



Linux Foundation

CKA Exam

Certified Kubernetes Administrator (CKA) Program

Thank you for Downloading CKA exam PDF Demo

You can also try our CKA practice exam software

Download Free Demo

<https://prepbolt.com/CKA.html>

QUESTIONS & ANSWERS

DEMO VERSION

(LIMITED CONTENT)

Version: 8.0

Question: 1

Monitor the logs of pod foo and:
Extract log lines corresponding to error
unable-to-access-website

Write

them

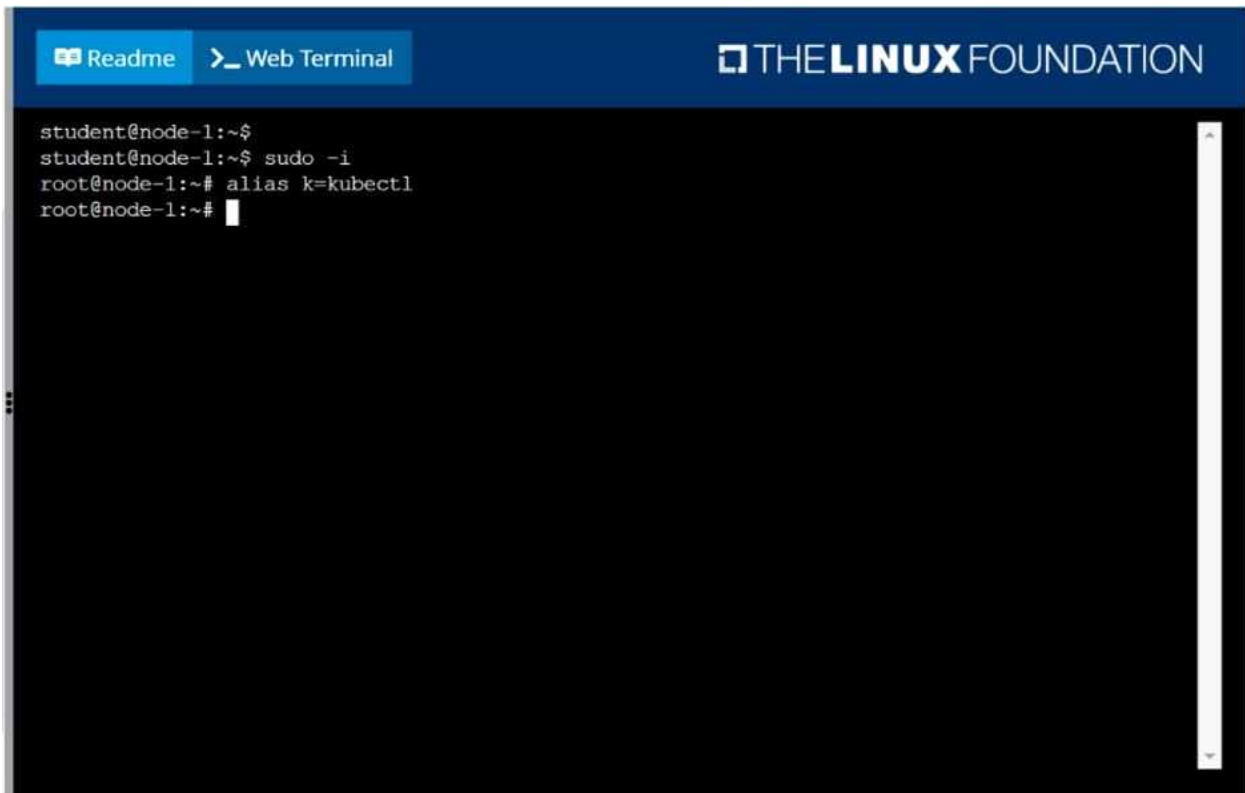
to

/opt/KULM00201/foo



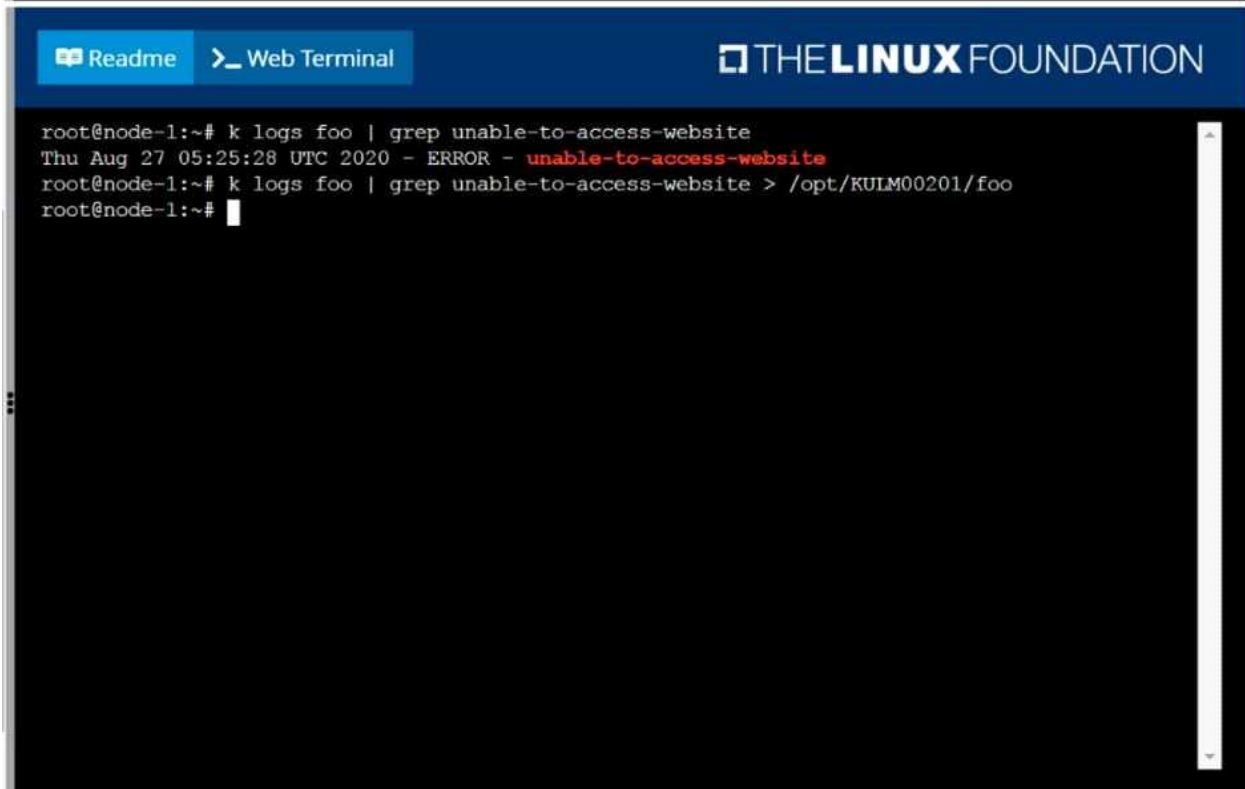
**Answer: See the
solution below.**

Explanation:
solution



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 text 'THE LINUX FOUNDATION' is displayed. The terminal content shows a user named 'student' at 'node-1' in the home directory. They run 'sudo -i' to become root. Then, they run 'alias k=kubect1' to create an alias for the 'kubect1' command. The prompt returns to root@node-1:~#.

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



The screenshot shows the same web terminal window. The user is now root@node-1:~#. They run 'k logs foo | grep unable-to-access-website', which outputs 'Thu Aug 27 05:25:28 UTC 2020 - ERROR - unable-to-access-website'. Then, they run 'k logs foo | grep unable-to-access-website > /opt/KULM00201/foo'. The prompt returns to root@node-1:~#.

