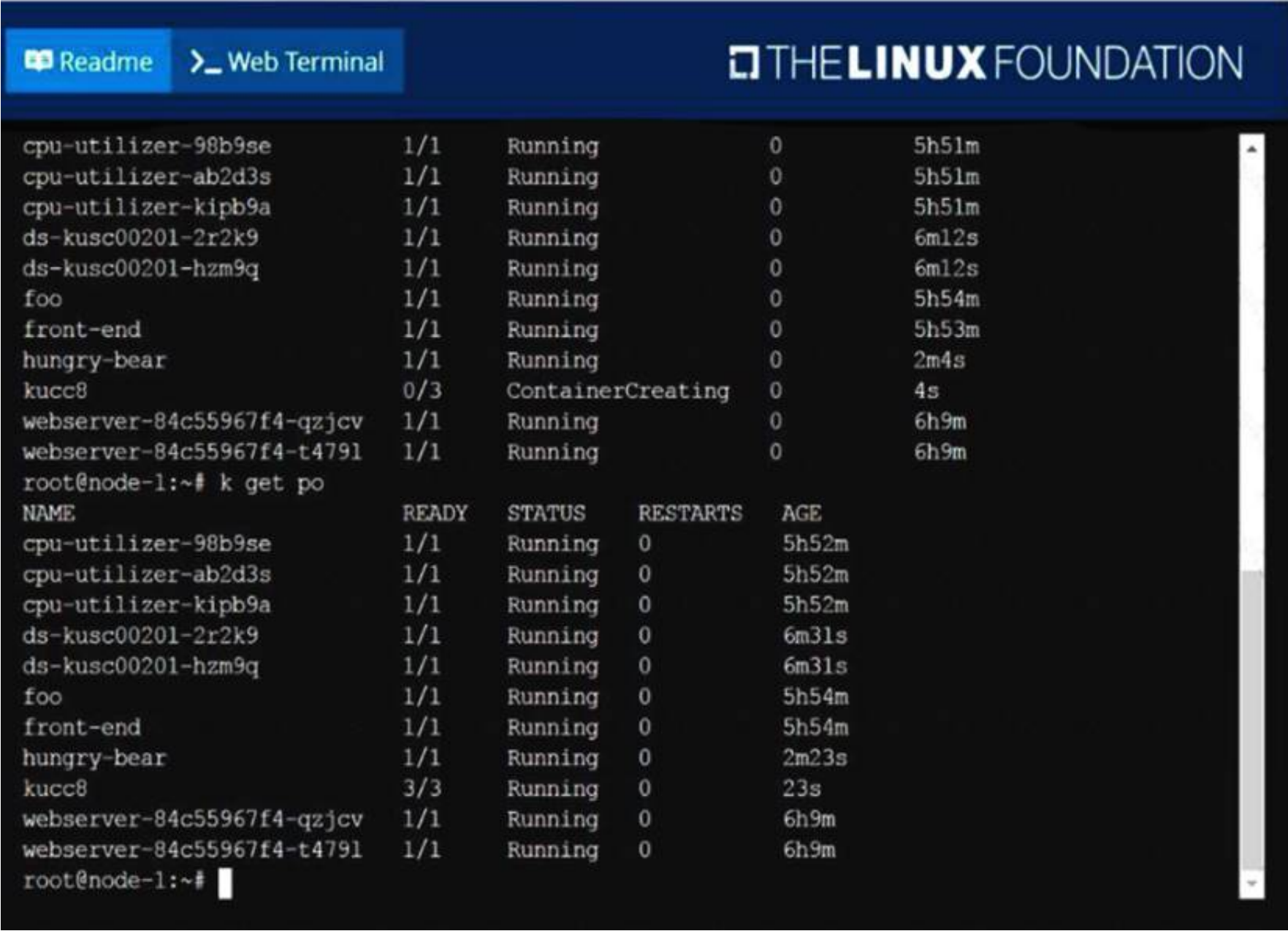
Answer: A

Explanation:

solution

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





NEW QUESTION 27

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 28

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 33

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 Persistent Volume
kind: PersistentVolumeapiVersion: v1metadata: name:app-dataspec: capacity: # defines the capacity of PV we are creating storage: 2Gi #the amount of storage

Challenge

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

* 3. View the persistent volume.

Challenge

* 3. View the pvc Image for post

visit - <https://www.surepassexam.com>

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: v1kind: Podmetadata: creationTimestamp: null name: app-dataspec: volumes: - name:congigpvc persistenVolumeClaim: claimName: app-data
containers: - image: nginx name: app volumeMounts: - mountPath: "/srv/app-data " name: configpvc

NEW QUESTION 34

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



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

```

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-t479l	1/1	Running	0	6h43m

```

root@node-1:~# █

```

visit - <https://www.surepassexam.com>

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 41

.....

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](#)