

Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

<https://www.2passeasy.com/dumps/CKA/>



NEW QUESTION 1

CORRECT TEXT

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 2

CORRECT TEXT

Create a pod with image nginx called nginx and allow traffic on port 80

- A. Mastered
- B. Not Mastered

Answer: A

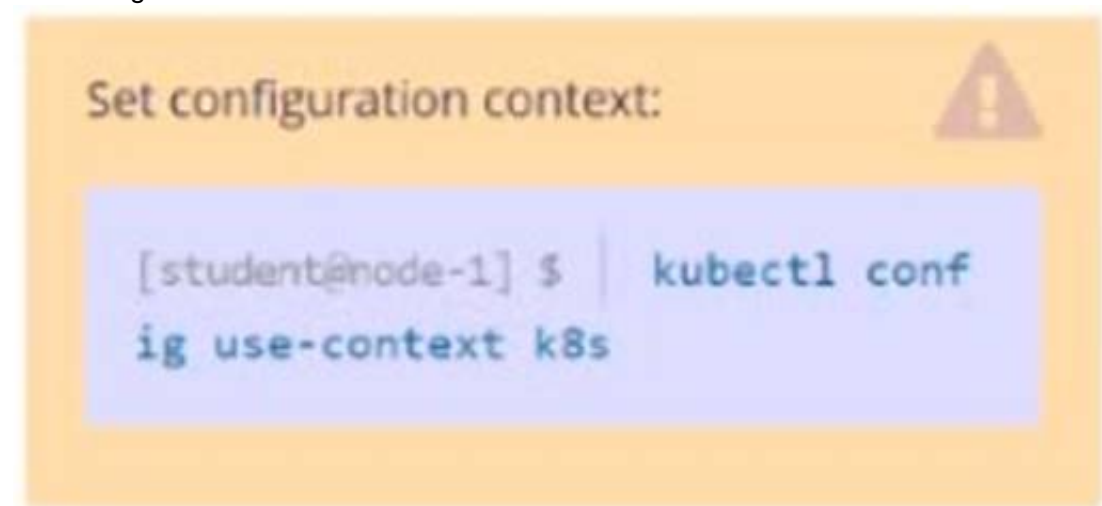
Explanation:

kubectl run nginx --image=nginx --restart=Never --port=80

NEW QUESTION 3

CORRECT TEXT

Task Weight: 4%



Task

Schedule a Pod as follows:

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

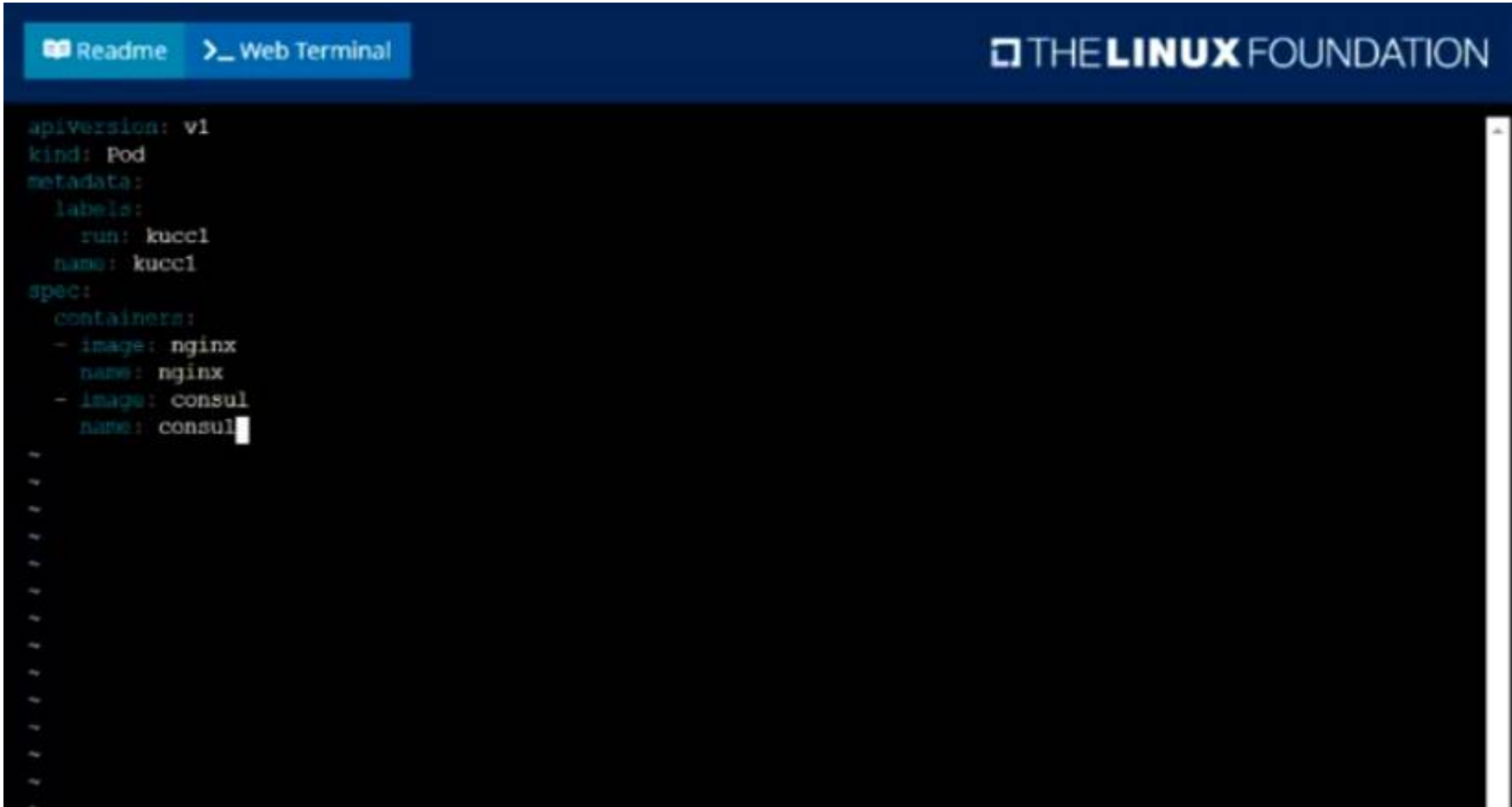
- A. Mastered
- B. Not Mastered

Answer: A

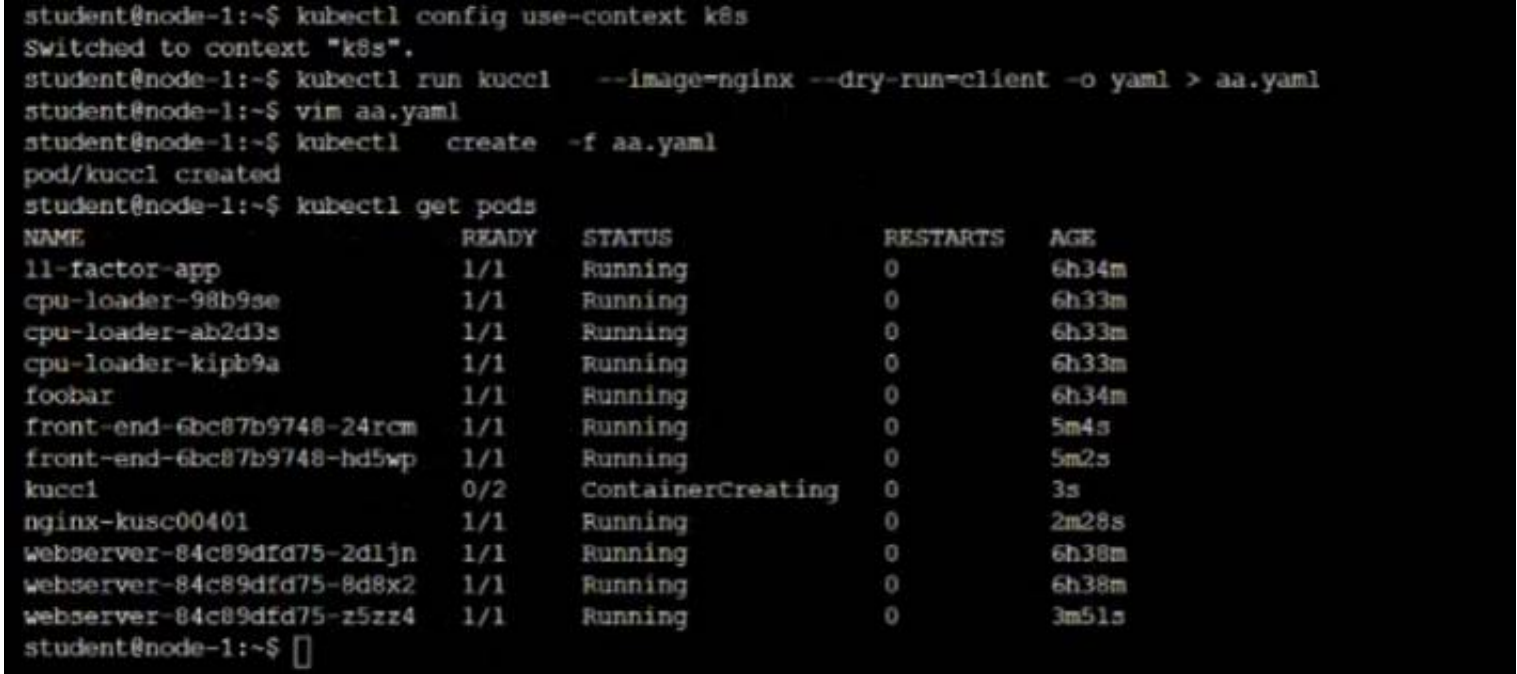
Explanation:

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

NEW QUESTION 4

CORRECT TEXT

Get IP address of the pod – “nginx-dev”

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Kubect1 get po -o wide
Using JsonPath
kubect1 get pods -o=jsonpath='{range items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'

NEW QUESTION 5

CORRECT TEXT

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 6

CORRECT TEXT

Monitor the logs of pod foo and:

? Extract log lines corresponding to error

unable-to-access-website

? Write them to /opt/KULM00201/foo

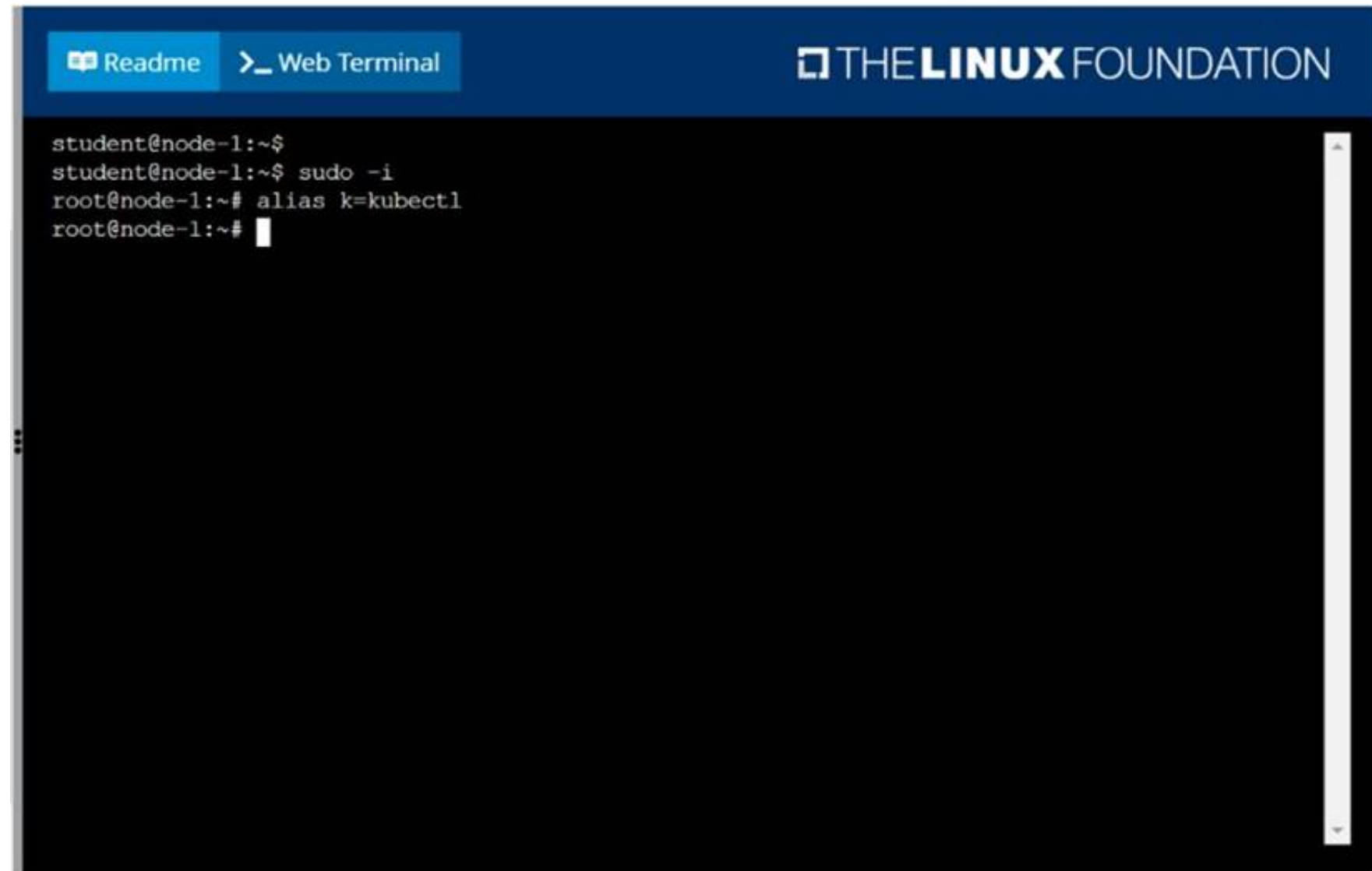
A. Mastered

B. Not Mastered

Answer: A

Explanation:

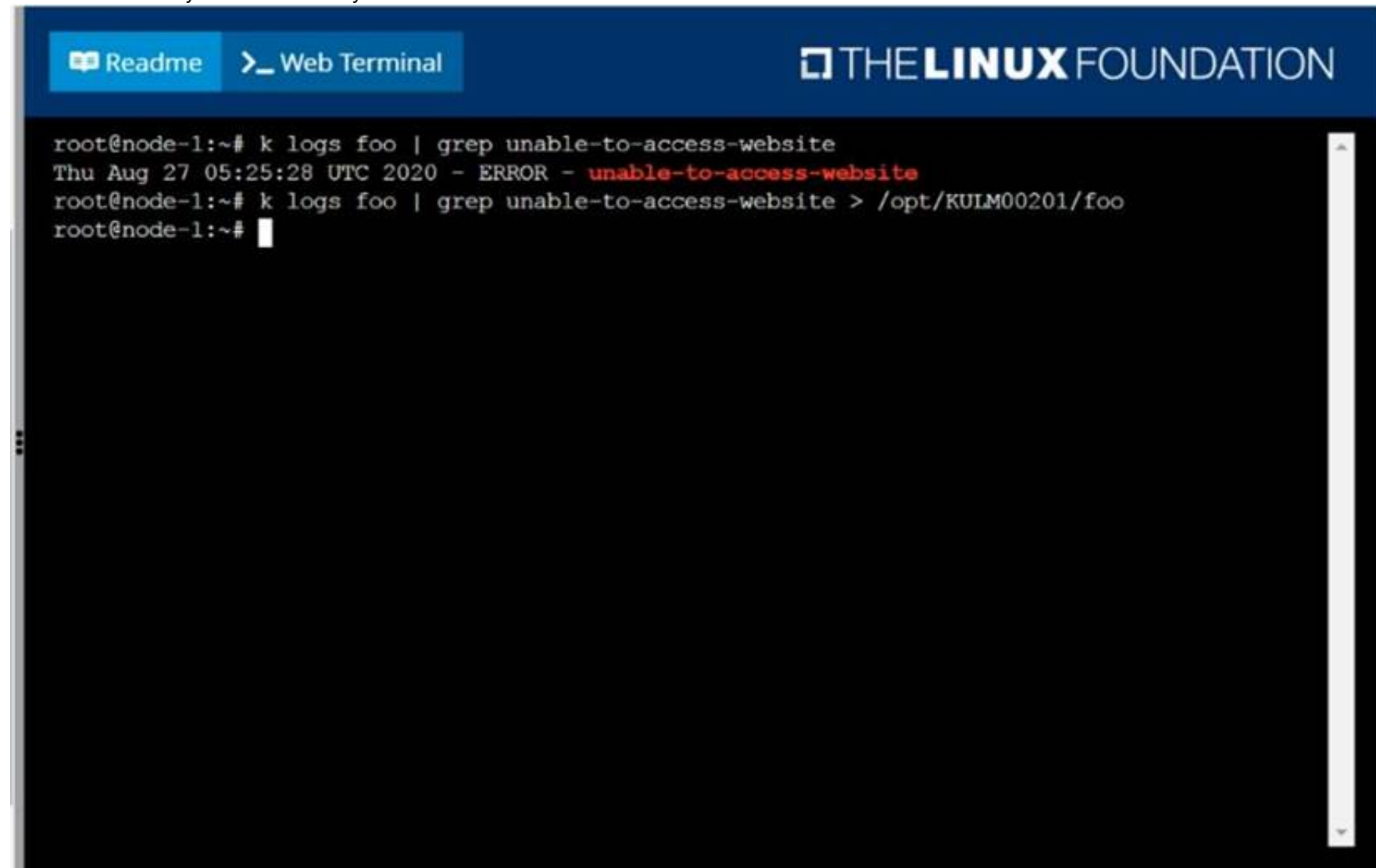
solution



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

```
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 B.JPG



The screenshot shows the same web terminal interface as the previous one. The terminal window shows the following commands and output:

```
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:~#
```

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

NEW QUESTION 7

CORRECT TEXT
Score: 7%



Task
Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace echo. Ensure that the new NetworkPolicy allows Pods in namespace my-app to connect to port 9000 of Pods in namespace echo.
Further ensure that the new NetworkPolicy:

- does not allow access to Pods, which don't listen on port 9000
- does not allow access from Pods, which are not in namespace my-app

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
#network.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: internal
spec:
podSelector:
matchLabels: {
}
policyTypes:
- Ingress
ingress:
- from:
- podSelector: {
}
ports:
- protocol: TCP
port: 8080
#spec.podSelector namespace pod
kubectl create -f network.yaml

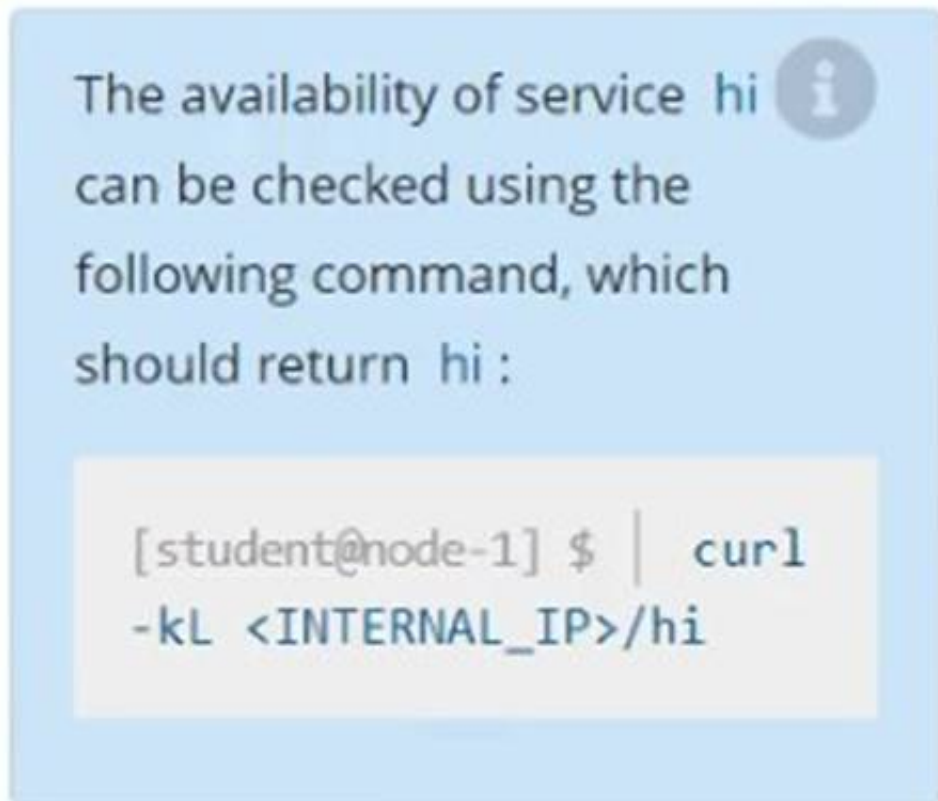
NEW QUESTION 8

CORRECT TEXT
