

# Linux Foundation

## CKA Exam

### Certified Kubernetes Administrator (CKA) Program



Thank you for Downloading CKA exam PDF Demo

You can Buy Latest CKA Full Version Download

<https://www.certkillers.net/Exam/CKA>

<https://www.certkillers.net>

# Version: 8.0

---

## Question: 1

---

Monitor the logs of pod foo and:  
Extract log lines corresponding to error  
unable-to-access-website  
Write `cat /opt/KULM00201/fo`

them

to

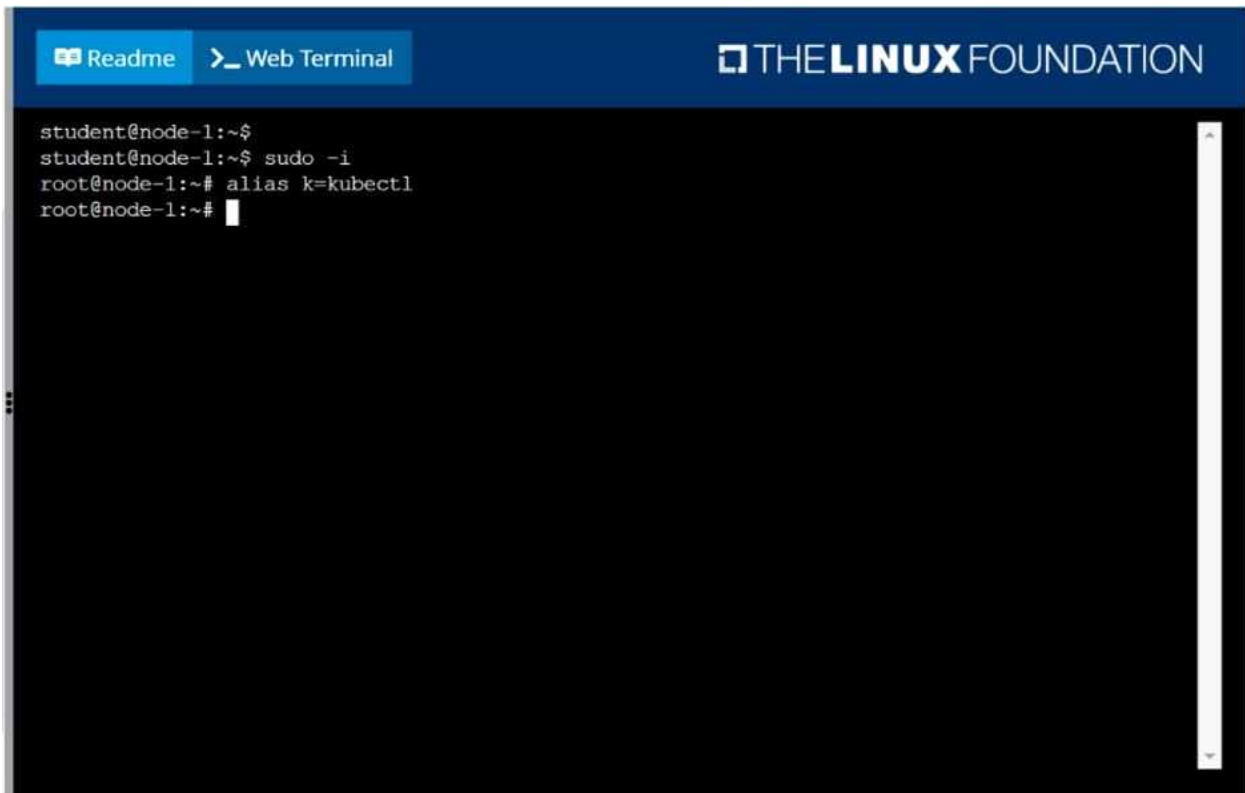


---

**Answer: See the solution below.**

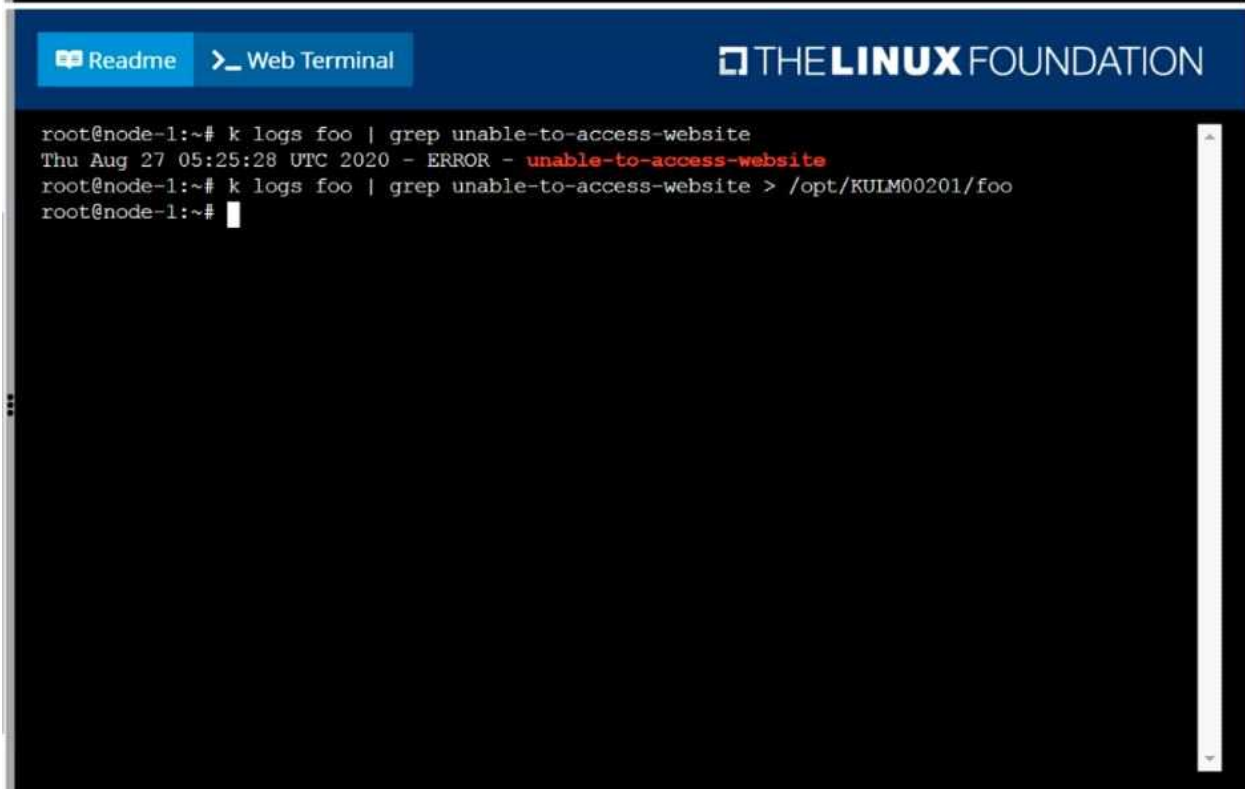
---

Explanation:  
solution



The screenshot shows a web terminal window with a dark background and a blue header. The header contains a 'Readme' button, a 'Web Terminal' button, and the 'THE LINUX FOUNDATION' logo. The terminal text shows a user named 'student' at 'node-1' who runs 'sudo -i' to become 'root'. The root user then sets an alias 'k=kubect1'.

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



The screenshot shows a web terminal window with a dark background and a blue header. The header contains a 'Readme' button, a 'Web Terminal' button, and the 'THE LINUX FOUNDATION' logo. The terminal text shows a user named 'root' at 'node-1' who runs 'k logs foo | grep unable-to-access-website'. The output shows a log entry: 'Thu Aug 27 05:25:28 UTC 2020 - ERROR - unable-to-access-website'. The user then runs 'k logs foo | grep unable-to-access-website > /opt/KULM00201/foo'.

```
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:~#
```

