EXAMS TEST

# Free Questions for CKA by vceexamstest

## Shared by Kramer on 07-06-2022

For More Free Questions and Preparation Resources

Check the Links on Last Page

# Question 1

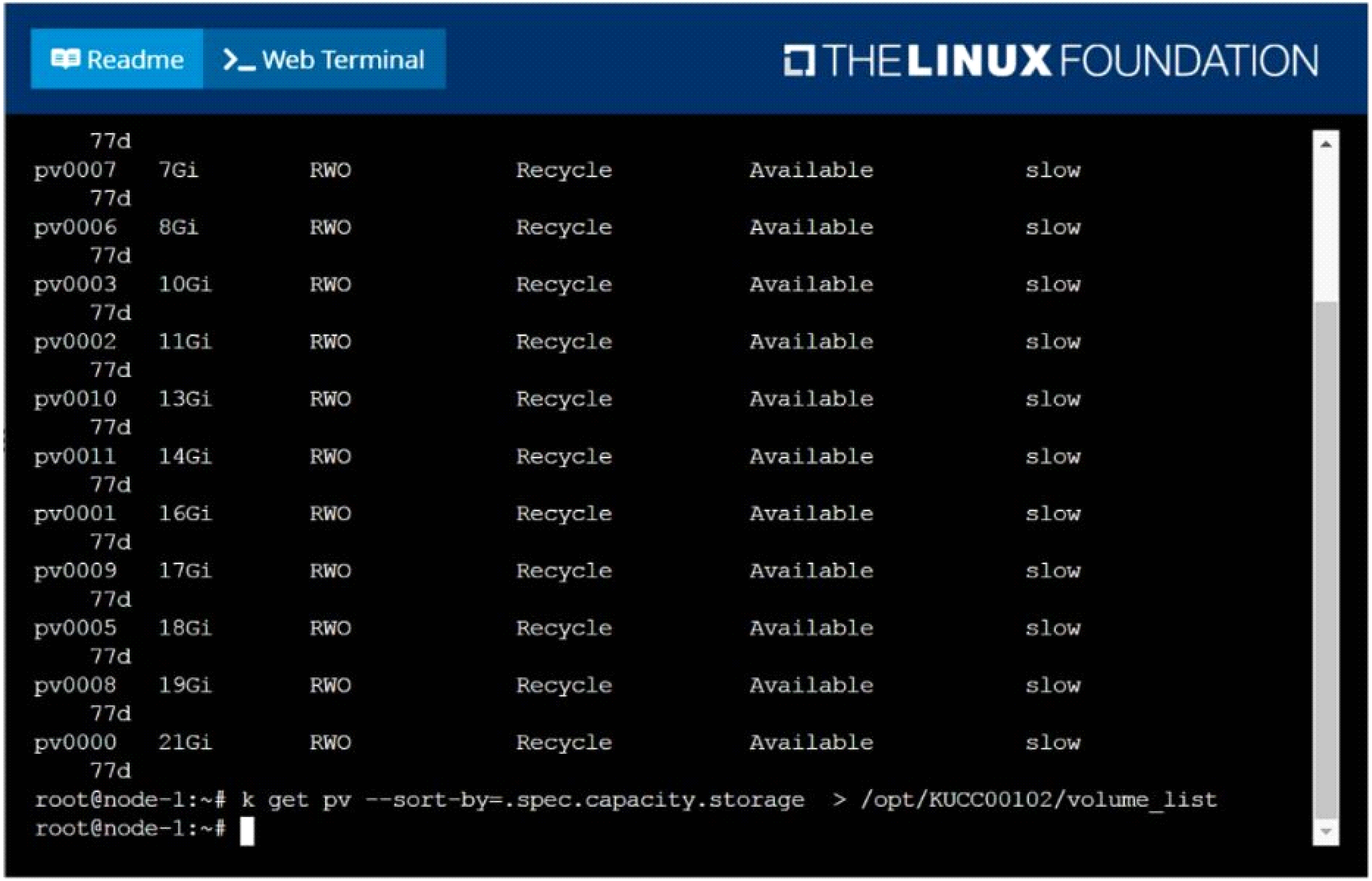
**Question Type: MultipleChoice**

SIMULATION

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.

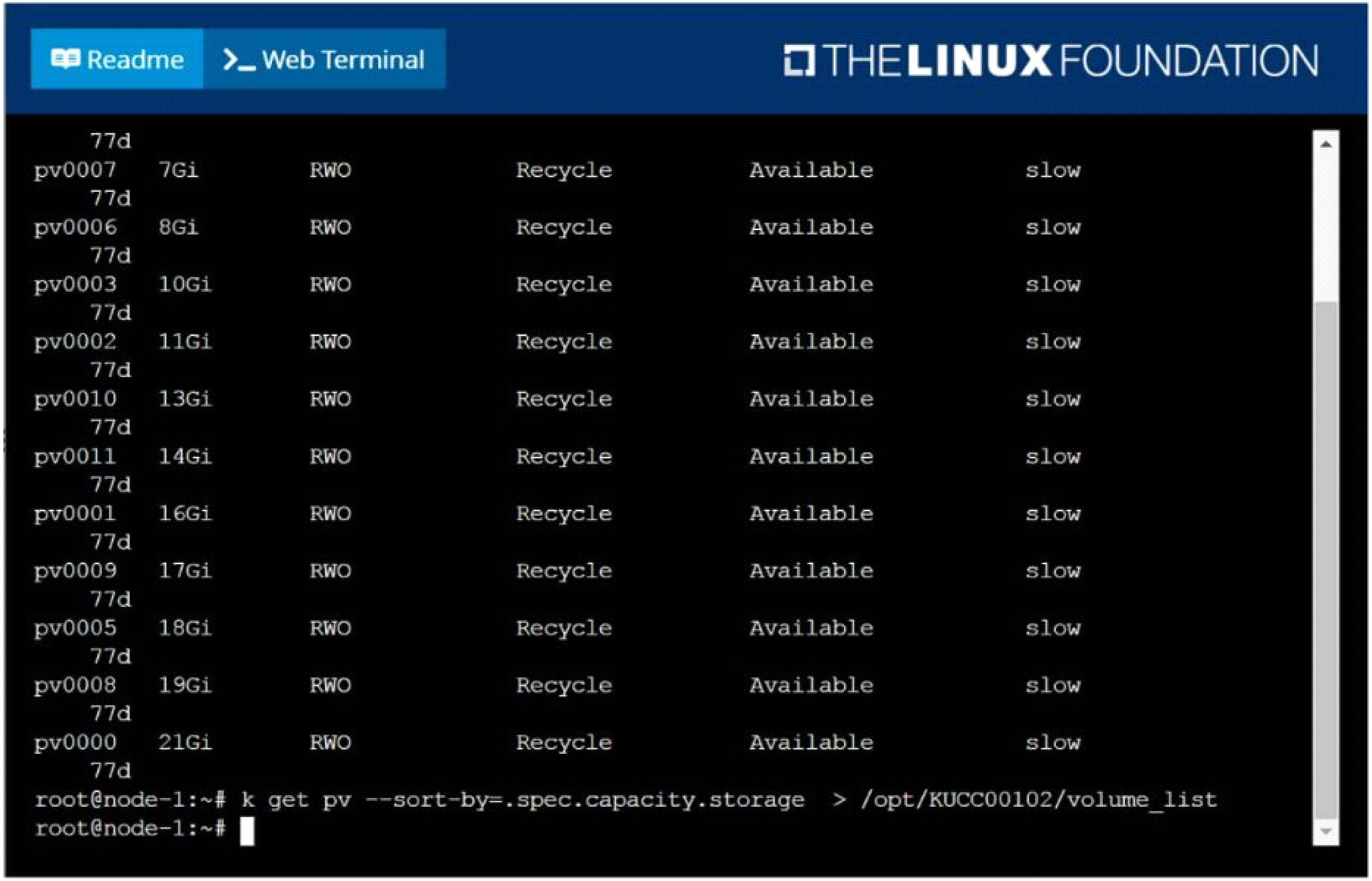
## Options:

**A)** solution



```
    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:~# 
```

**B)** solution



```
    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:~# 
```

### Answer:

B

## Question 2

**Question Type:** **MultipleChoice**

SIMULATION

Monitor the logs of pod foo and:

* Extract log lines corresponding to error

unable-to-access-website

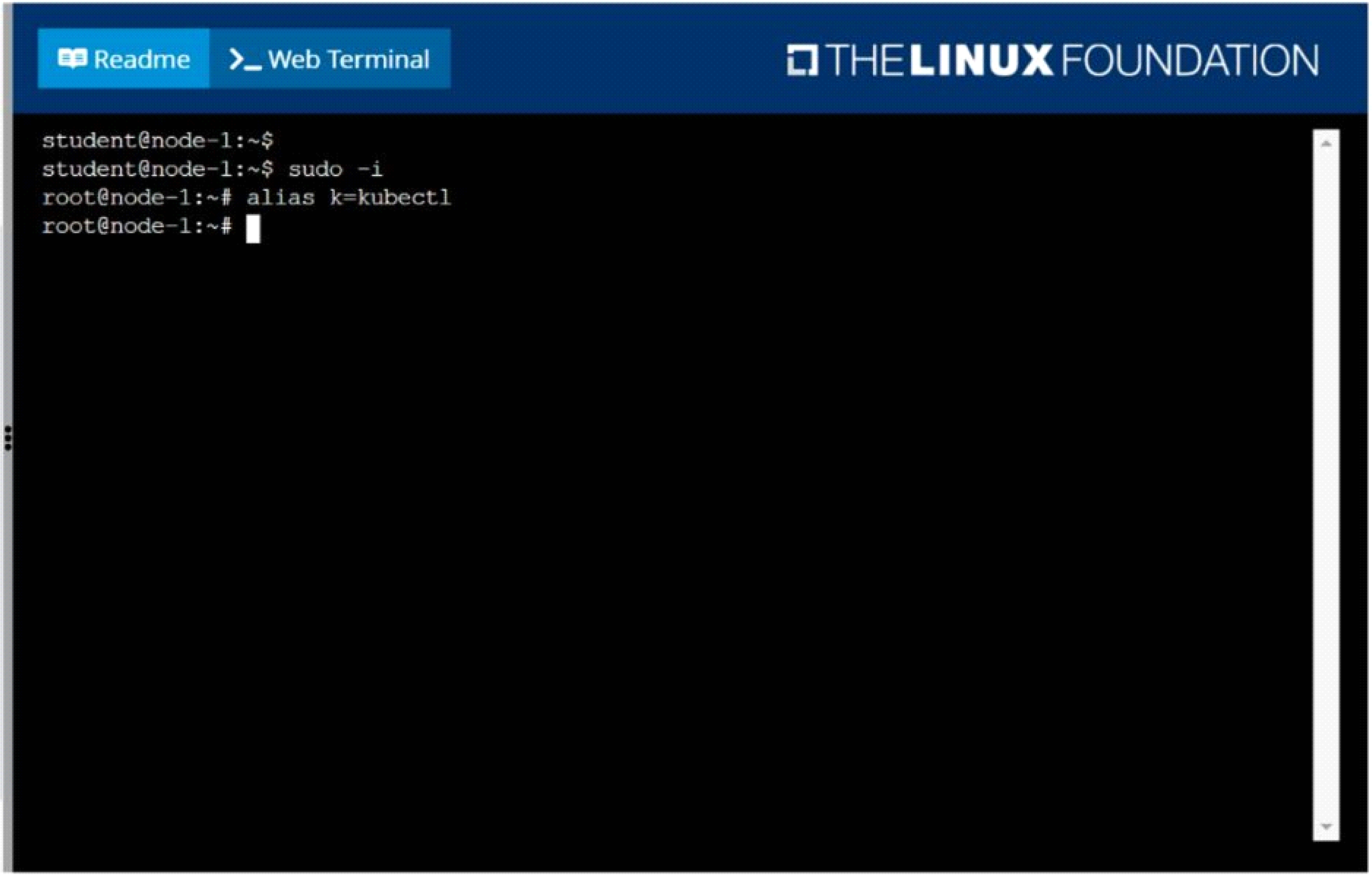
* Write them to

/opt/KULM00201/foo

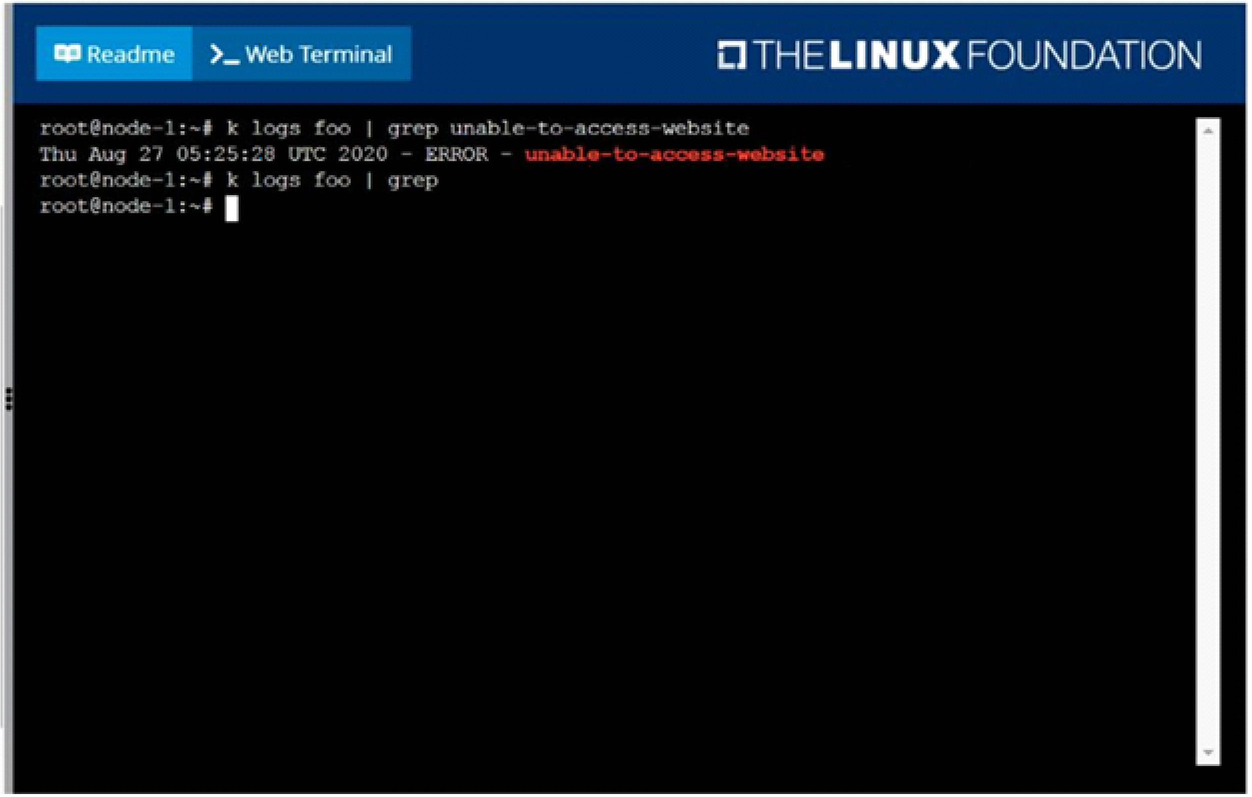
Image not found or type unknown

## Options:

**A)** solution

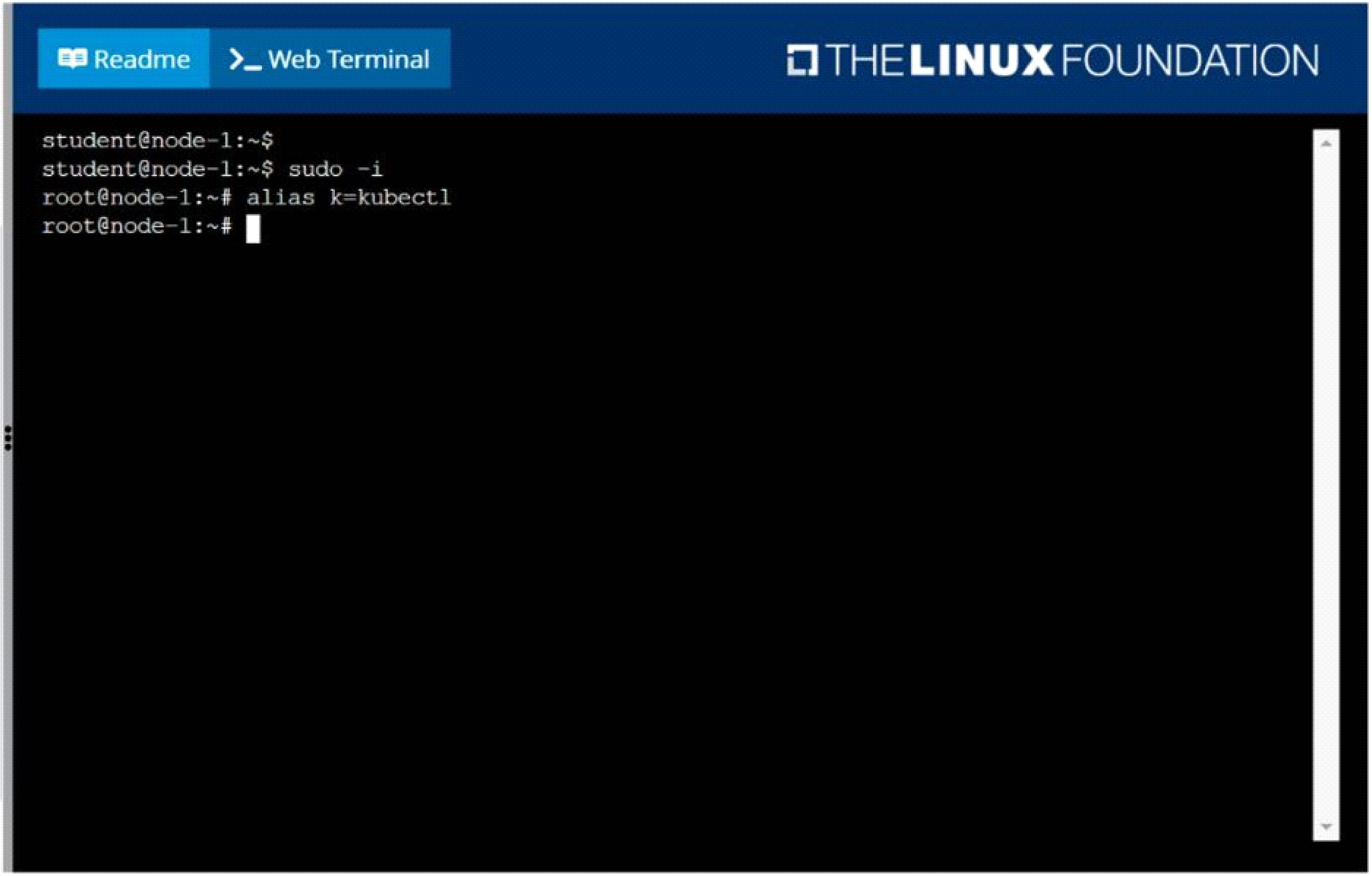


```