Score: 7%



Task
Create a new nginx Ingress resource as follows:

- Name: ping
- Namespace: ing-internal
- Exposing service hi on path /hi using service port 5678



- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:
vi ingress.yaml

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
name: ping
namespace: ing-internal
spec:
rules:
- http:
paths:
- path: /hi
pathType: Prefix
backend:
service:
name: hi
port:
number: 5678

kubectl create -f ingress.yaml

NEW QUESTION 9

CORRECT TEXT

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}'{"\n"}

NEW QUESTION 10

CORRECT TEXT

Create a snapshot of the etcd instance running at <https://127.0.0.1:2379>, saving the snapshot to the file path /srv/data/etcd-snapshot.db.

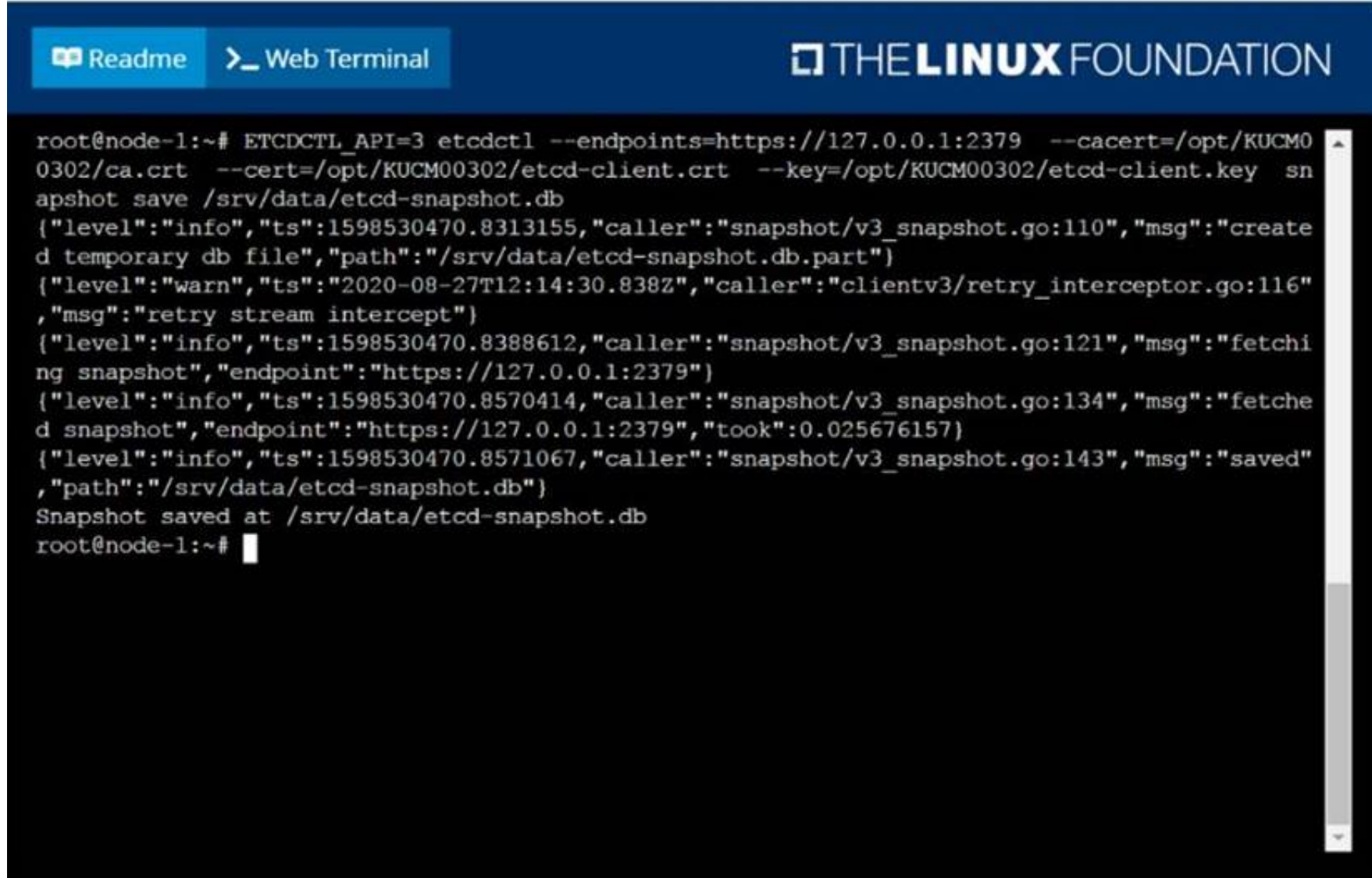
The following TLS certificates/key are supplied for connecting to the server with etcdctl:

- ? 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