---

Question: 2

---

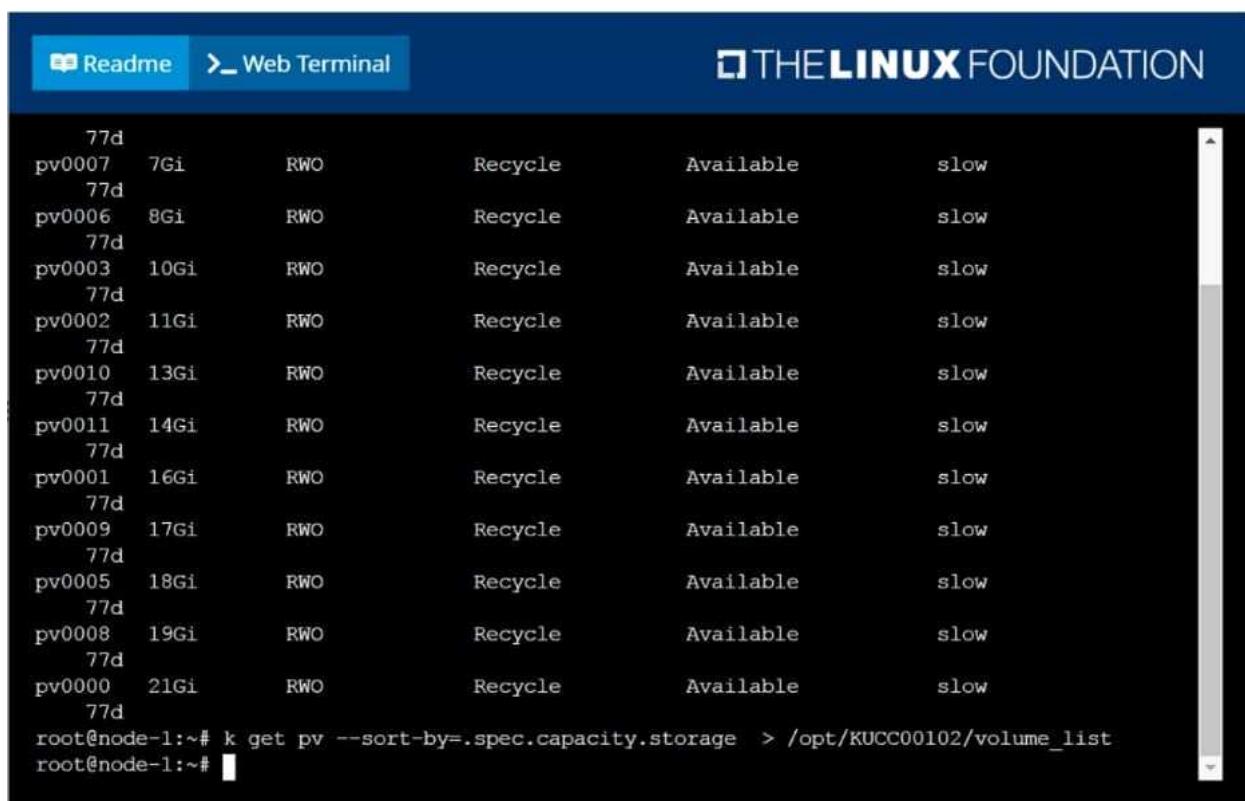
List all persistent volumes sorted by capacity, saving the full kubectl output to /opt/KUCC00102/volume\_list. Use kubectl's own functionality for sorting the output, and do not manipulate it any further.

---

**Answer: See the solution below.**

---

Explanation:  
solution



```
Readme >_ Web Terminal THE LINUX FOUNDATION
77d
pv0007 7Gi RWO Recycle Available slow
77d
pv0006 8Gi RWO Recycle Available slow
77d
pv0003 10Gi RWO Recycle Available slow
77d
pv0002 11Gi RWO Recycle Available slow
77d
pv0010 13Gi RWO Recycle Available slow
77d
pv0011 14Gi RWO Recycle Available slow
77d
pv0001 16Gi RWO Recycle Available slow
77d
pv0009 17Gi RWO Recycle Available slow
77d
pv0005 18Gi RWO Recycle Available slow
77d
pv0008 19Gi RWO Recycle Available slow
77d
pv0000 21Gi RWO Recycle Available slow
77d
root@node-1:~# k get pv --sort-by=.spec.capacity.storage > /opt/KUCC00102/volume_list
root@node-1:~#
```

