

# Linux Foundation

**CKA Exam**

**Kubernetes Administrator**

## Questions & Answers (Demo Version - Limited Content)

Thank you for Downloading CKA exam PDF Demo

Get Full File:

<https://neodumps.com/cka/>

# Version: 8.0

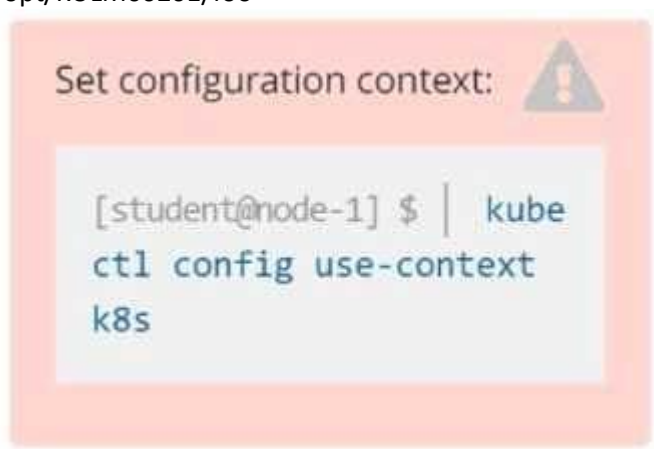
---

## Question: 1

---

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

Write \_\_\_\_\_ them \_\_\_\_\_  
/opt/KULM00201/foo

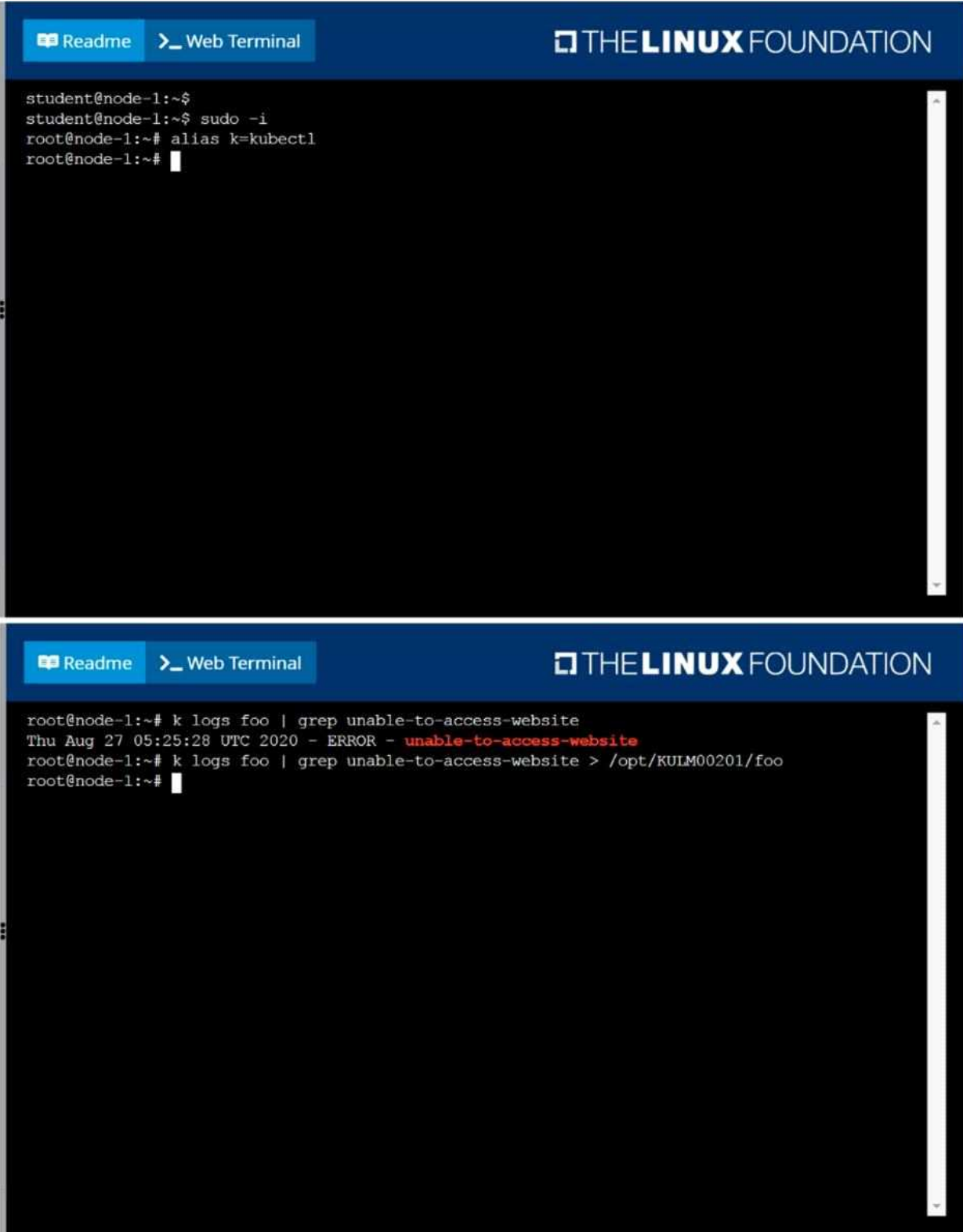


---

**Answer: See the solution below.**

---

Explanation:  
solution



The image displays two screenshots of a web terminal interface, likely from The Linux Foundation. The interface has a dark blue header with a "Readme" button and a "Web Terminal" button. The terminal shows a user logging in as root and setting an alias for kubectl.

**Top Screenshot:**

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

**Bottom Screenshot:**

```
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

---

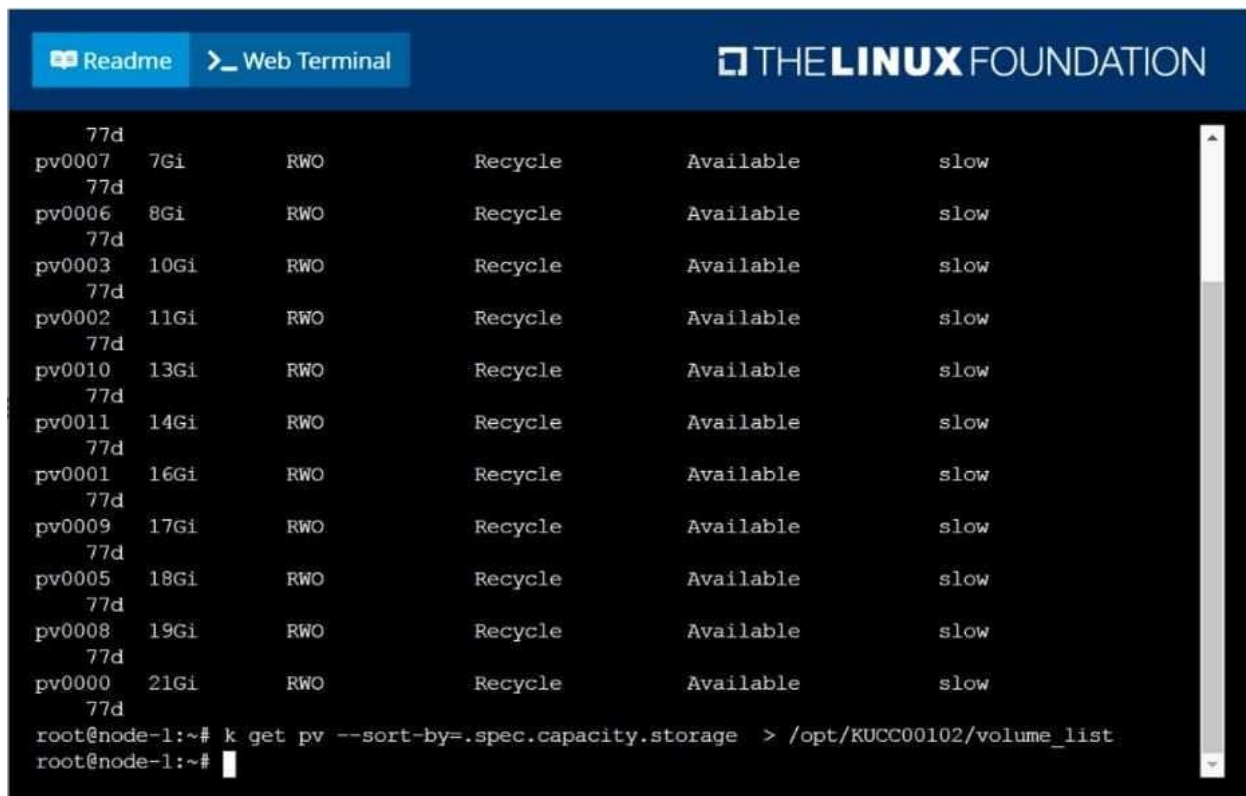
List all persistent volumes sorted by capacity, saving the full kubectl output to /opt/KUCC00102/volume\_list. Use kubectl 's own functionality for sorting the output, and do not manipulate it any further.

---

**Answer: See the solution below.**

---

Explanation:  
solution



The screenshot shows a web terminal interface 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 list of persistent volumes (pv) sorted by capacity, with columns for name, capacity, access mode, storage class, status, and storage class. The volumes are listed in descending order of capacity, from 7Gi to 21Gi. At the bottom, the command used to generate the output is shown: `k get pv --sort-by=.spec.capacity.storage > /opt/KUCC00102/volume_list`.

Name	Capacity	Access Mode	Storage Class	Status	Storage Class
pv0007	7Gi	RWO	Recycle	Available	slow
pv0006	8Gi	RWO	Recycle	Available	slow
pv0003	10Gi	RWO	Recycle	Available	slow
pv0002	11Gi	RWO	Recycle	Available	slow
pv0010	13Gi	RWO	Recycle	Available	slow
pv0011	14Gi	RWO	Recycle	Available	slow
pv0001	16Gi	RWO	Recycle	Available	slow
pv0009	17Gi	RWO	Recycle	Available	slow
pv0005	18Gi	RWO	Recycle	Available	slow
pv0008	19Gi	RWO	Recycle	Available	slow
pv0000	21Gi	RWO	Recycle	Available	slow

