

Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

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



NEW QUESTION 1

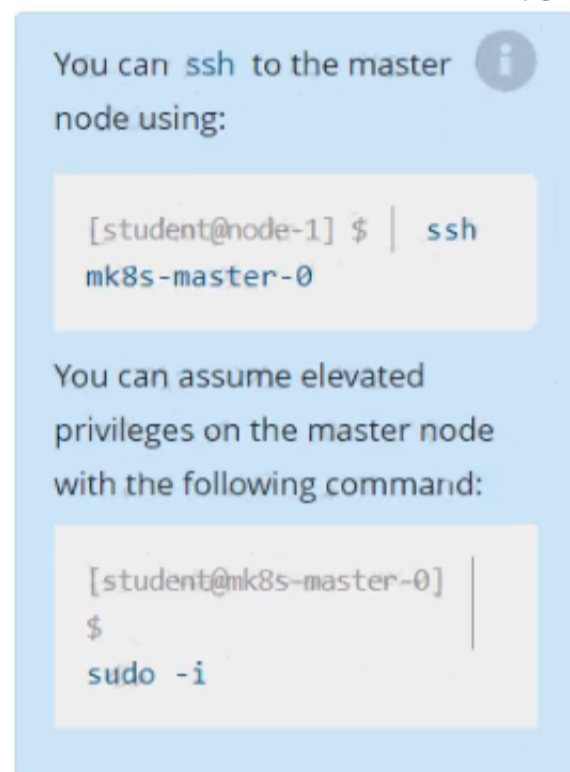
Score: 7%



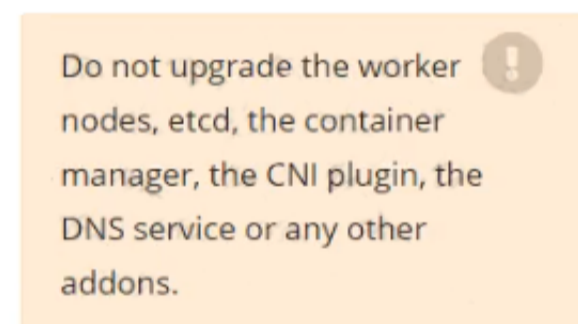
Task

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

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

NEW QUESTION 3

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 4

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

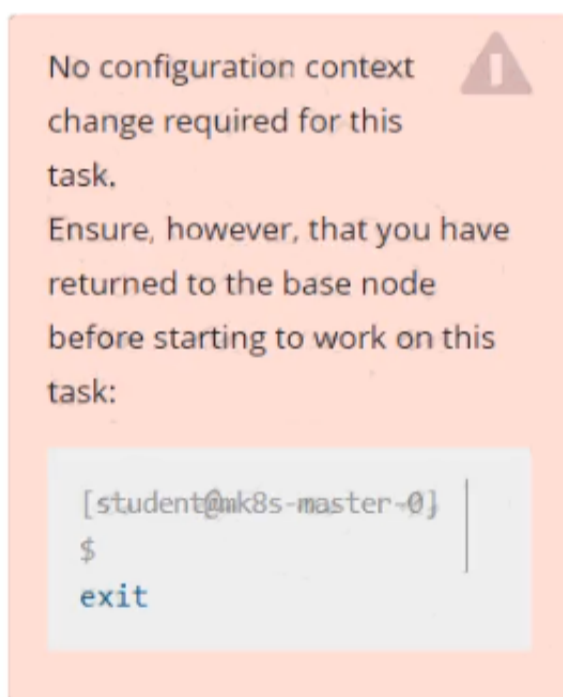
Answer: A

Explanation:

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

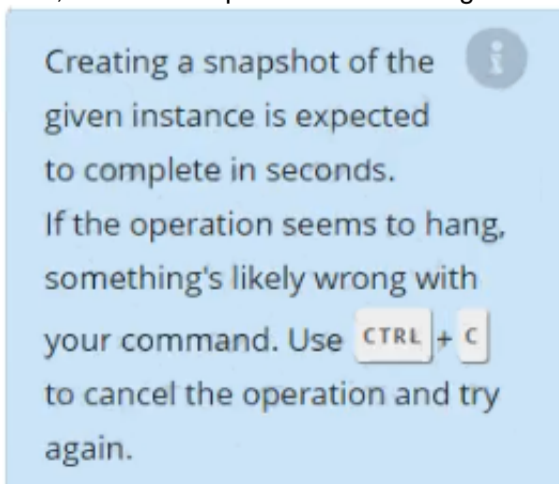
NEW QUESTION 5

Score: 7%

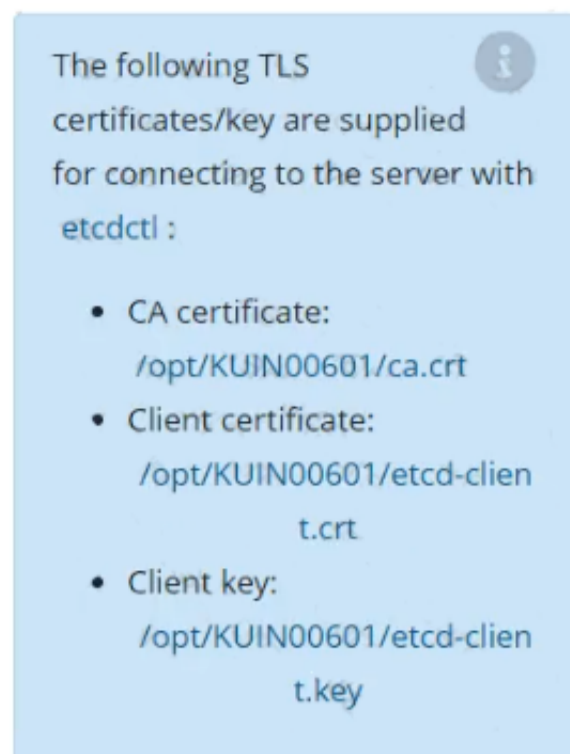


Task

First, create a snapshot of the existing etcd instance running at <https://127.0.0.1:2379>, saving the snapshot to /srv/data/etcd-snapshot.db.



Next, restore an existing, previous snapshot located at /var/lib/backup/etcd-snapshot-previo us.db



- A. Mastered
 B. Not Mastered

Answer: A

Explanation:

Solution:
 #backup
 ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt --key=/opt/KUIN000601/etcd-client.key snapshot save /etc/data/etcd-snapshot.db
 #restore
 ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt --key=/opt/KUIN000601/etcd-client.key snapshot restore /var/lib/backup/etcd-snapshot-previoys.db

NEW QUESTION 6

Score: 4%



Task
 Check to see how many nodes are ready (not including nodes tainted NoSchedule) and write the number to /opt/KUSC00402/kusc00402.txt.

- A. Mastered
 B. Not Mastered

Answer: A

Explanation:

Solution:
 kubectl describe nodes | grep ready|wc -l
 kubectl describe nodes | grep -i taint | grep -i noschedule |wc -l echo 3 > /opt/KUSC00402/kusc00402.txt
 #
 kubectl get node | grep -i ready |wc -l
 # taintsnoSchedule
 kubectl describe nodes | grep -i taints | grep -i noschedule |wc -l
 #
 echo 2 > /opt/KUSC00402/kusc00402.txt