---

### Question: 3

---

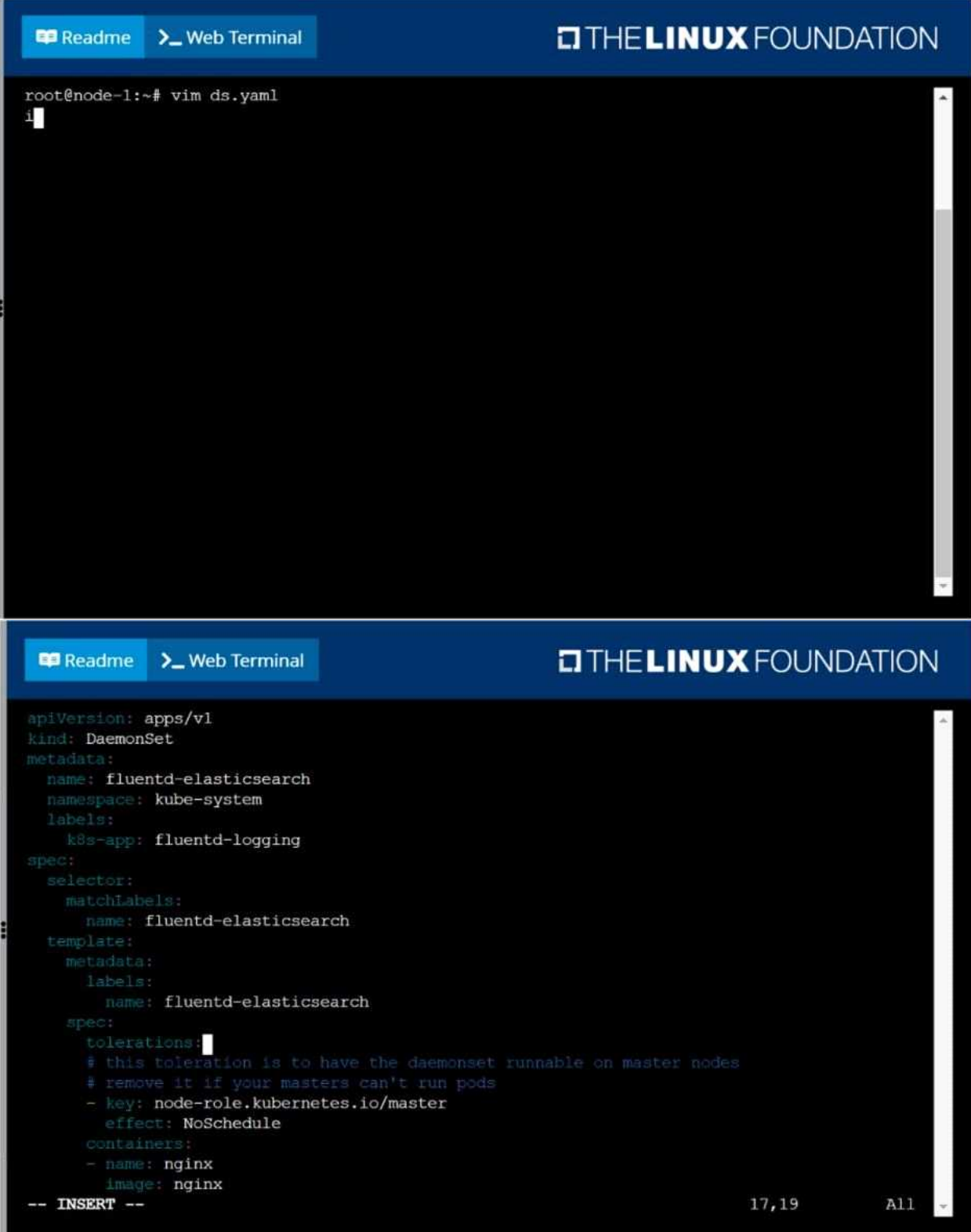
Ensure a single instance of pod nginx is running on each node of the Kubernetes cluster where nginx also represents the Image name which has to be used. Do not override any taints currently in place. Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