```
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 screenshot shows a web terminal interface with a blue header bar containing 'Readme' and 'Web Terminal' tabs, and 'THE LINUX FOUNDATION' logo. The terminal window shows a user at 'root@node-1:~#' editing 'ds.yaml' with 'vim'. The content of 'ds.yaml' is a Kubernetes DaemonSet configuration for 'fluentd-elasticsearch' in the 'kube-system' namespace. The configuration includes labels, a selector, a template with a container named 'nginx', and a toleration for master nodes. The terminal shows the file being edited in real-time, with the cursor at the end of the 'tolerations' list. The bottom of the terminal shows the file name 'ds.yaml' and the line numbers '17,19' and 'All'.

```
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
~
~
~
~
~
~
~
~
:wc
```

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

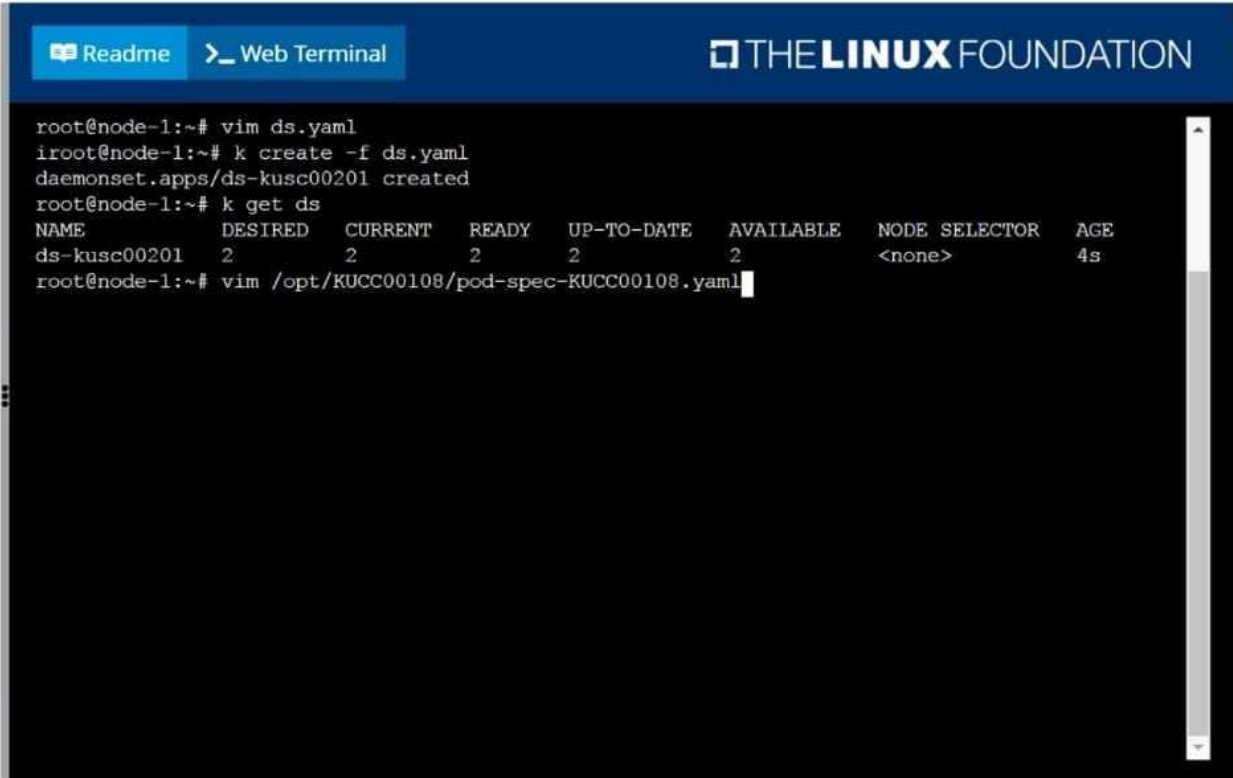
Once the spec file has been updated with the init container definition, the pod should be created