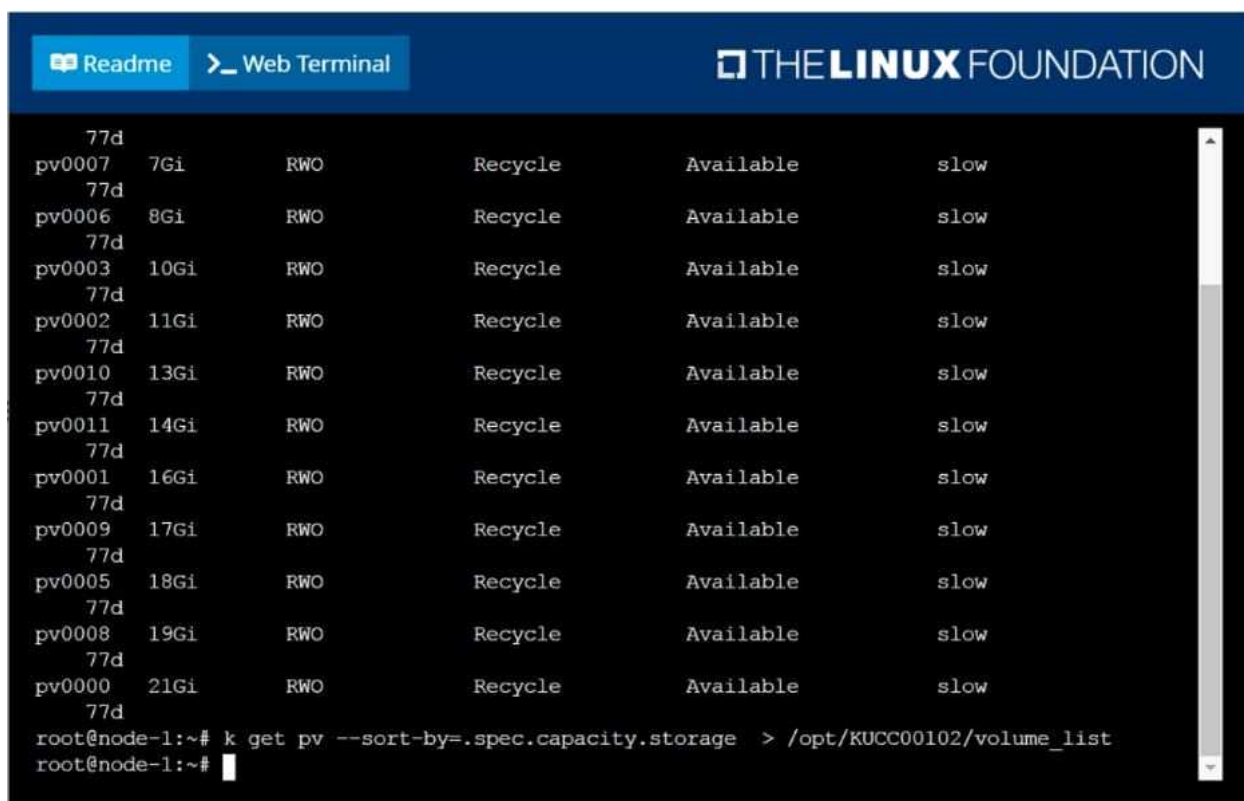
```
root@node-1:~# k logs foo | grep unable-to-access-website
Thu Aug 27 05:25:28 UTC 2020 - ERROR - unable-to-access-website
root@node-1:~# k logs foo | grep unable-to-access-website > /opt/KULM00201/foo
root@node-1:~#
```