---

**Answer: See the solution below.**

---

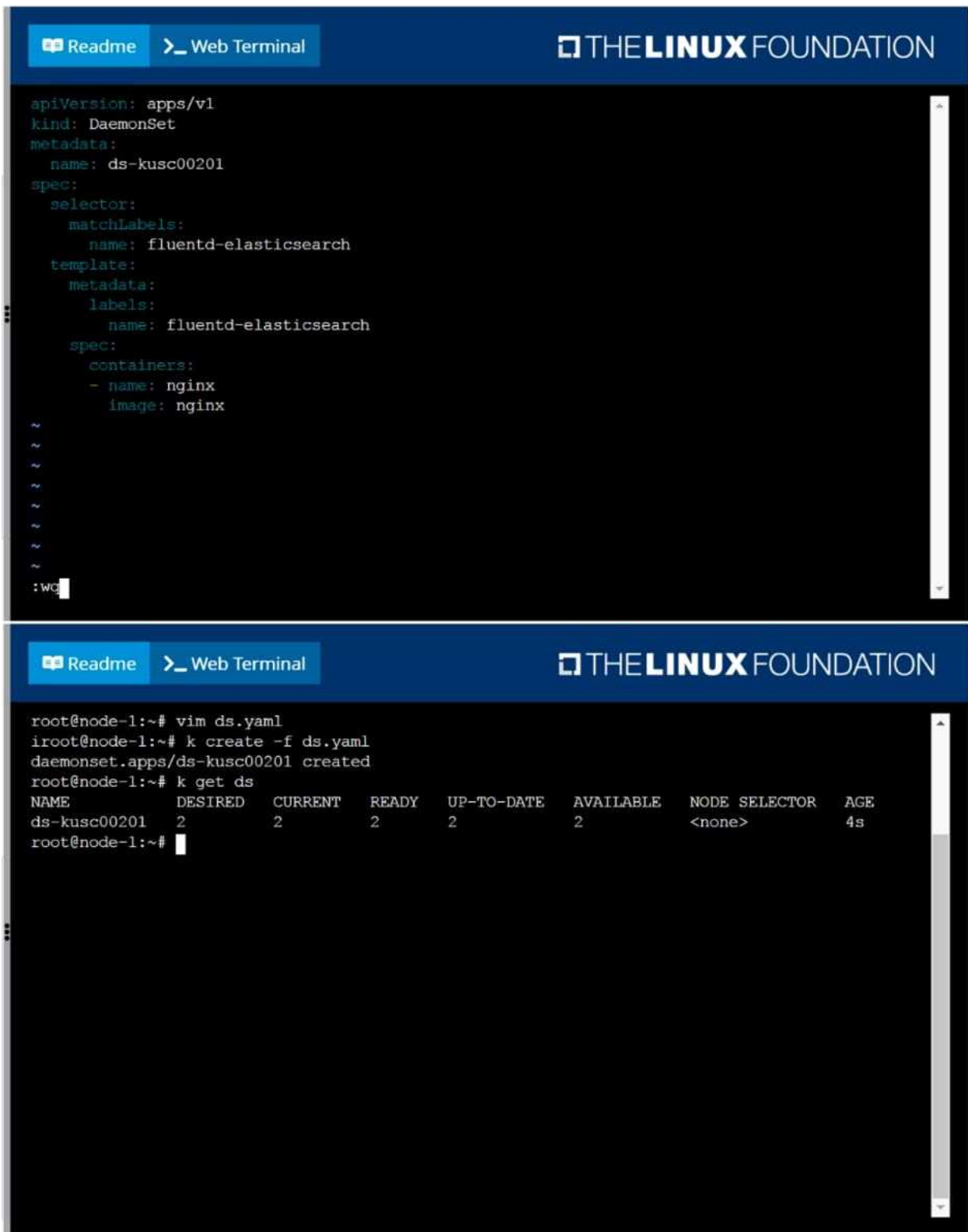
Explanation:  
solution



The image shows two screenshots of a web terminal interface from The Linux Foundation. The top screenshot shows the terminal prompt `root@node-1:~# vim ds.yaml` with the character `i` entered, indicating the start of an insert mode in vim. The bottom screenshot shows the contents of `ds.yaml` being edited, which defines a DaemonSet for `fluentd-elasticsearch` in the `kube-system` namespace. The configuration includes labels, a selector, and a template with a container named `nginx`. A toleration is also defined to allow the DaemonSet to run on master nodes. The terminal shows the cursor at the end of the `tolerations` list.

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

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 --
```



The screenshot displays a web terminal interface for The Linux Foundation. The terminal is divided into two sections. The top section shows the content of a Kubernetes DaemonSet manifest file named `ds-kusc00201`. The manifest defines a DaemonSet with the following configuration:

```
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
```

The bottom section shows the terminal output of the following commands:

```
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

The terminal prompt is `root@node-1:~#`.