NEW QUESTION 7

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

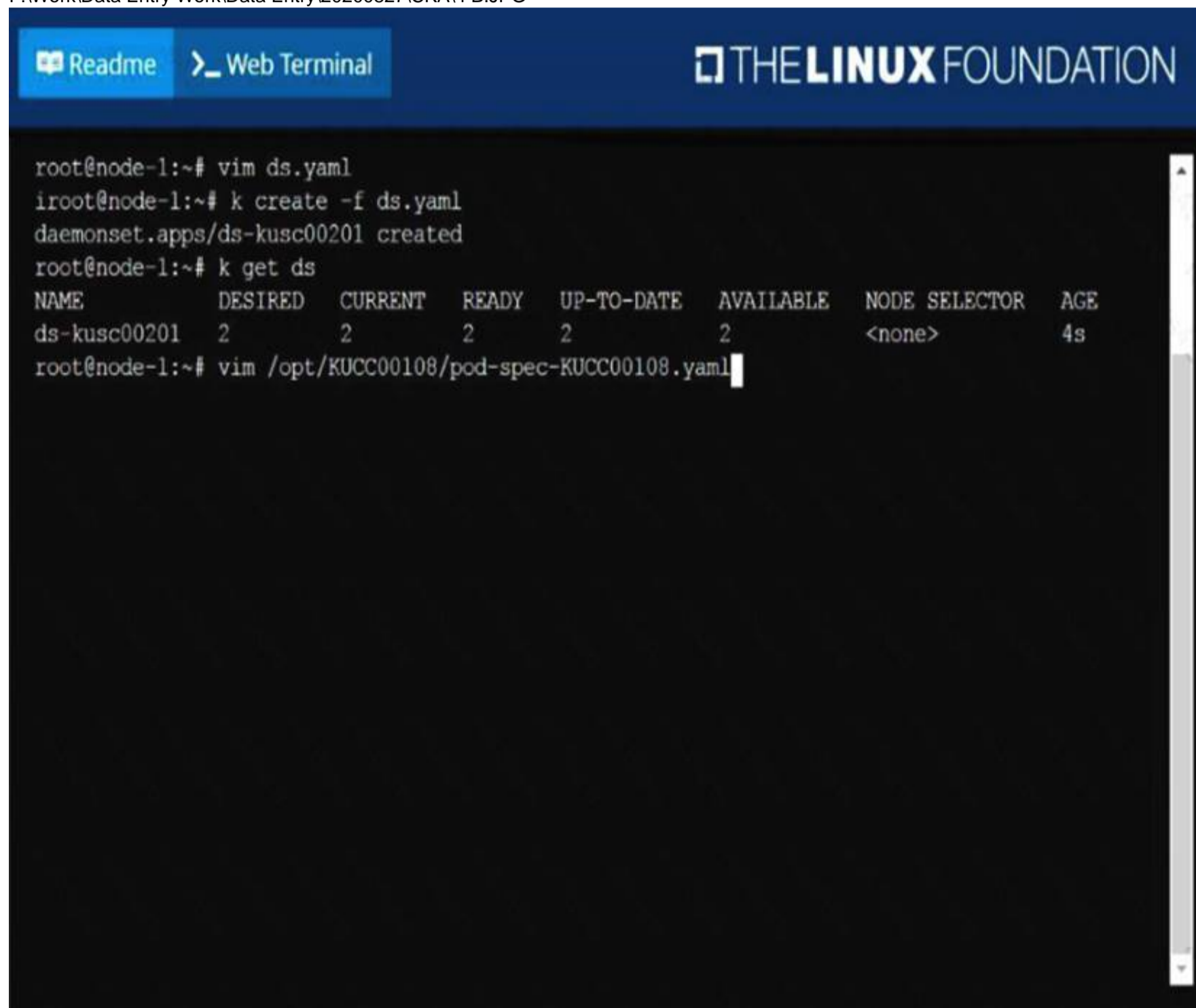
- A. Mastered
B. Not Mastered

Answer: A

Explanation:

solution

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



```
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

```

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

```

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

Readme
Web Terminal

```

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 8

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 9

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
```

NEW QUESTION 10

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 10

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 13

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 17

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 19

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 23

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.

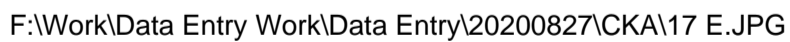
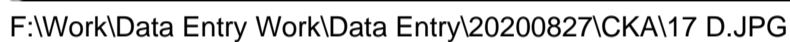
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

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



Readme
Web Terminal
THE **LINUX** FOUNDATION

```

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 27

Create a busybox pod and add “sleep 3600” command

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubect! run busybox --image=busybox --restart=Never -- /bin/sh -c "sleep 3600"

NEW QUESTION 28

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

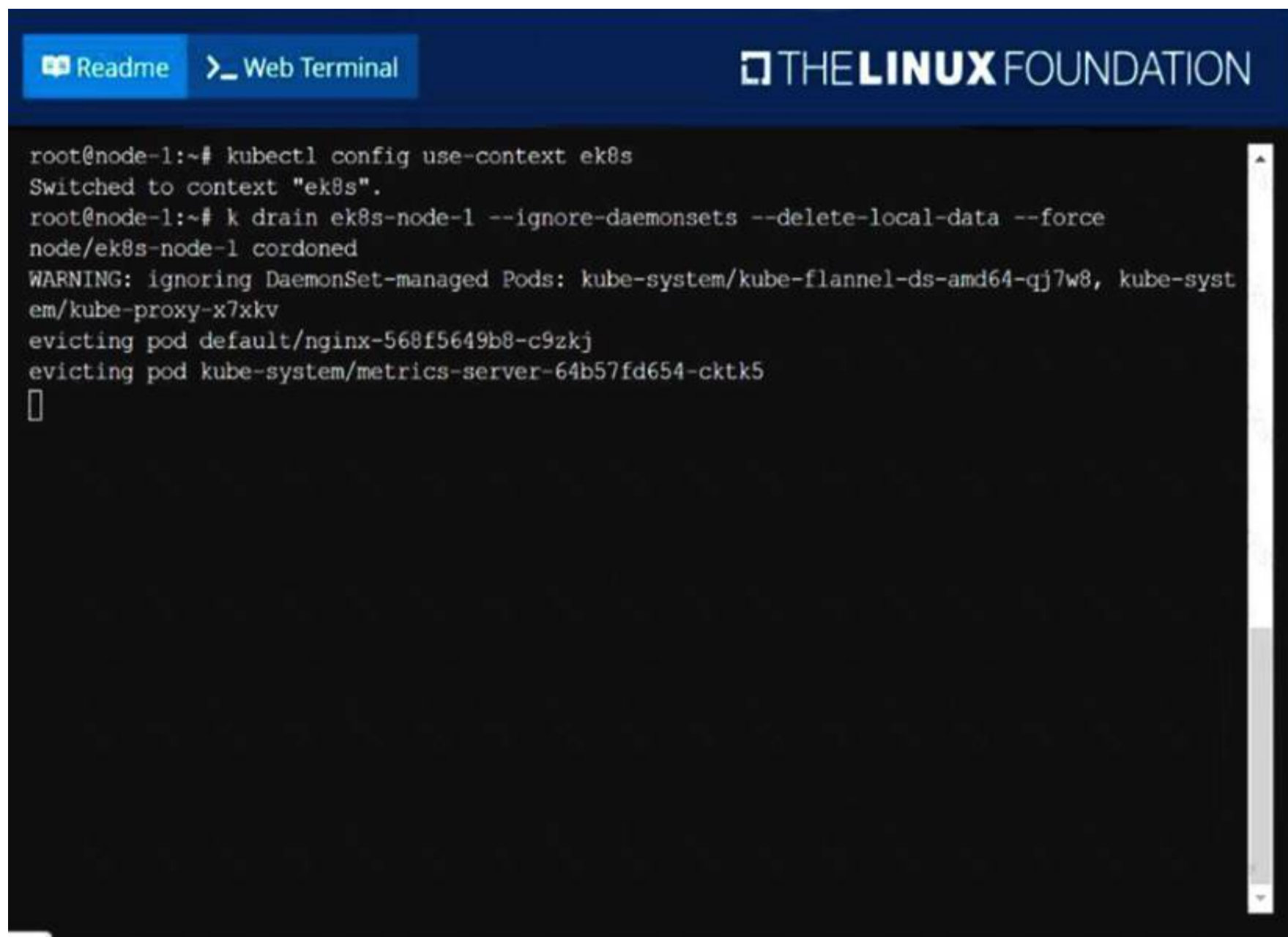
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