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



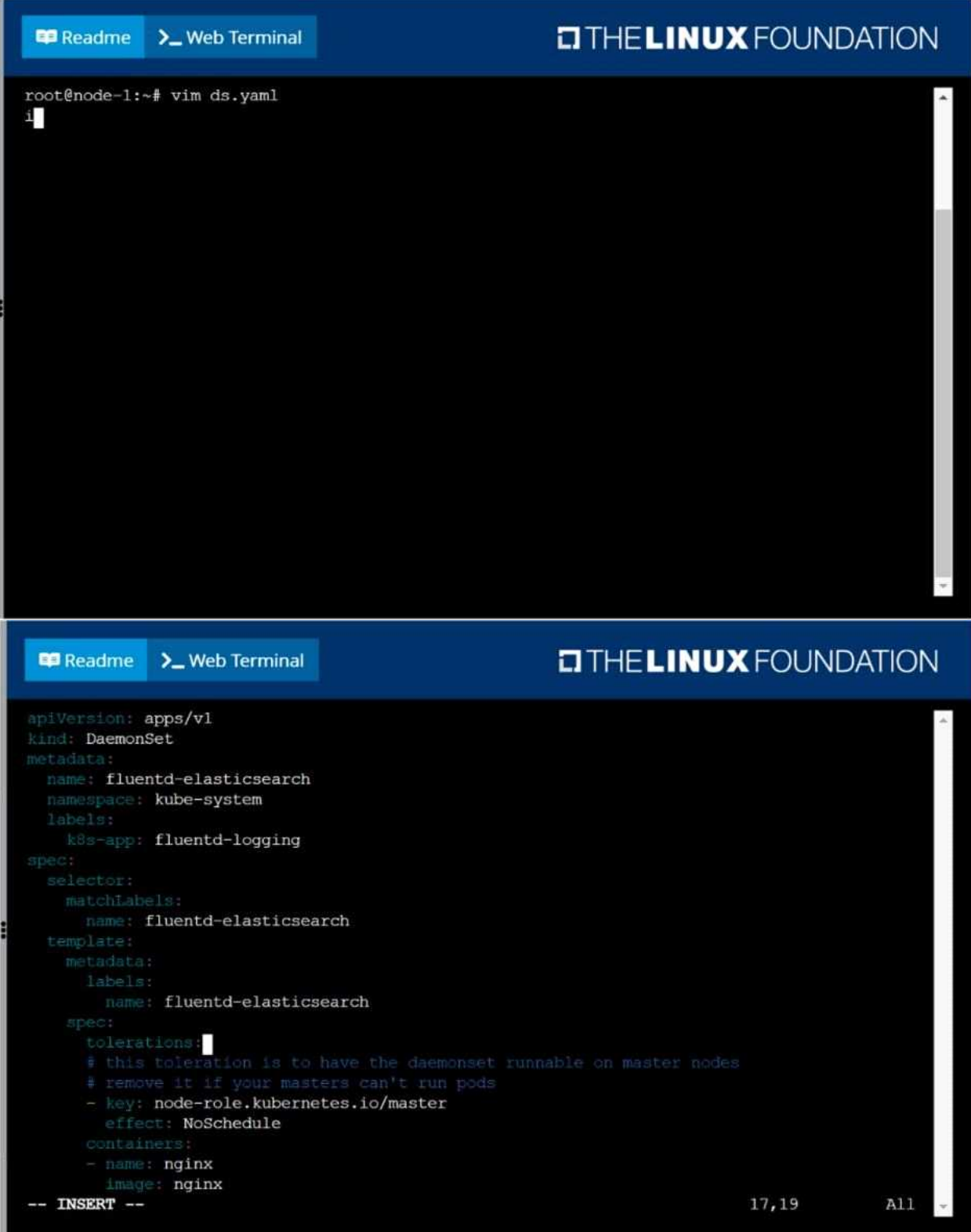
```
Readme >_ Web Terminal THE LINUX FOUNDATION
77d
pv0007 7Gi RWO Recycle Available slow
77d
pv0006 8Gi RWO Recycle Available slow
77d
pv0003 10Gi RWO Recycle Available slow
77d
pv0002 11Gi RWO Recycle Available slow
77d
pv0010 13Gi RWO Recycle Available slow
77d
pv0011 14Gi RWO Recycle Available slow
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



```
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 --
```

Readme Web 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
~
~
~
~
~
~
~
~
~
:wg
    
```