---

**Question: 4**

---

Perform the following tasks:

Add an init container to hungry-bear (which has been defined in spec file /opt/KUCC00108/pod-spec-KUC

C00108.yaml

)

The init container should create an empty file named /workdir/calm.txt

If /workdir/calm.txt is not detected, the pod should exit

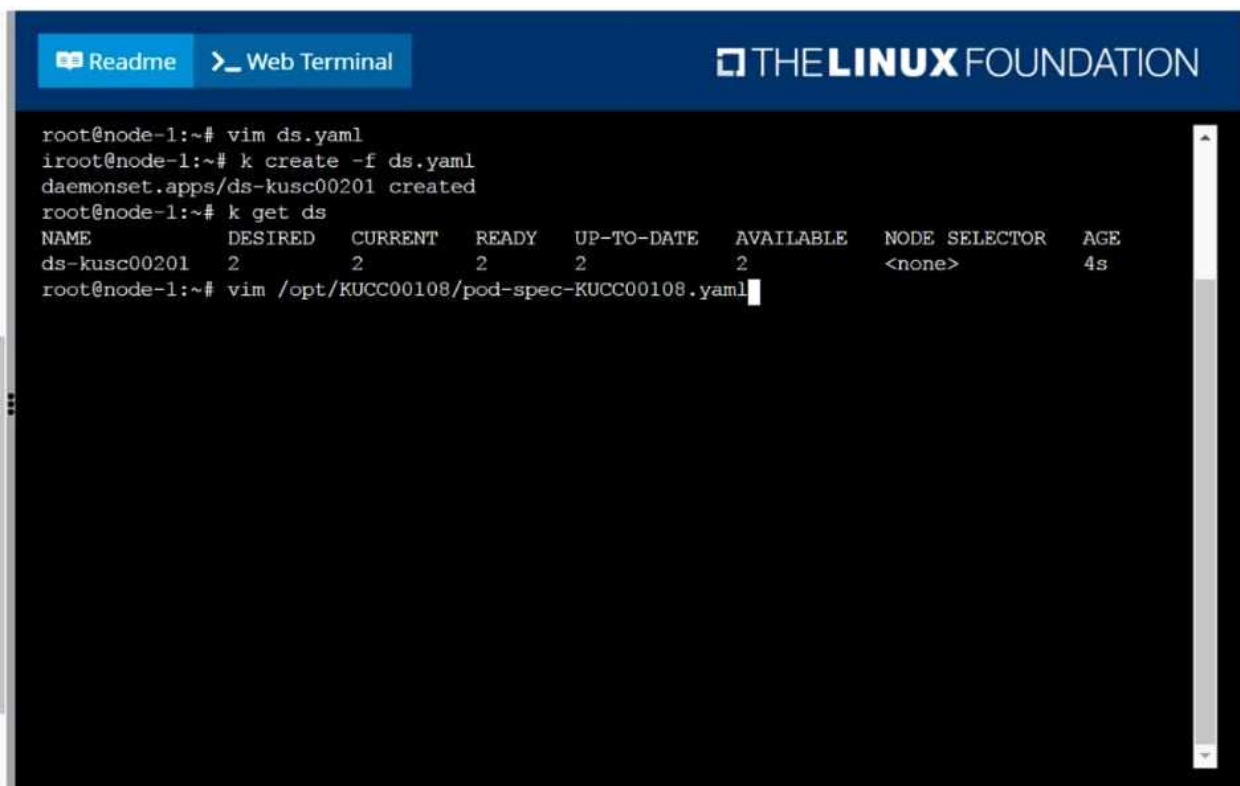
Once the spec file has been updated with the init container definition, the pod should be created