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



The screenshot shows a web terminal window with a dark background. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active. In the top right corner, the 'THE LINUX FOUNDATION' logo is visible. The terminal displays the following commands and output:

```
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-syst
em/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zkj
evicting pod kube-system/metrics-server-64b57fd654-cktk5
█
```

NEW QUESTION 29

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.

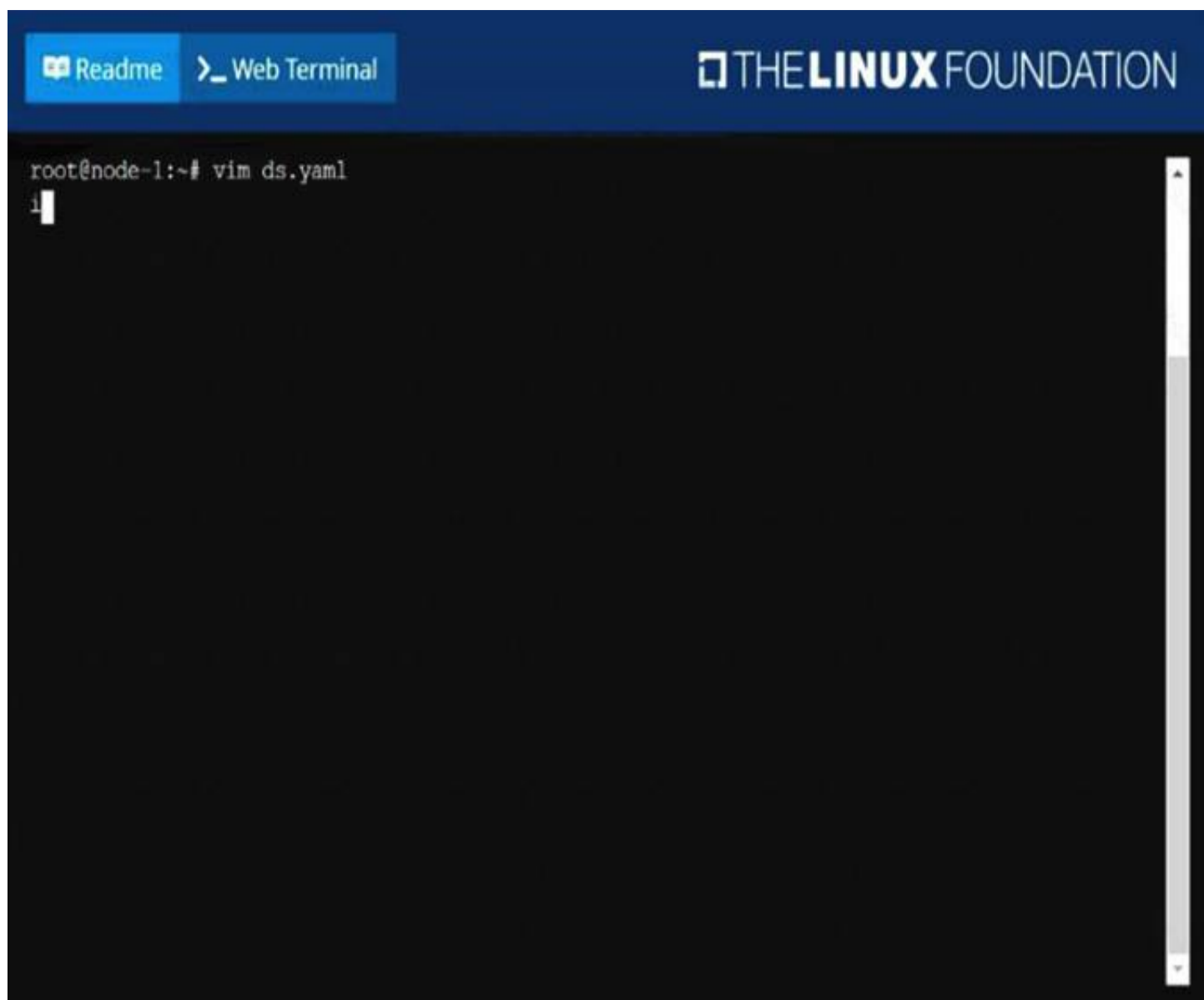
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