Readme Web 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:~#
    
```

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

)

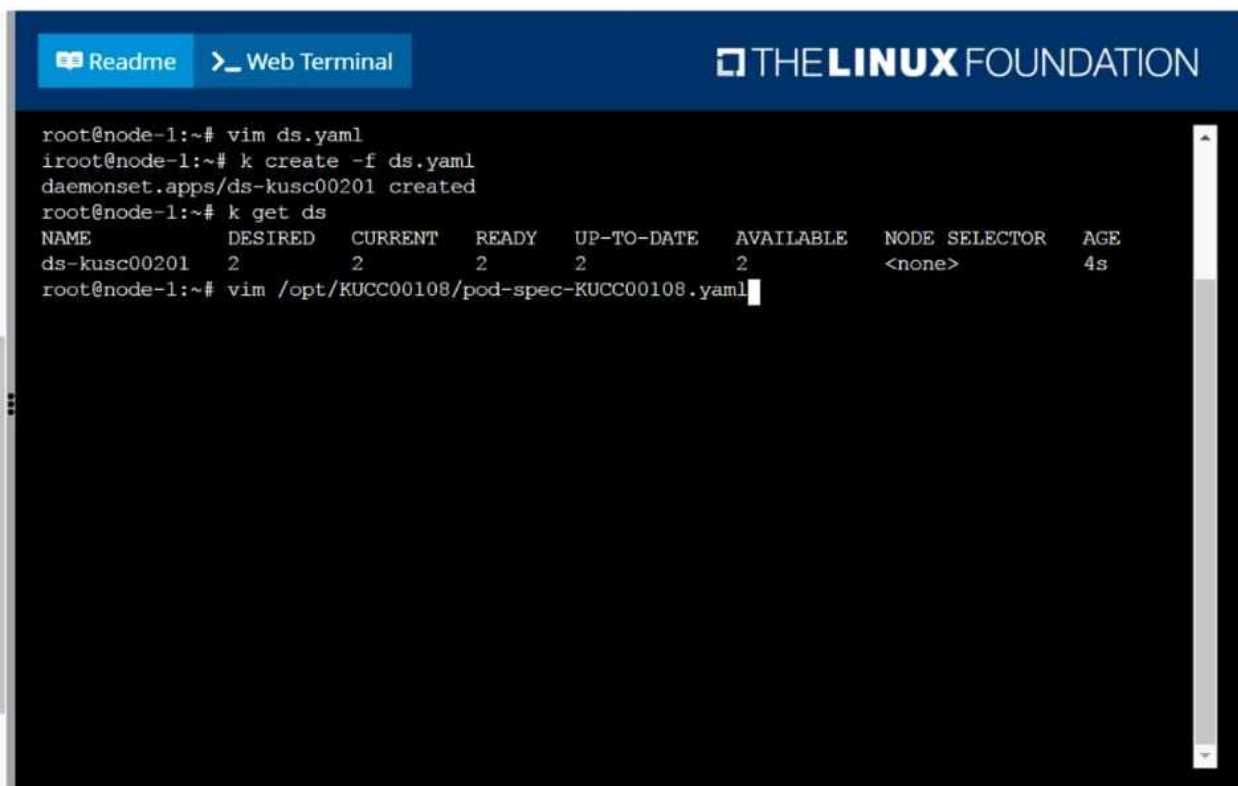
The init container should create an empty file named /workdir/calm.txt

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