---

**Answer: See the solution below.**

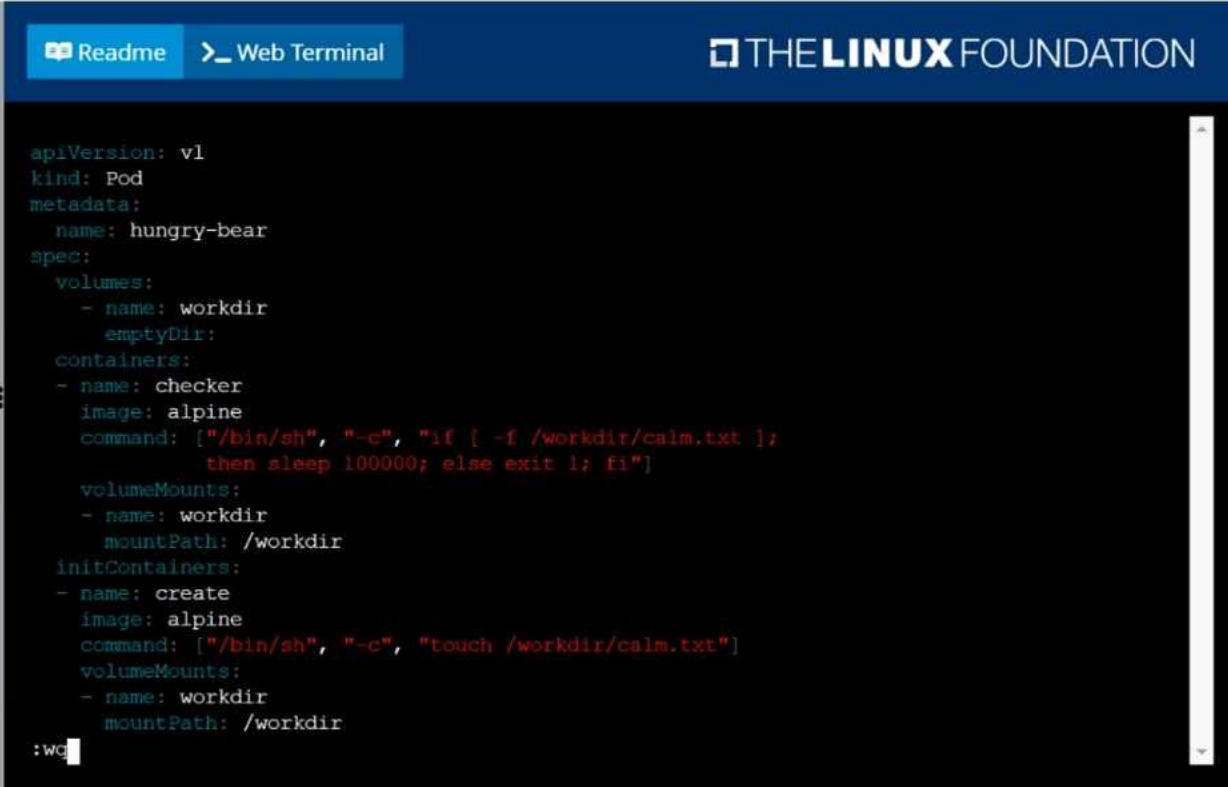
---

Explanation:  
solution




The screenshot shows a web terminal interface with a dark background and light text. 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 output shows the following sequence of commands and results:

```
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
```



The screenshot shows a web terminal interface with a dark background. At the top, there are two buttons: 'Readme' and 'Web Terminal'. The 'THE LINUX FOUNDATION' logo is in the top right corner. The terminal displays a YAML configuration for a Kubernetes pod named 'hungry-bear'. The configuration includes metadata, a spec with volumes and containers, and an initContainer. The terminal prompt is ':wq'.

```
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
```



The screenshot shows a web terminal interface with a dark background. At the top, there are two buttons: 'Readme' and 'Web Terminal'. The 'THE LINUX FOUNDATION' logo is in the top right corner. The terminal shows a sequence of commands and their outputs in a shell environment. The commands include creating a deployment, getting the deployment status, and creating a pod. The output shows the deployment 'ds-kusc00201' is created and ready, and the pod 'pod/hungry-bear' is created.

```
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:~#
```