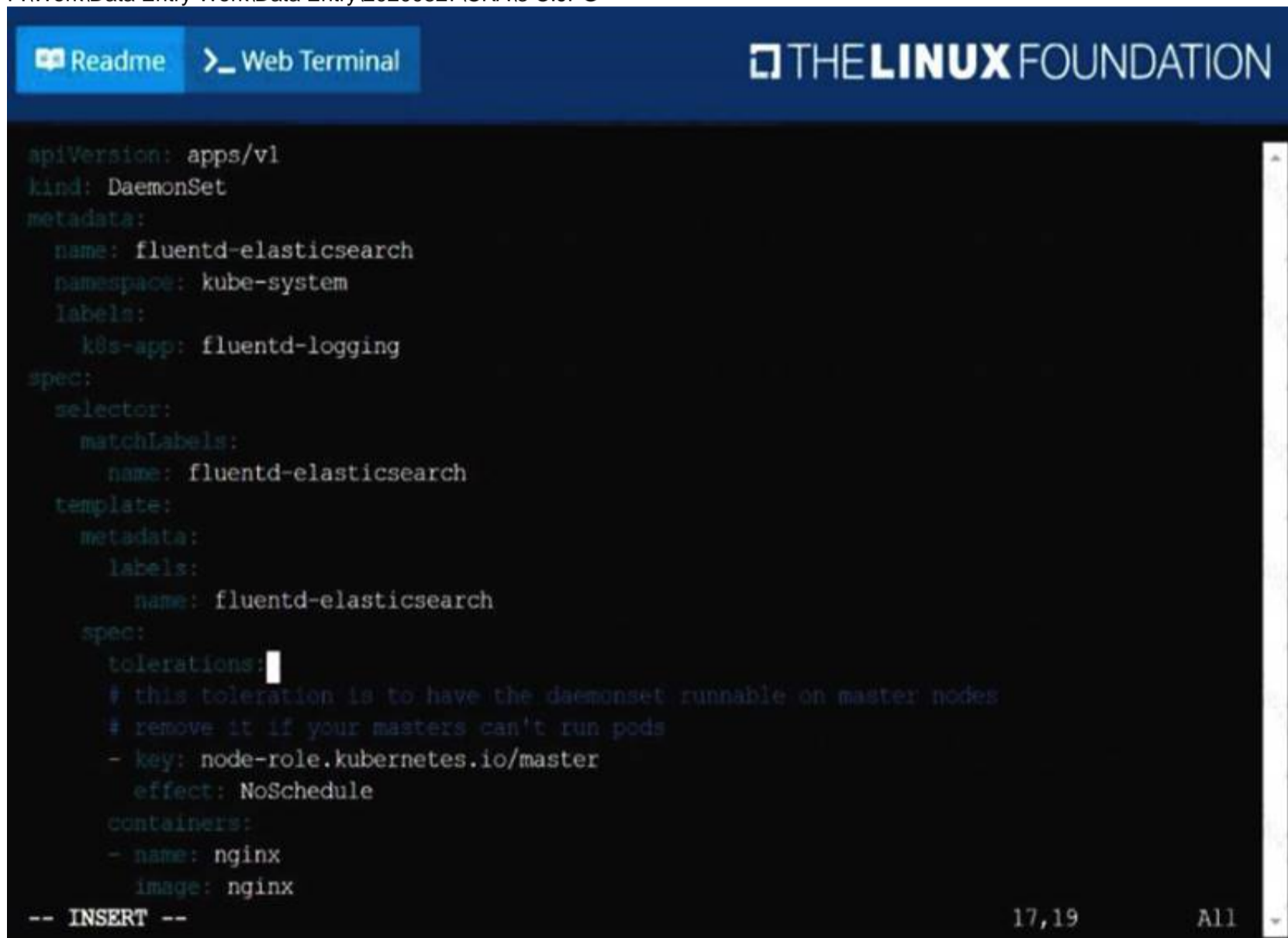
F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 B.JPG



The screenshot shows a web terminal window 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 content shows a user at the root of a node-1 machine opening a vim editor to edit 'ds.yaml'. The cursor is at the beginning of the first line.