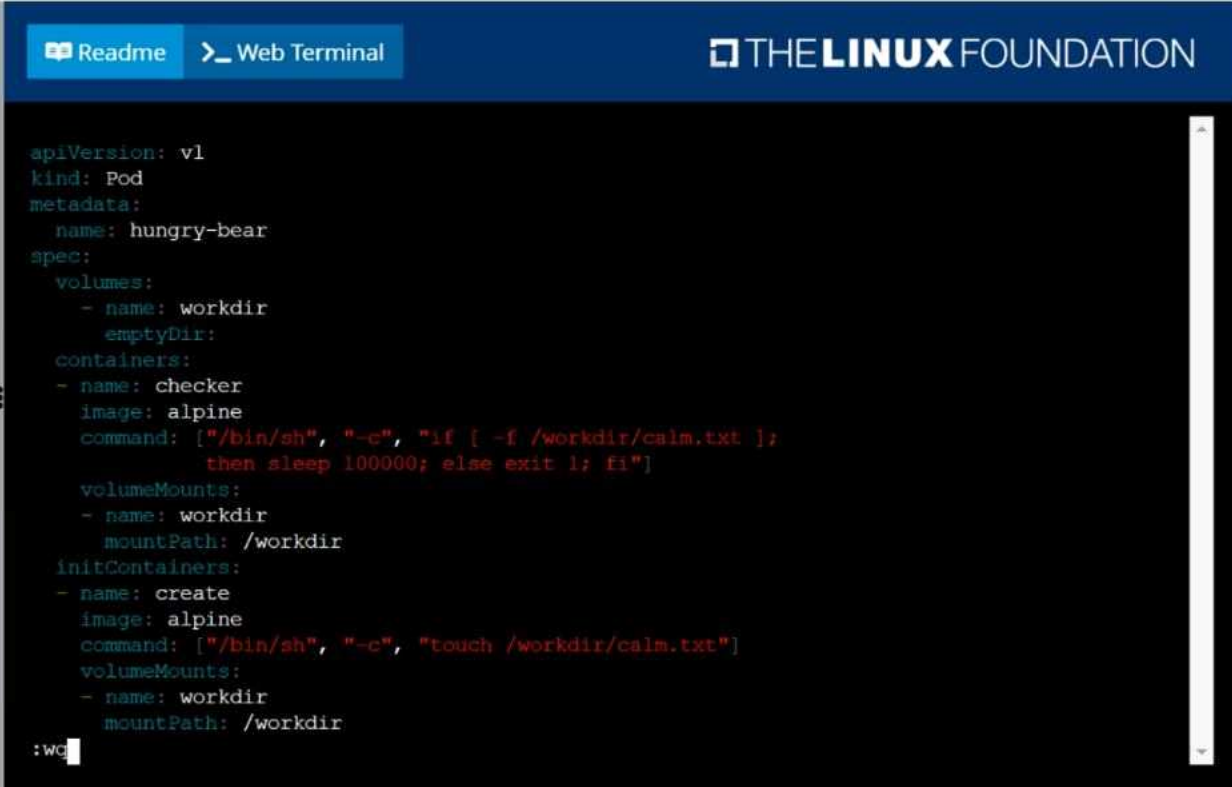


The screenshot shows a web terminal interface with a dark background and light text. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active. The terminal output shows the following commands and their results:

```
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
```

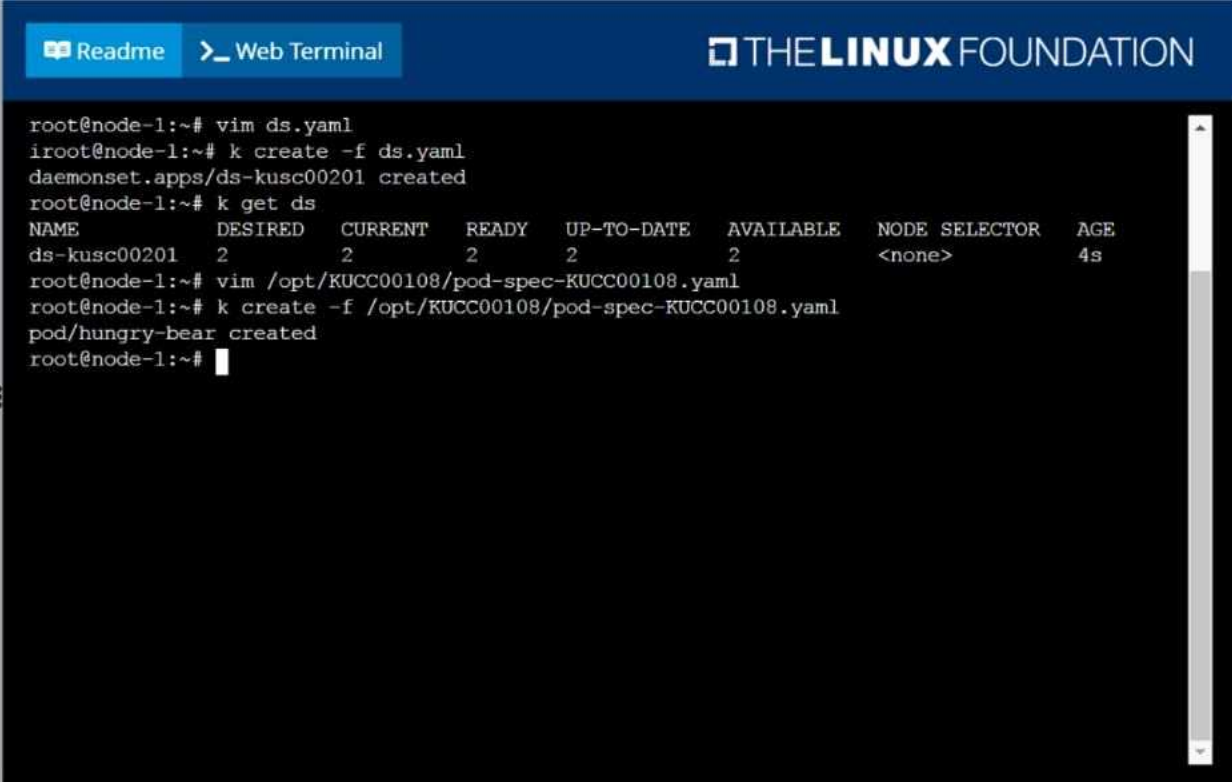
NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
ds-kusc00201	2	2	2	2	2	<none>	4s