```

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:~#
  
```

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

NEW QUESTION 10

CORRECT TEXT

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 15

CORRECT TEXT

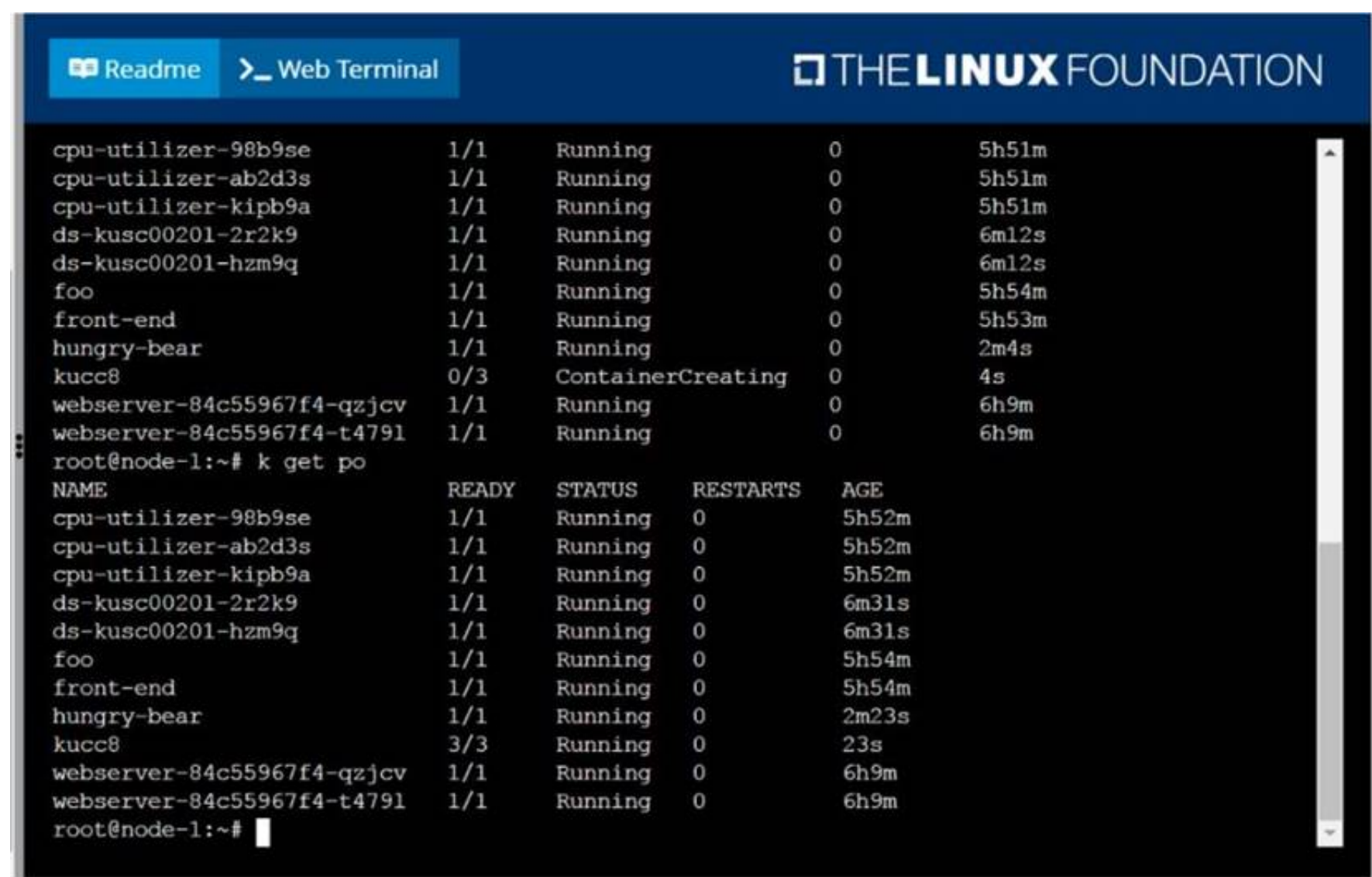
Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified):
 nginx + redis + memcached.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:
 solution





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

NEW QUESTION 19

CORRECT TEXT

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 20

CORRECT TEXT

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 25

CORRECT TEXT

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 26

CORRECT TEXT

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 29

CORRECT TEXT

Get list of all pods in all namespaces and write it to file "/opt/pods-list.yaml"

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubectl get po --all-namespaces > /opt/pods-list.yaml

Image for post

* 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: PersistentVolumeapiVersion: v1metadata: 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

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pvc
NAME      STATUS   VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS
pv        Bound    pv          512m       RWX            shared
```