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

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



The screenshot shows a web terminal window 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 content shows a user at the root of a node-1 machine opening a vim editor to edit a Kubernetes DaemonSet YAML file. The cursor is at the end of the 'tolerations' section.

```
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 --
```

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

The screenshot shows a web terminal interface with a dark background. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active, displaying a Kubernetes manifest for a DaemonSet named 'ds-kusc00201'. The manifest is in YAML format and defines a DaemonSet for fluentd-elasticsearch. The 'spec' section includes a 'selector' with 'matchLabels' for 'name: fluentd-elasticsearch' and a 'template' with 'metadata' labels for 'name: fluentd-elasticsearch' and a 'spec' section with a 'containers' list containing one container named 'nginx' with image 'nginx'. The terminal shows the manifest content line by line, with a cursor at the bottom.

```
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
```

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

The screenshot shows a web terminal interface with a dark background. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active, displaying a series of commands and their outputs in a terminal session. The commands include creating a DaemonSet from a manifest file, and then getting the status of the DaemonSet. The output shows the DaemonSet 'ds-kusc00201' is created and its status is 'Ready'.

```
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:~#
```

NEW QUESTION 31

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: 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 classname 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-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: 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 32

Create a Kubernetes secret as follows:

- > Name: super-secret
- > password: bob

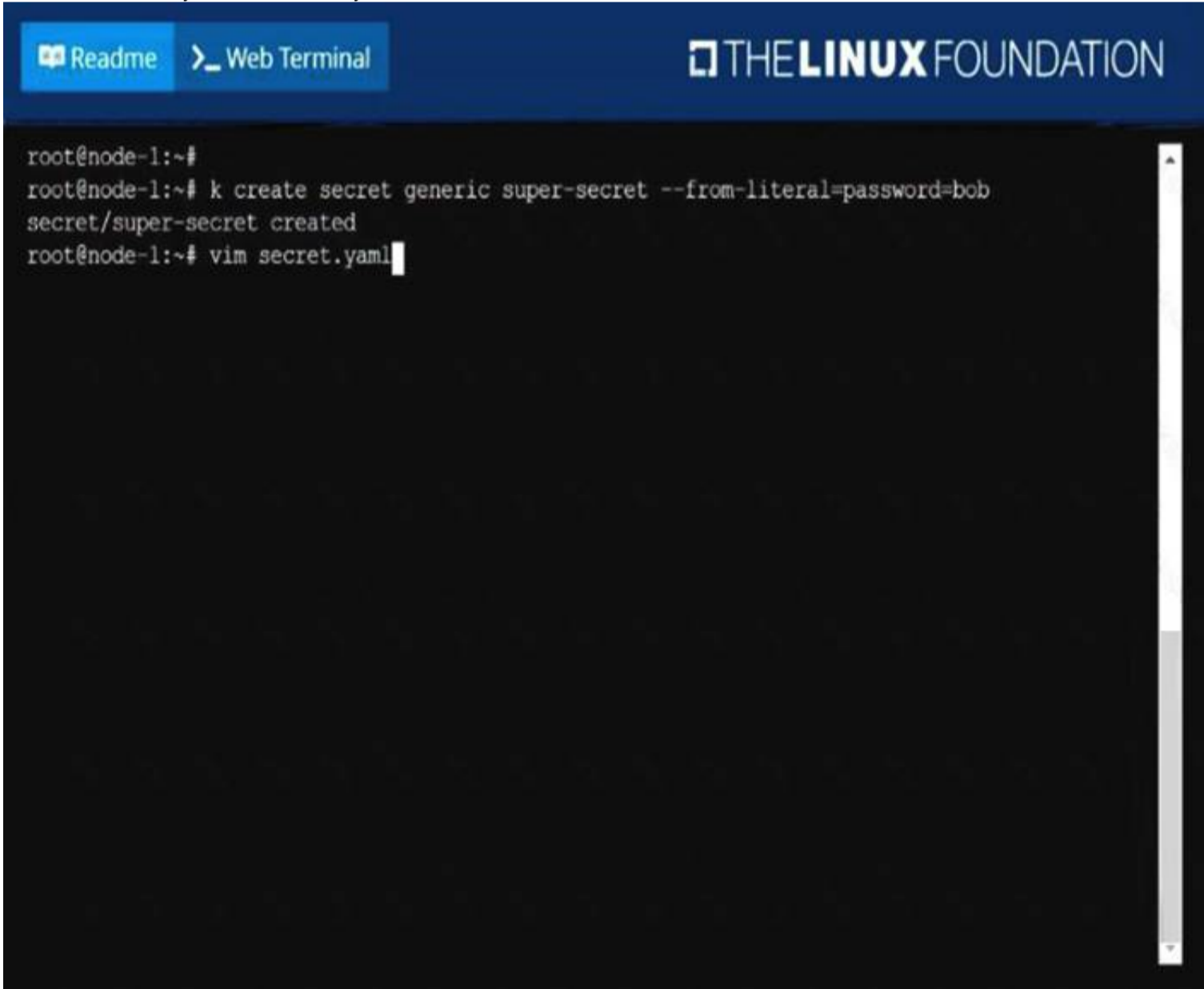
Create a pod named pod-secrets-via-file, using the redis Image, which mounts a secret named super-secret at /secrets.
Create a second pod named pod-secrets-via-env, using the redis Image, which exports password as
CONFIDENTIAL

- 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

Readme
Web Terminal

```

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

```

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:~#

```

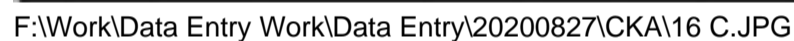
NEW QUESTION 33

From the pod label name=cpu-utilizer, find pods running high CPU workloads and write the name of the pod consuming most CPU to the file /opt/KUTR00102/KUTR00102.txt (which already exists).

- A. Mastered
- B. Not Mastered

Explanation:

F:\Work\Data Entry Work\Data Entry\20200827\CKA\16 B.JPG



Determine the node, the failing service, and take actions to bring up the failed service and restore the health of the cluster. Ensure that any changes are made permanently.