student@node-1:~$
student@node-1:~$ sudo -i
root@node-1:~# alias k=kubectl
root@node-1:~#
```

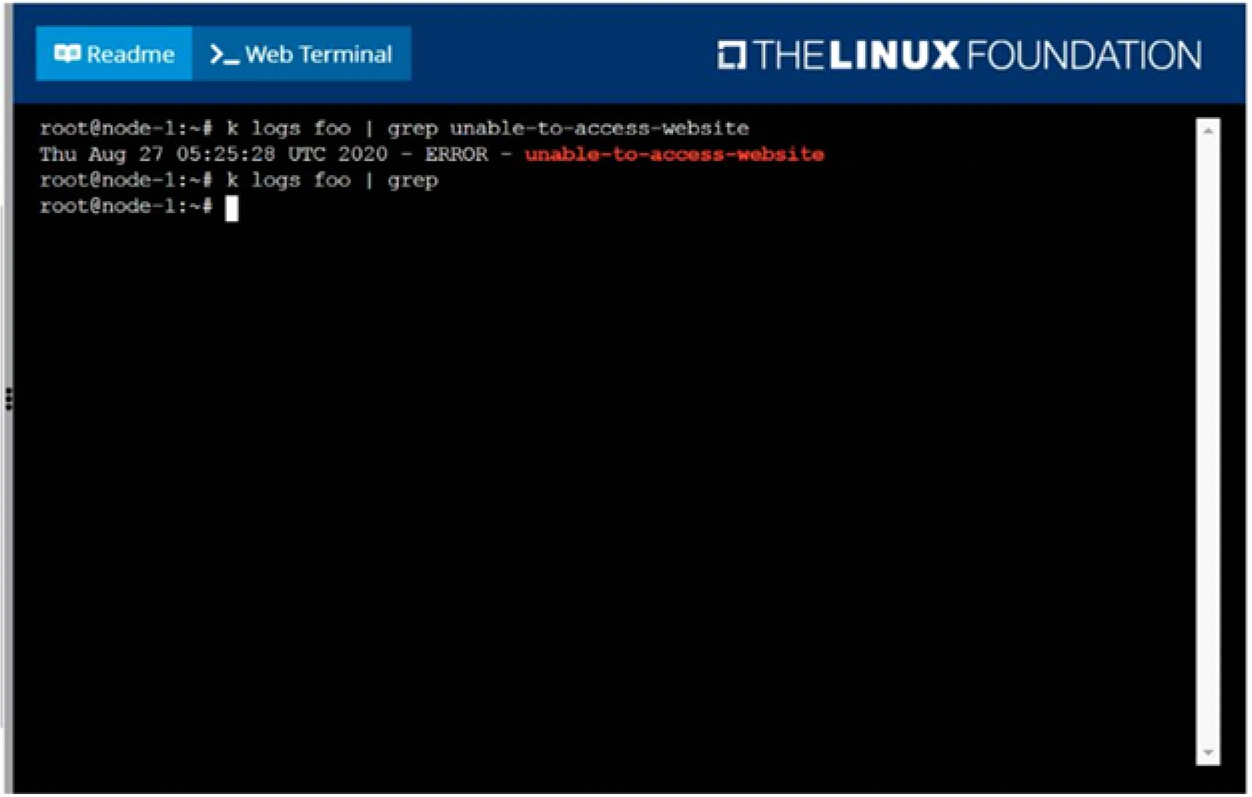


```
root@node-1:~# k logs foo | grep unable-to-access-website
Thu Aug 27 05:25:28 UTC 2020 - ERROR - unable-to-access-website
root@node-1:~# k logs foo | grep
root@node-1:~#
```