```
root@node-1:~# 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'. To the right, the 'THE LINUX FOUNDATION' logo is displayed. The terminal content shows a YAML configuration for a 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'. To the right, the 'THE LINUX FOUNDATION' logo is displayed. The terminal content shows a series of commands and their outputs. The commands include creating a pod, checking its status, and creating another pod. The terminal prompt is 'root@node-1:~#'.

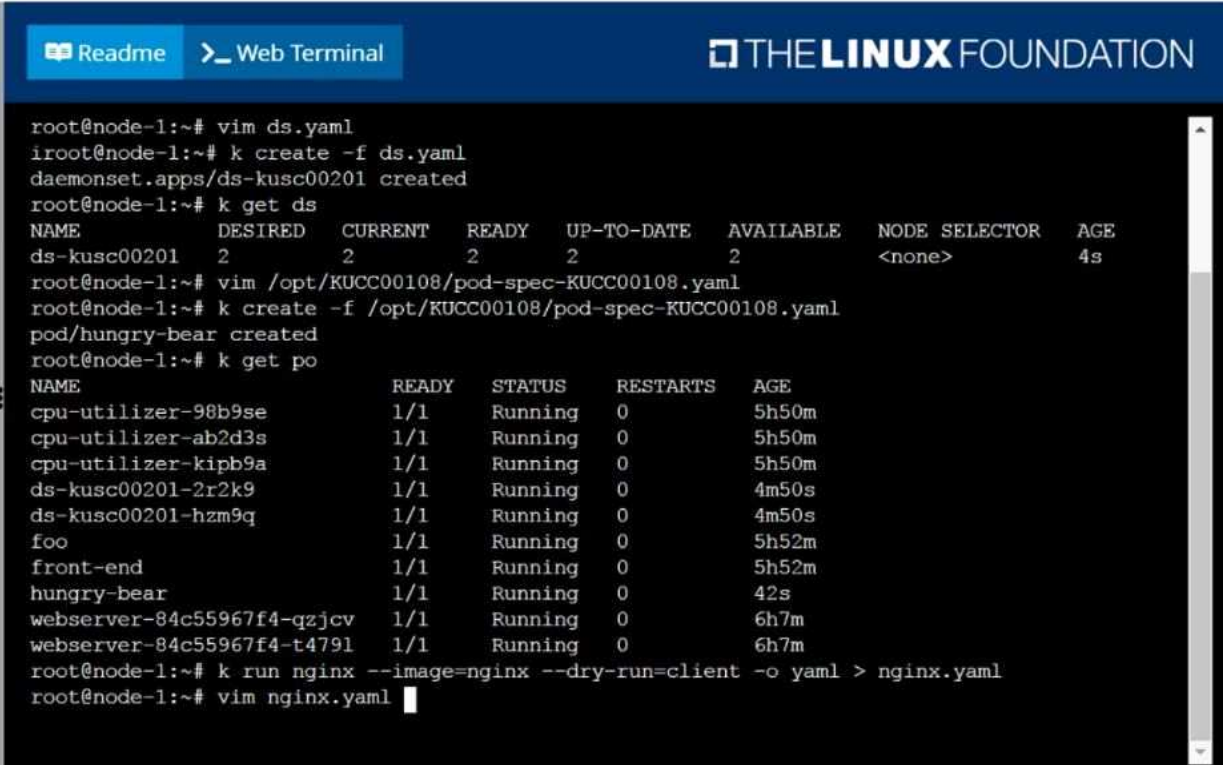
```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME          DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ds-kusc00201  2        2        2      2          2          <none>         4s
root@node-1:~# 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
```

Readme > Web Terminal
THE **LINUX** FOUNDATION

```

apiVersion: v1
kind: Pod
metadata:
  name: kucc8
spec:
  containers:
  - image: nginx
    name: nginx
  - image: redis
    name: redis
  - image: memcached
    name: memcached
~
~
~
~
~
~
~
~
~
~
~
:~

```


Readme > Web Terminal
THE **LINUX** FOUNDATION

```

cpu-utilizer-98b9se      1/1   Running   0          5h51m
cpu-utilizer-ab2d3s     1/1   Running   0          5h51m
cpu-utilizer-kipb9a    1/1   Running   0          5h51m
ds-kusc00201-2r2k9     1/1   Running   0          6m12s
ds-kusc00201-hzm9q     1/1   Running   0          6m12s
foo                     1/1   Running   0          5h54m
front-end              1/1   Running   0          5h53m
hungry-bear            1/1   Running   0          2m4s
kucc8                  0/3   ContainerCreating 0          4s
webserver-84c55967f4-qzjcv 1/1   Running   0          6h9m
webserver-84c55967f4-t4791 1/1   Running   0          6h9m
root@node-1:~# k get po
NAME                                READY    STATUS      RESTARTS   AGE
cpu-utilizer-98b9se                 1/1     Running    0           5h52m
cpu-utilizer-ab2d3s                 1/1     Running    0           5h52m
cpu-utilizer-kipb9a                 1/1     Running    0           5h52m
ds-kusc00201-2r2k9                  1/1     Running    0           6m31s
ds-kusc00201-hzm9q                  1/1     Running    0           6m31s
foo                                  1/1     Running    0           5h54m
front-end                           1/1     Running    0           5h54m
hungry-bear                          1/1     Running    0           2m23s
kucc8                                3/3     Running    0            23s
webserver-84c55967f4-qzjcv          1/1     Running    0           6h9m
webserver-84c55967f4-t4791         1/1     Running    0           6h9m
root@node-1:~#

```


Thank You for trying CKA PDF Demo

To try our CKA practice exam software visit link below

<https://prepbolt.com/CKA.html>

Start Your CKA Preparation

Use Coupon "SAVE50" for extra 50% discount on the purchase of Practice Test Software. Test your CKA preparation with actual exam questions.