```
[student@node-1] $ ssh <nodename>
```

```
[student@nodename] $ | sudo -i
```

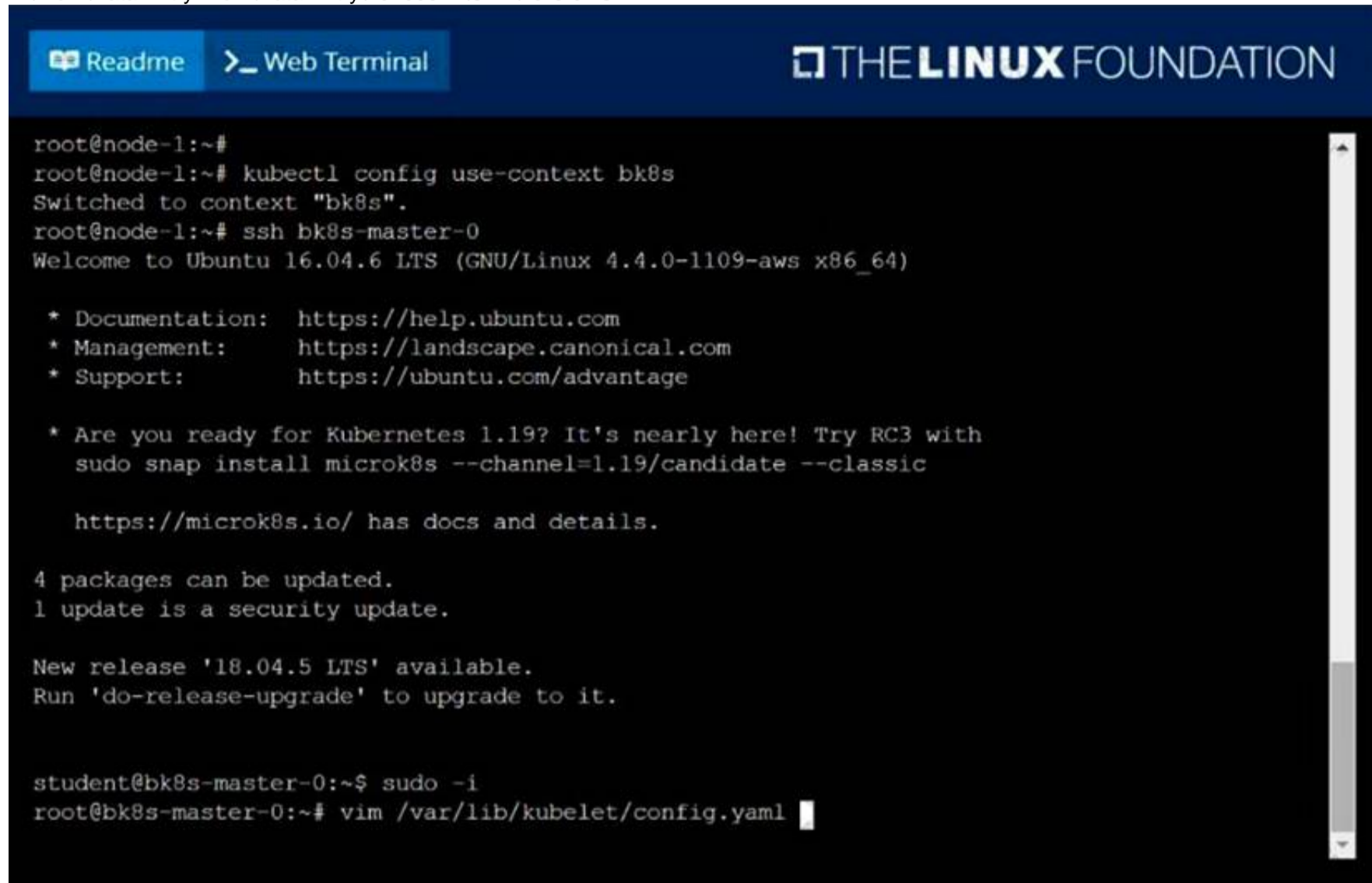
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\23 C.JPG



```
root@node-1:~#
root@node-1:~# kubectl config use-context bk8s
Switched to context "bk8s".
root@node-1:~# ssh bk8s-master-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