B) solution



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

## Answer:

A

# Question 3

**Question Type:** **MultipleChoice**

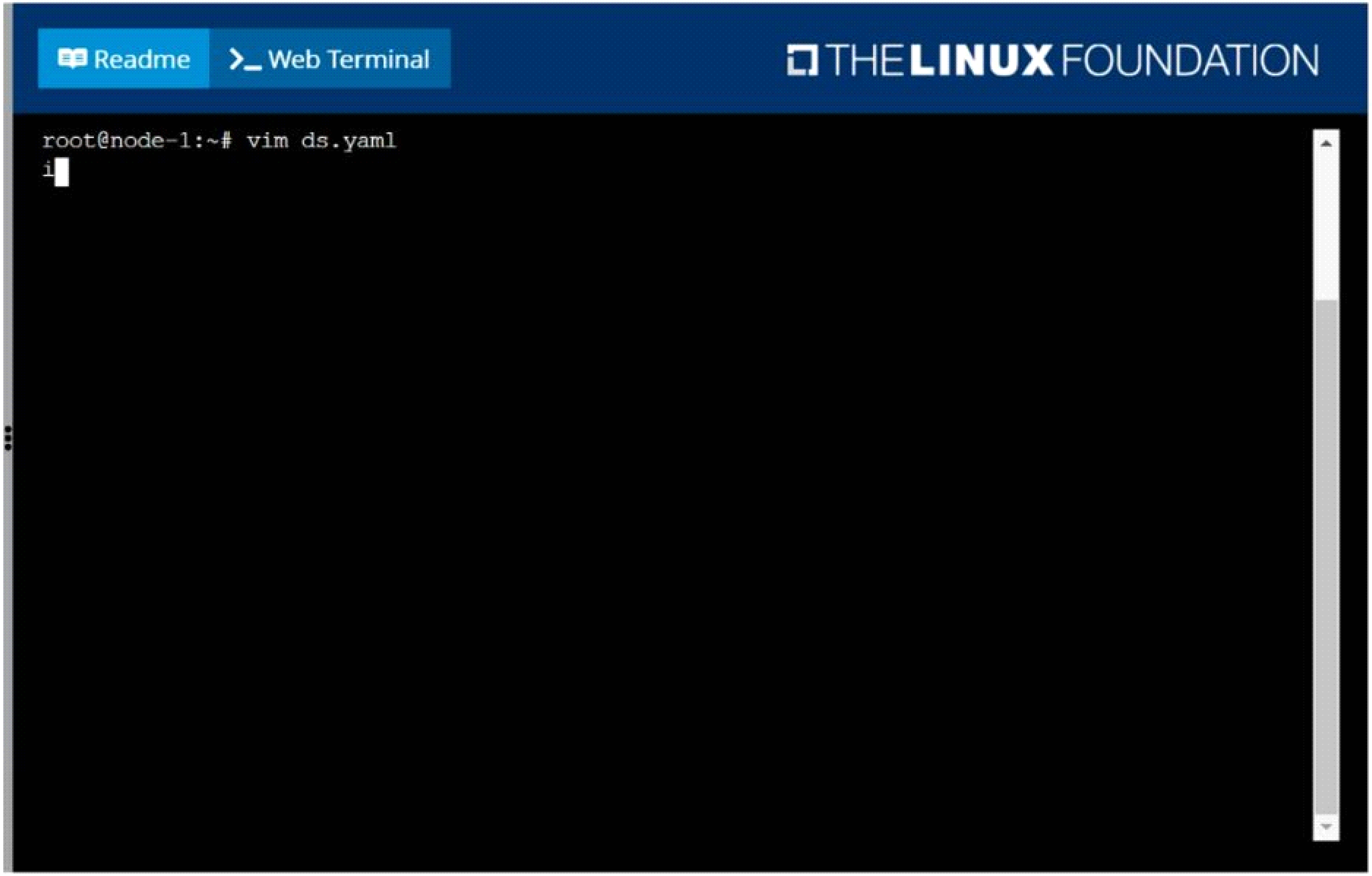
SIMULATION

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.

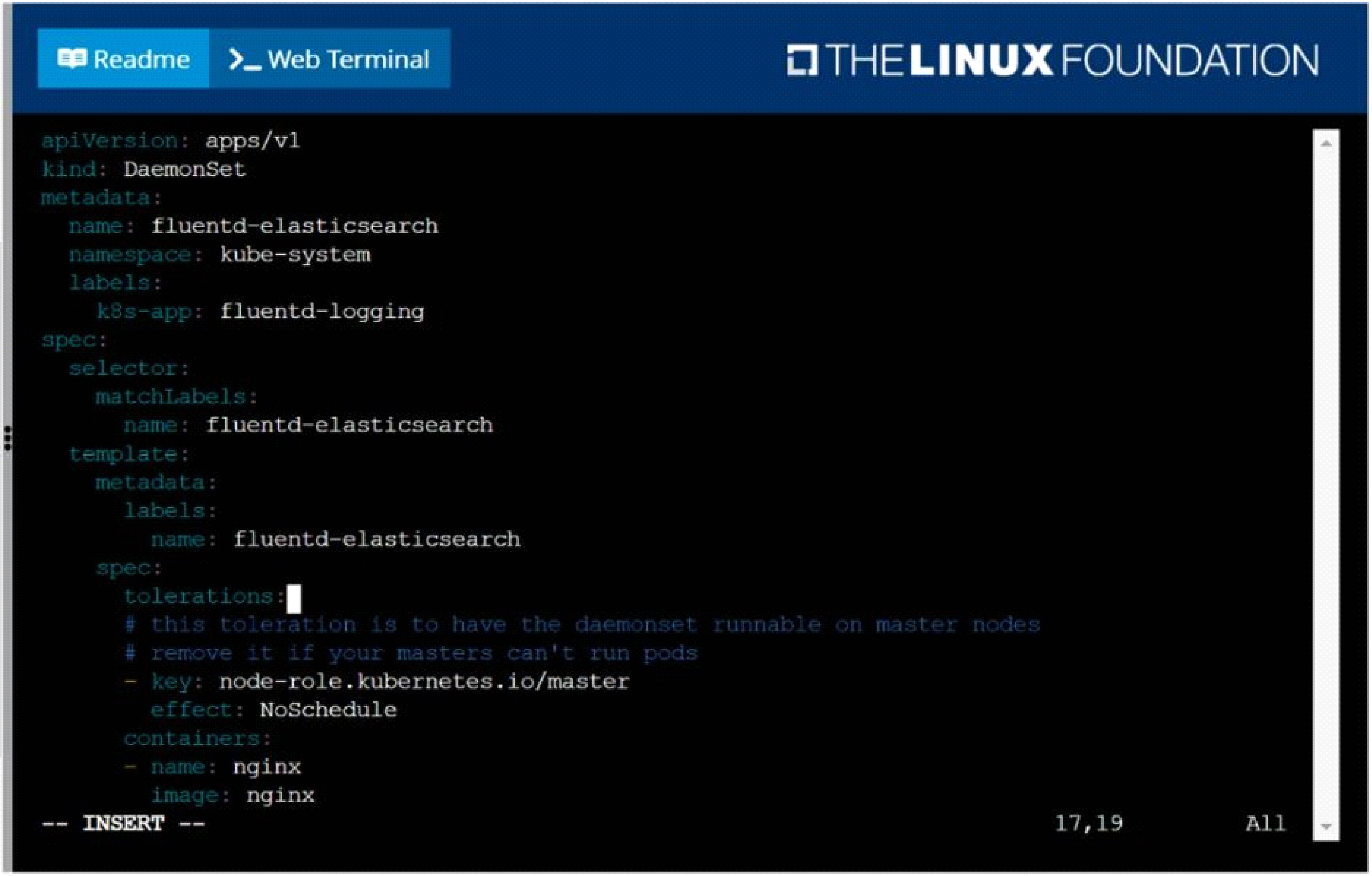
Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