---

**Question: 5**

---

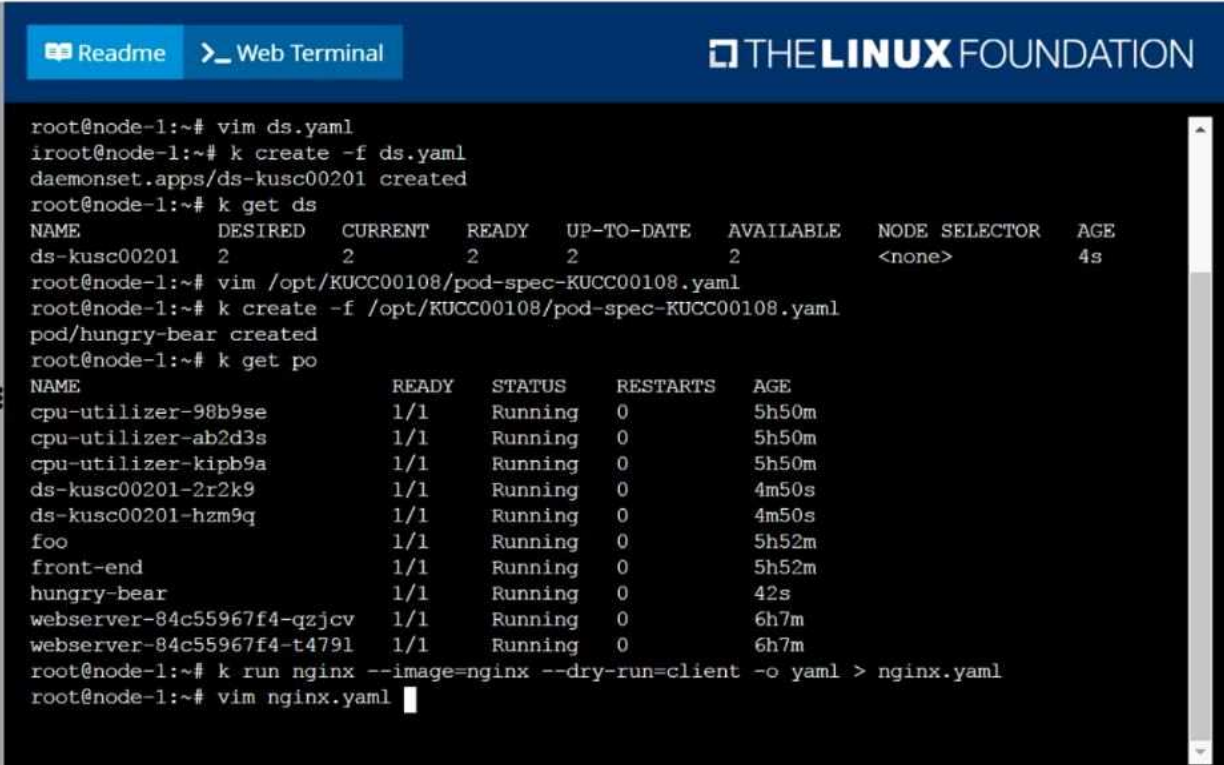
Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified):  
nginx + redis + memcached.

---

**Answer: See the solution below.**

---

Explanation:  
solution



The screenshot shows a terminal window with a dark background and light text. At the top, there are tabs for 'Readme' and 'Web Terminal', and the logo for 'THE LINUX FOUNDATION'. The terminal output shows the following sequence of commands and results:

```
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:~# k get po
NAME           READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se   1/1     Running   0           5h50m
cpu-utilizer-ab2d3s   1/1     Running   0           5h50m
cpu-utilizer-kipb9a   1/1     Running   0           5h50m
ds-kusc00201-2r2k9    1/1     Running   0           4m50s
ds-kusc00201-hzm9q    1/1     Running   0           4m50s
foo               1/1     Running   0           5h52m
front-end         1/1     Running   0           5h52m
hungry-bear        1/1     Running   0           42s
webserver-84c55967f4-qzjcv  1/1     Running   0           6h7m
webserver-84c55967f4-t479l  1/1     Running   0           6h7m
root@node-1:~# k run nginx --image=nginx --dry-run=client -o yaml > nginx.yaml
root@node-1:~# vim nginx.yaml
```





## Thank You for trying CKA PDF Demo

To Buy Latest CKA Full Version Download visit link below

<https://www.certkillers.net/Exam/CKA>

## Start Your CKA Preparation

**[Limited Time Offer]** Use Coupon “CKNET” for Further 10% discount on your purchase. Test your CKA preparation with actual exam questions.

<https://www.certkillers.net>