---

**Answer: See the  
solution below.**

---

Explanation:  
solution



The screenshot shows a web terminal interface with a dark background. 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 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
```

Readme

Web Terminal

THE **LINUX** FOUNDATION

```
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
:wc
```

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:~# 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**

---

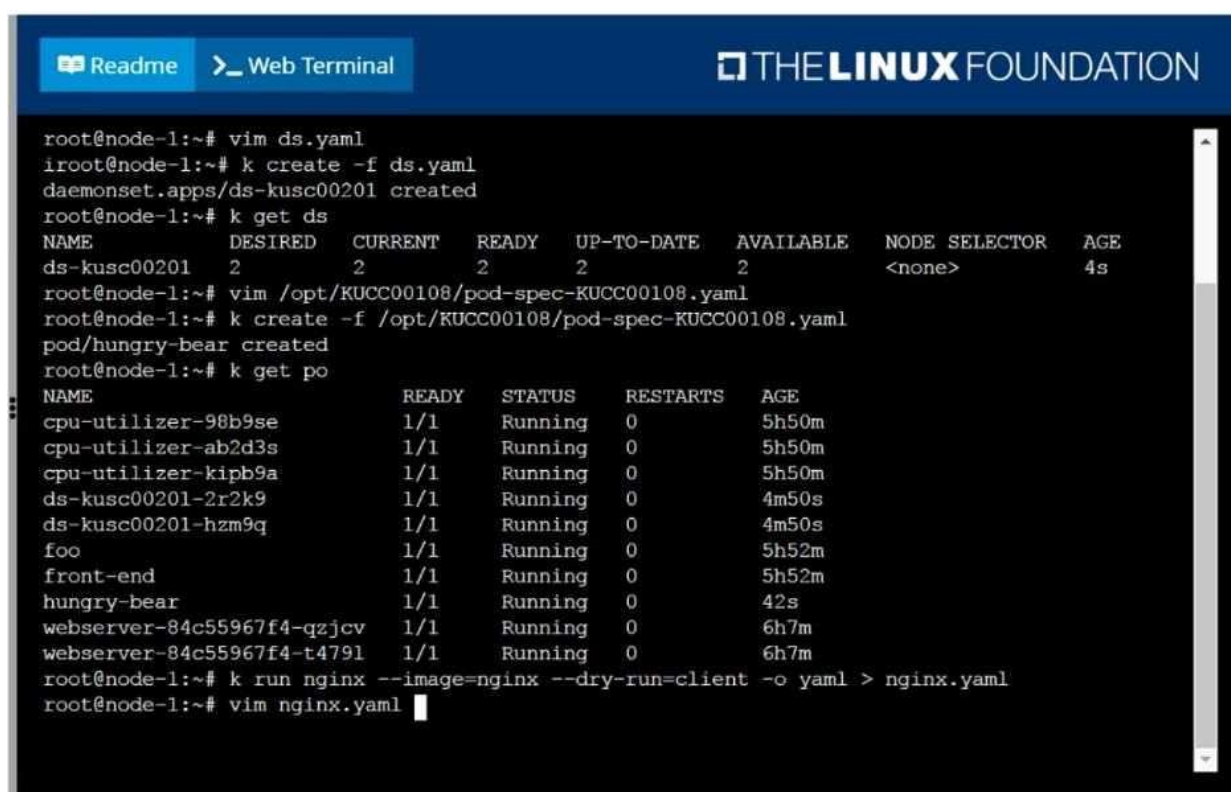
Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified):  
nginx + redis + memcached.

---

**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 two tabs: 'Readme' and 'Web Terminal'. The title bar of the terminal window says 'THE LINUX FOUNDATION'. The terminal output shows the following commands and their 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**

<https://neodumps.com/cka/>

**Start Your CKA Preparation**

*[Limited Time Offer]* Use Coupon "**SAVE20**" for extra 20% discount on the purchase of PDF file. Test your CKA preparation with actual exam questions

[www.neodumps.com](http://www.neodumps.com)