## Options:

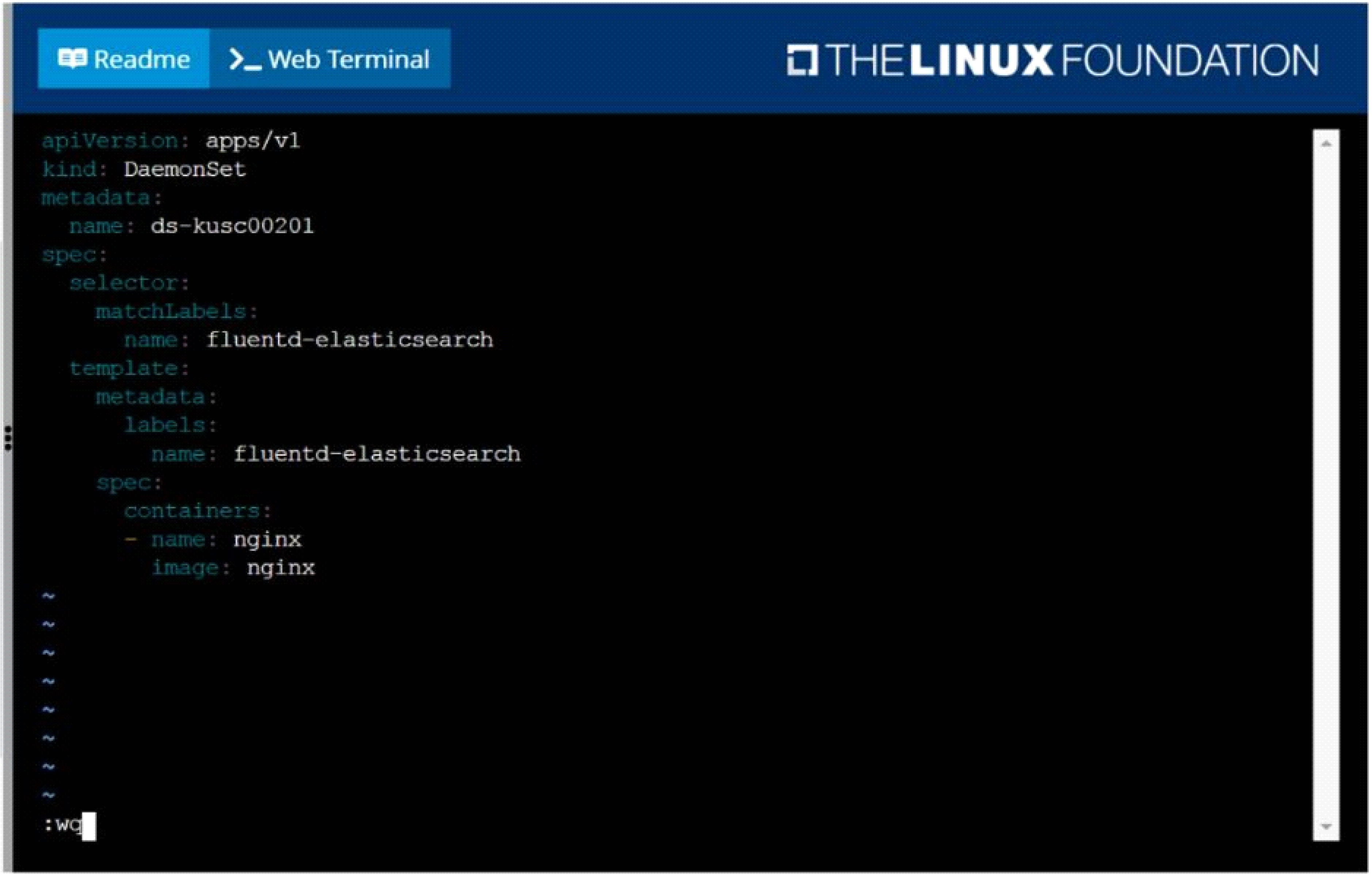
**A)** solution



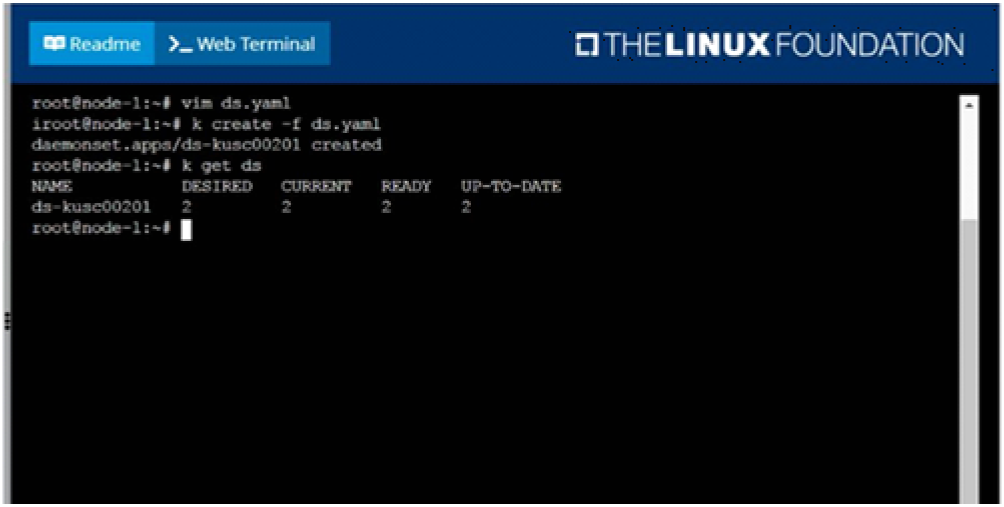
```
root@node-1:~# vim ds.yaml
i
```



```
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 --                                                          17,19         All
```

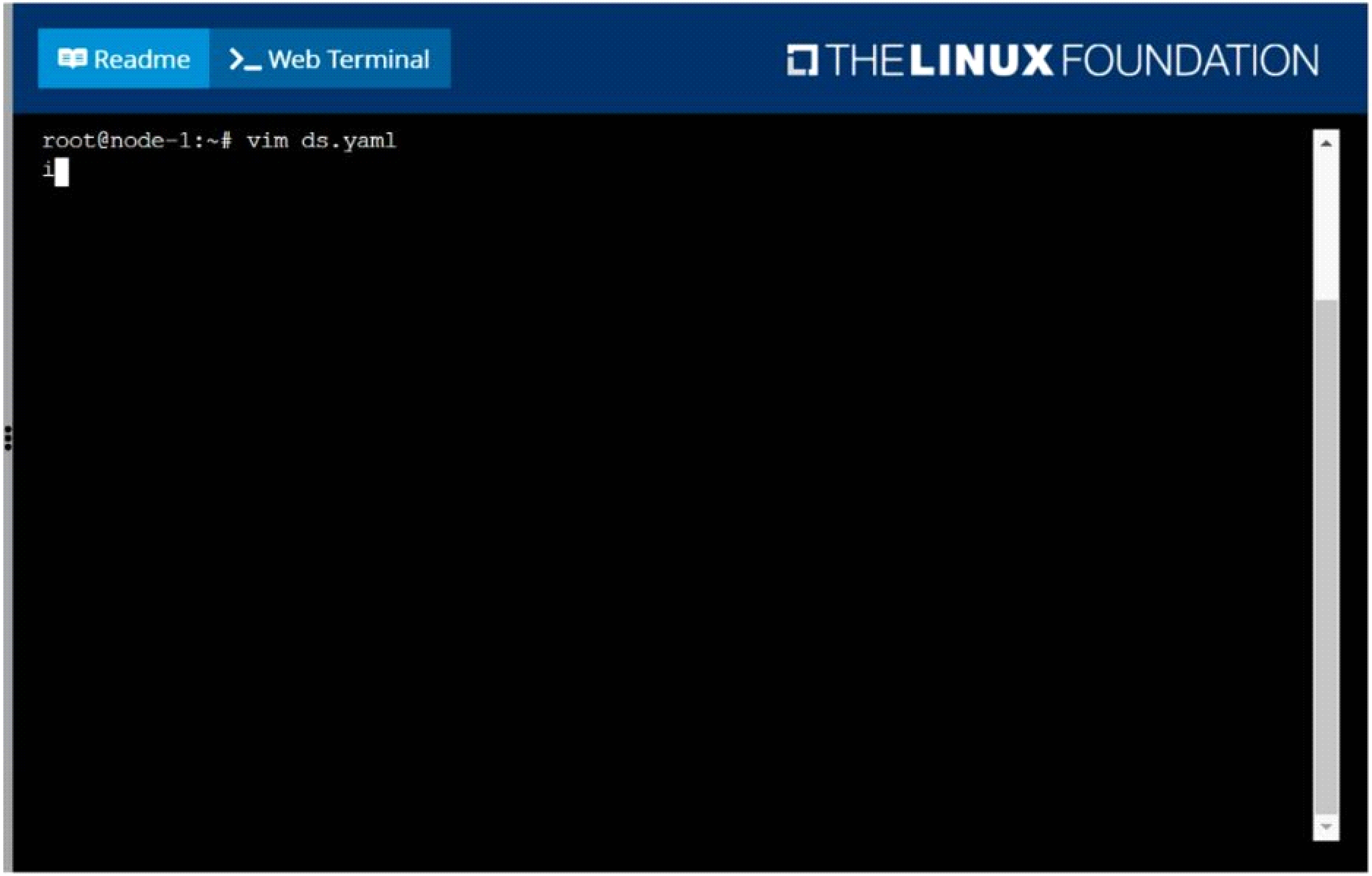


```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: ds-kusc00201
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
      - name: nginx
        image: nginx
~
~
~
~
~
~
~
~
:wq
```