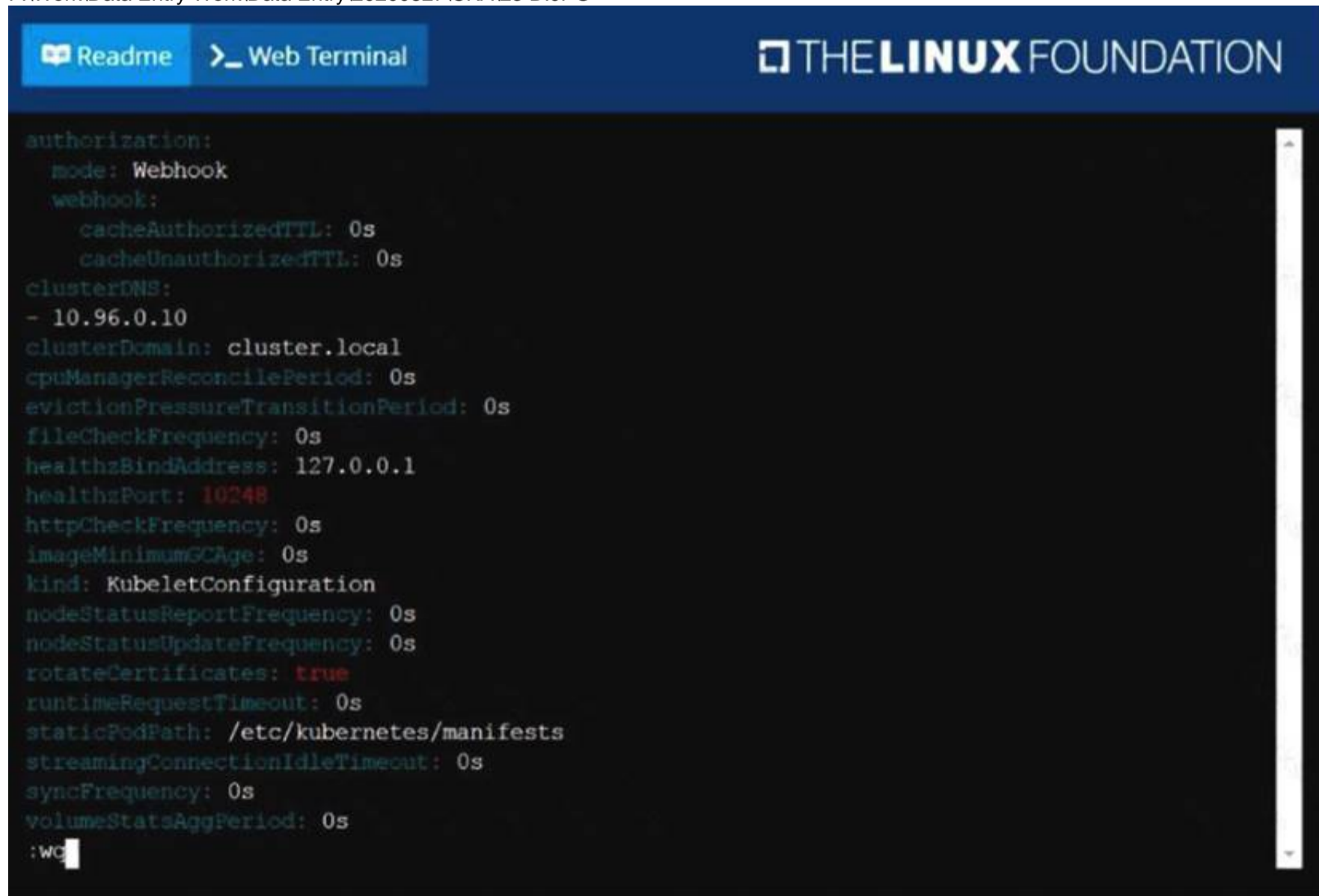
   https://microk8s.io/ has docs and details.

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@bk8s-master-0:~$ sudo -i
root@bk8s-master-0:~# vim /var/lib/kubelet/config.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\23 D.JPG



```
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
syncFrequency: 0s
volumeStatsAggPeriod: 0s
:wc
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\23 E.JPG

Readme
Web Terminal

THE LINUX FOUNDATION

```

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@bk8s-master-0:~$ sudo -i
root@bk8s-master-0:~# vim /var/lib/kubelet/config.yaml
root@bk8s-master-0:~# systemctl restart kubelet
root@bk8s-master-0:~# systemctl enable kubelet
root@bk8s-master-0:~# kubectl get nodes

NAME             STATUS    ROLES    AGE   VERSION
bk8s-master-0    Ready    master   77d   v1.18.2
bk8s-node-0      Ready    <none>   77d   v1.18.2
root@bk8s-master-0:~#
root@bk8s-master-0:~# exit
logout
student@bk8s-master-0:~$ exit
logout
Connection to 10.250.4.77 closed.
root@node-1:~#

```

NEW QUESTION 42

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 46

.....

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