Image for post

* 4. Let's see what has changed in the pv we had initially created.

```
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
```

Image for post

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 39

CORRECT TEXT

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 43

CORRECT TEXT

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 46

CORRECT TEXT

Score: 5%



Task

Monitor the logs of pod bar and:

- Extract log lines corresponding to error file-not-found
- Write them to /opt/KUTR00101/bar

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:

```
kubect1 logs bar | grep 'unable-to-access-website' > /opt/KUTR00101/bar  
cat /opt/KUTR00101/bar
```

NEW QUESTION 50

CORRECT TEXT

List all the pods sorted by created timestamp

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

```
kubect1 get pods--sort-by=.metadata.creationTimestamp
```

NEW QUESTION 51

CORRECT TEXT

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

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

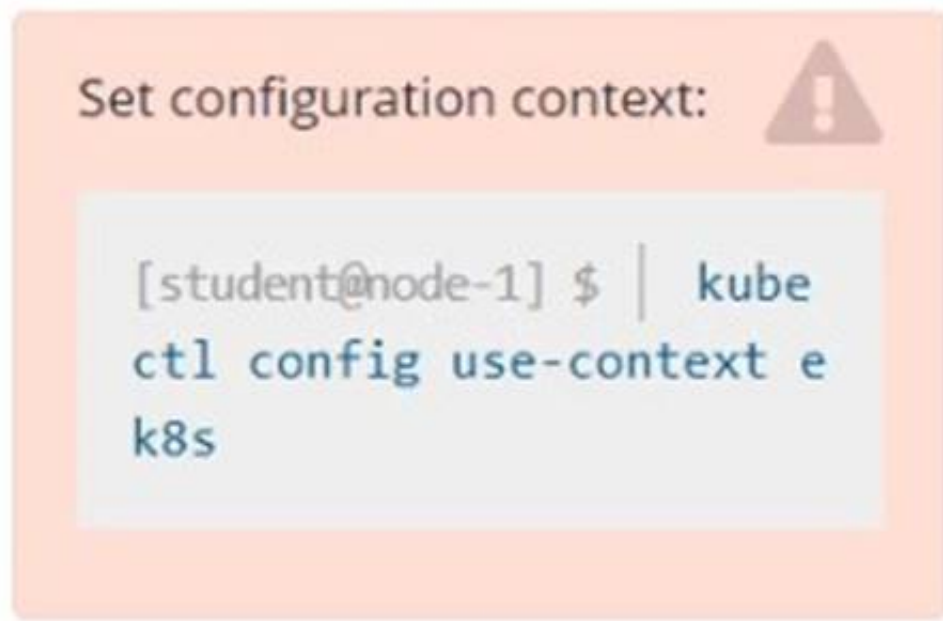
```
kubect1 get pods -o wide
```

```
kubect1 delete po “nginx-dev”kubect1 delete po “nginx-prod”
```

NEW QUESTION 53

CORRECT TEXT

Score: 4%



Task

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:

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

```
kubectl cordon ek8s-node-1
```

```
kubectl drain ek8s-node-1 --delete-local-data --ignore-daemonsets --force
```

NEW QUESTION 55

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual CKA Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the CKA Product From:

<https://www.2passeasy.com/dumps/CKA/>

Money Back Guarantee

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