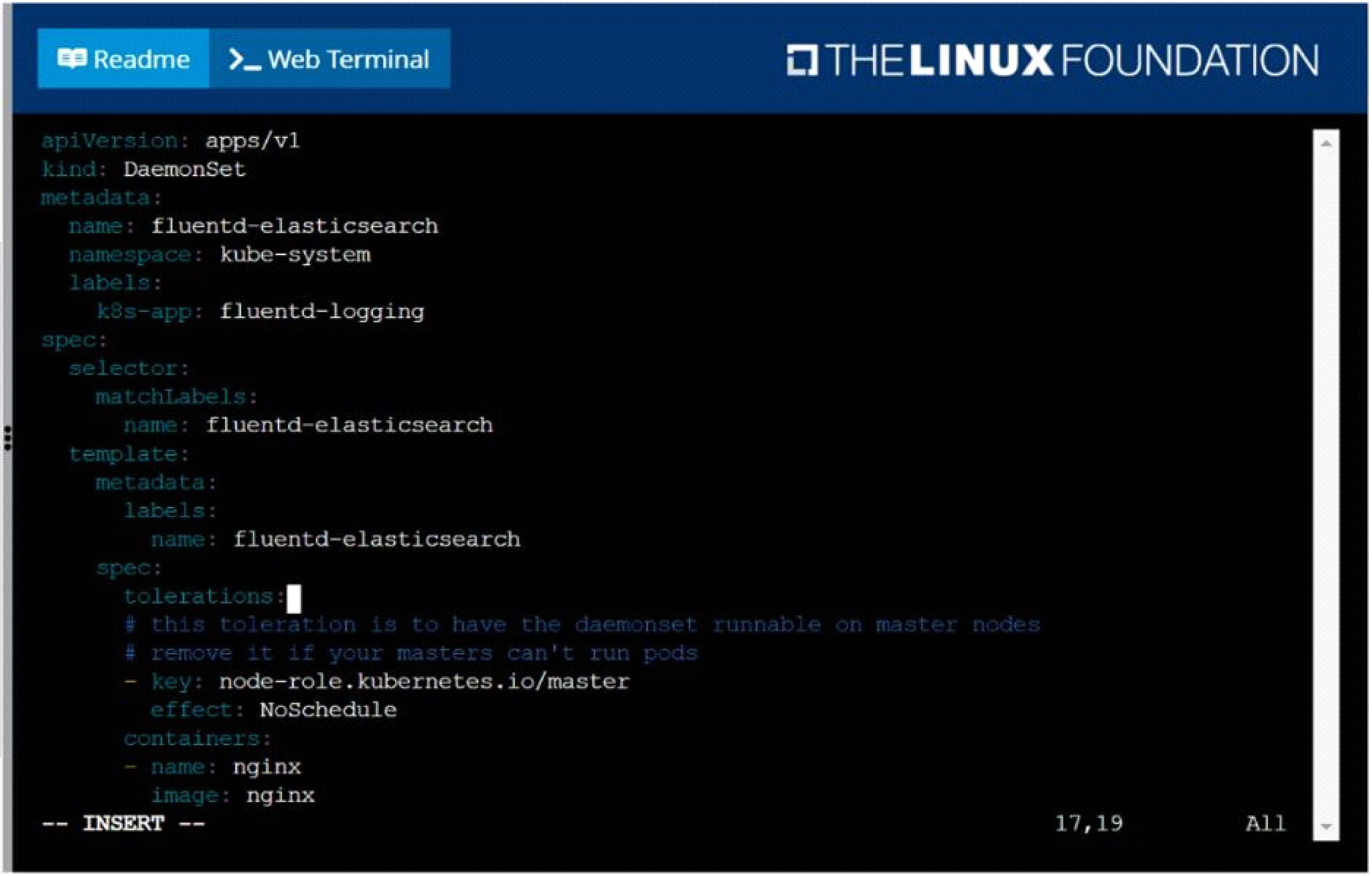


```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED   CURRENT   READY   UP-TO-DATE
ds-kusc00201   2         2         2       2
root@node-1:~# 
```

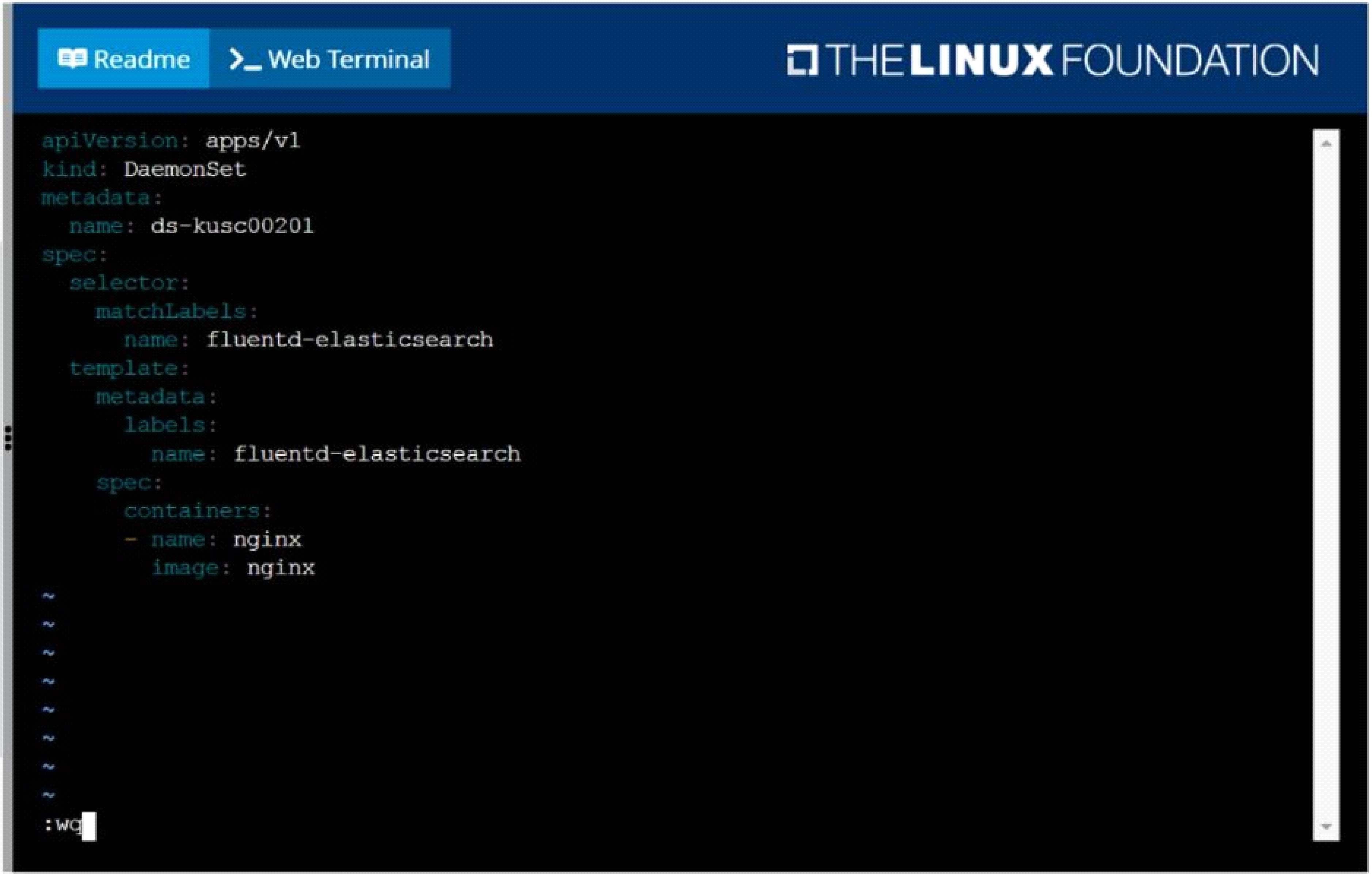
B) solution



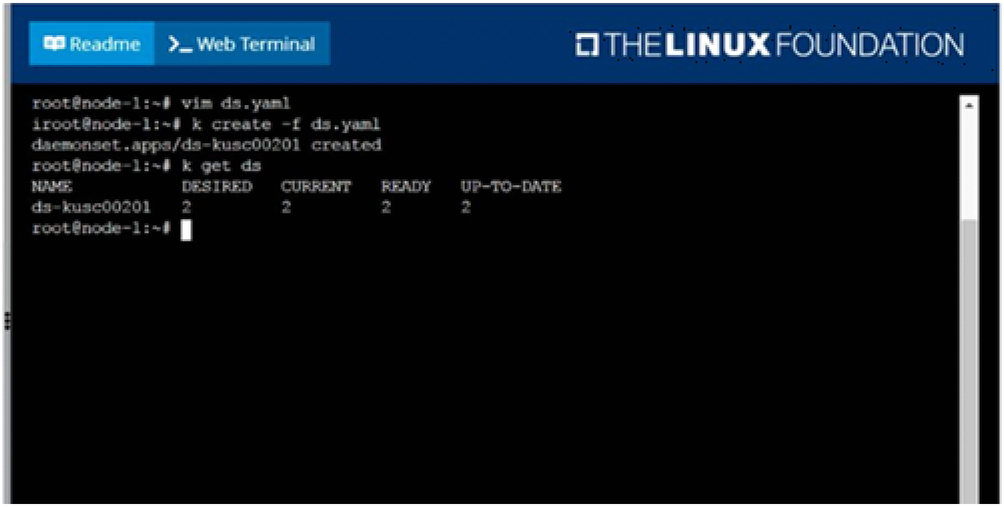
```
root@node-1:~# vim ds.yaml
i
```



```
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 --                                                                  17,19         All
```



```
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
~
~
~
~
~
~
~
~
:wq
```



```
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
ds-kusc00201   2         2         2       2
root@node-1:~# 
```

## Answer:

A

# Question 4

**Question Type:** **MultipleChoice**

SIMULATION

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

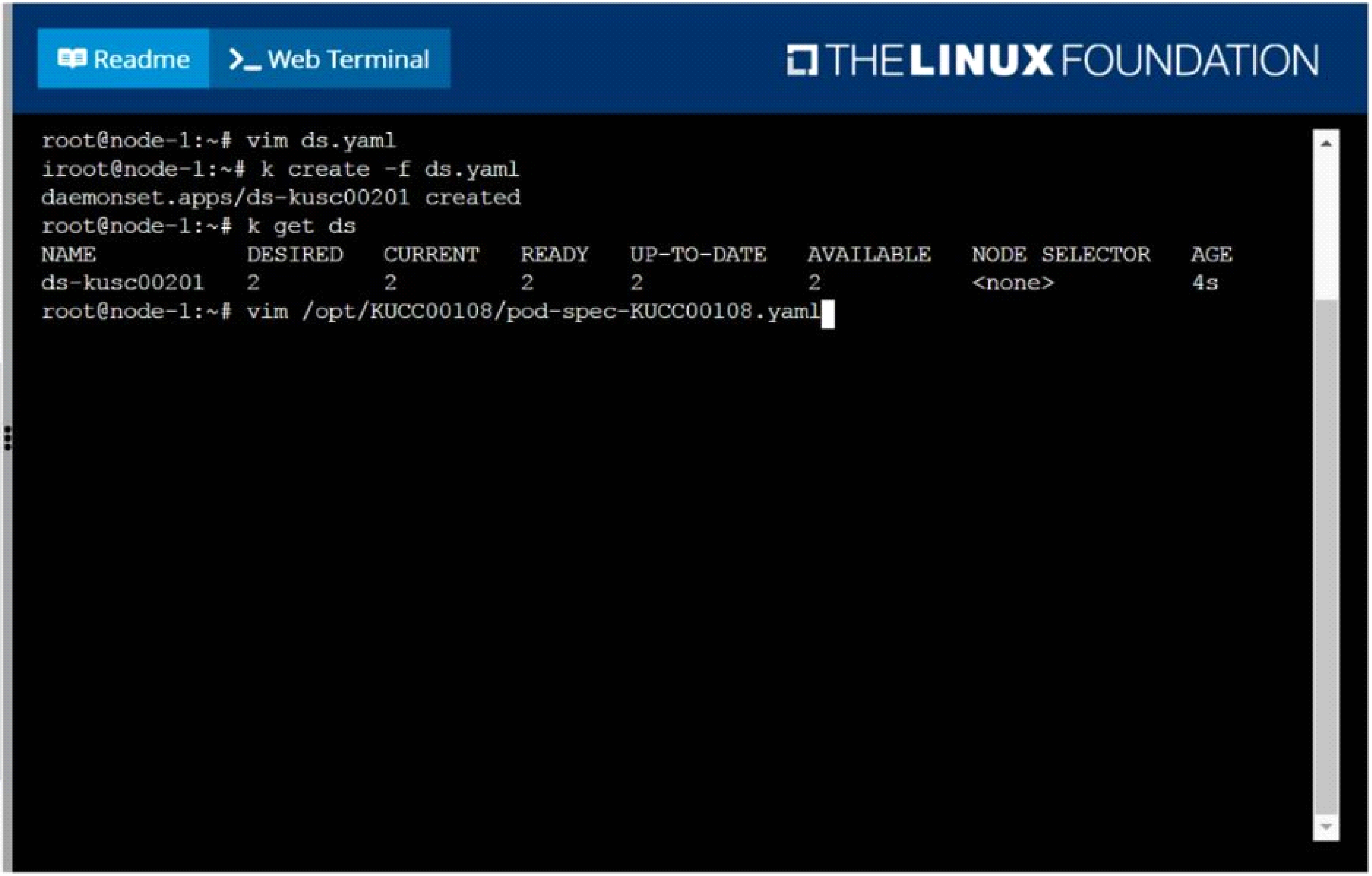
/workdir/calm.txt

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

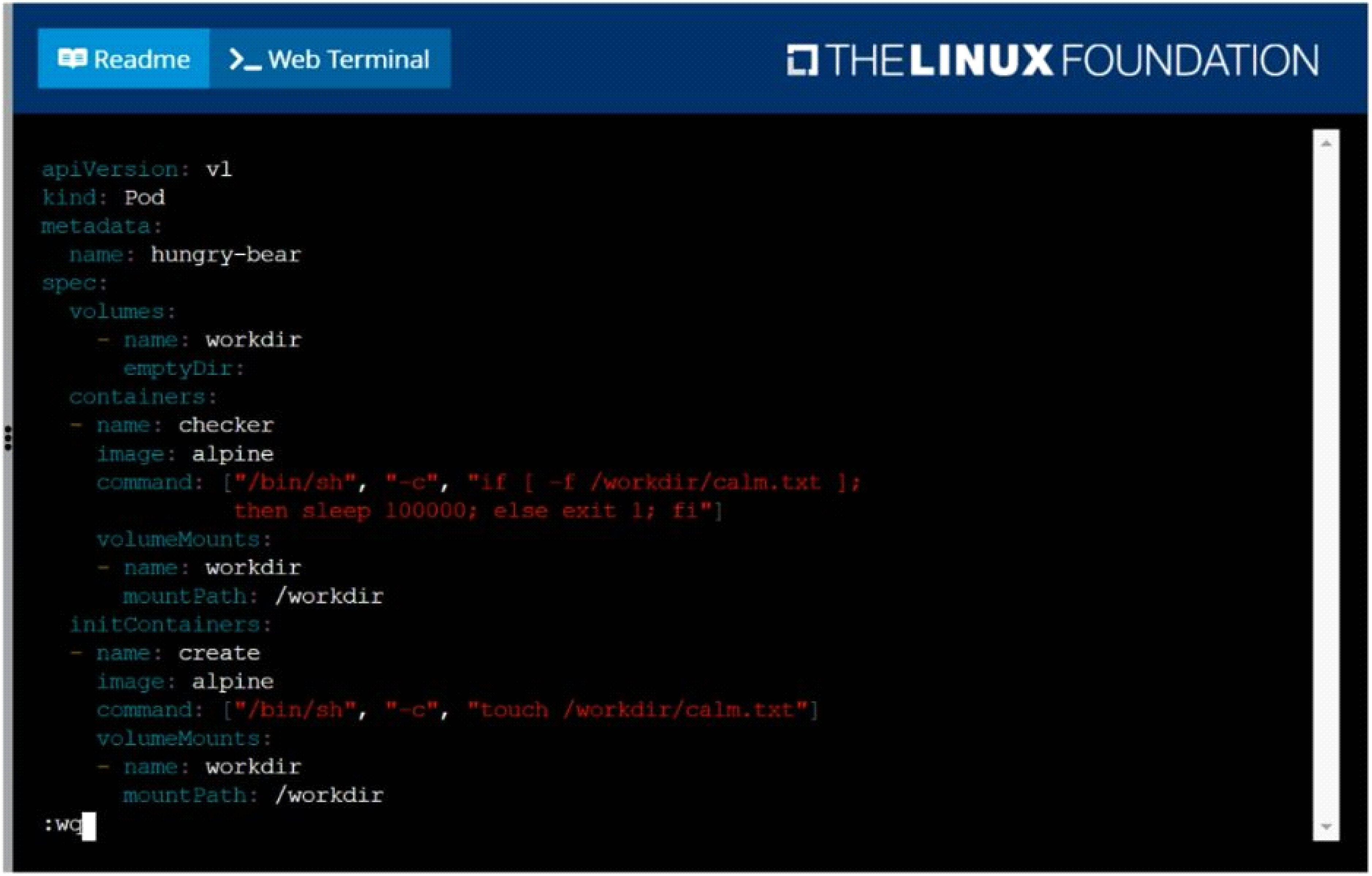
* Once the spec file has been updated with the init container definition, the pod should be created

## Options:

**A)** solution

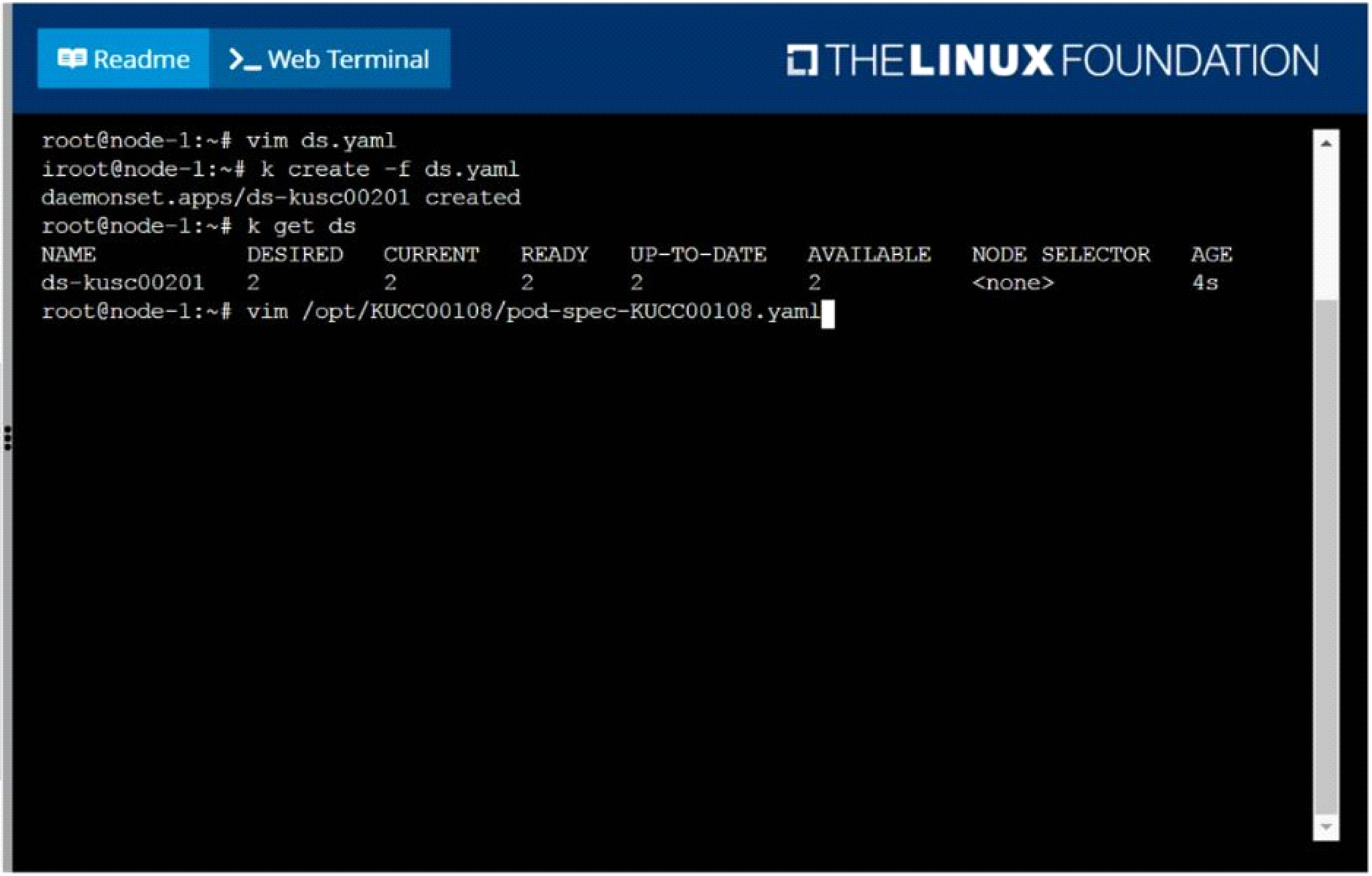


```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME            DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201    2         2         2       2            2           <none>          4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
```

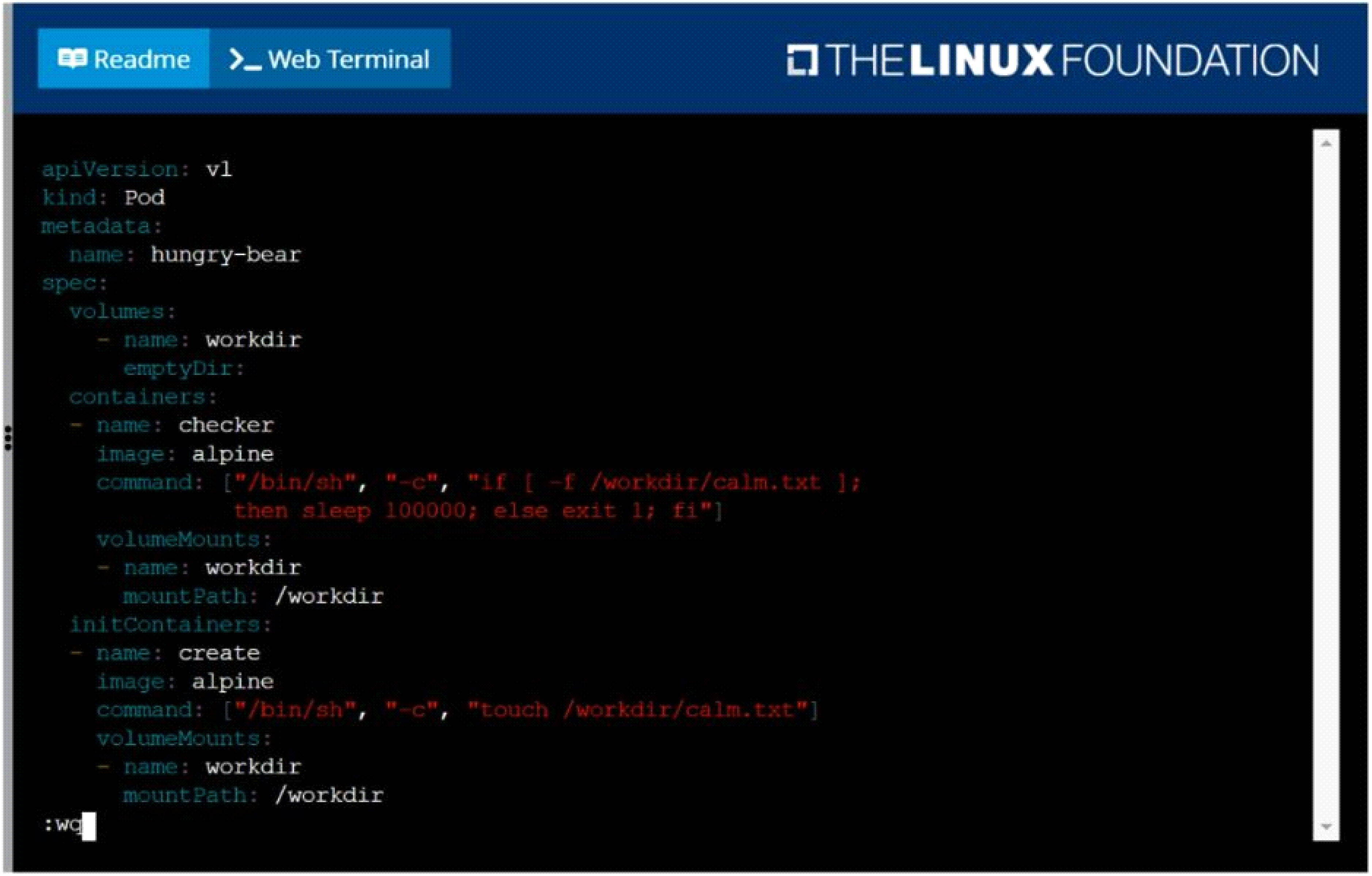


```
apiVersion: v1
kind: Pod
metadata:
  name: hungry-bear
spec:
  volumes:
    - name: workdir
      emptyDir:
  containers:
  - name: checker
    image: alpine
    command: ["/bin/sh", "-c", "if [ -f /workdir/calm.txt ];
              then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
  initContainers:
  - name: create
    image: alpine
    command: ["/bin/sh", "-c", "touch /workdir/calm.txt"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
:wq
```

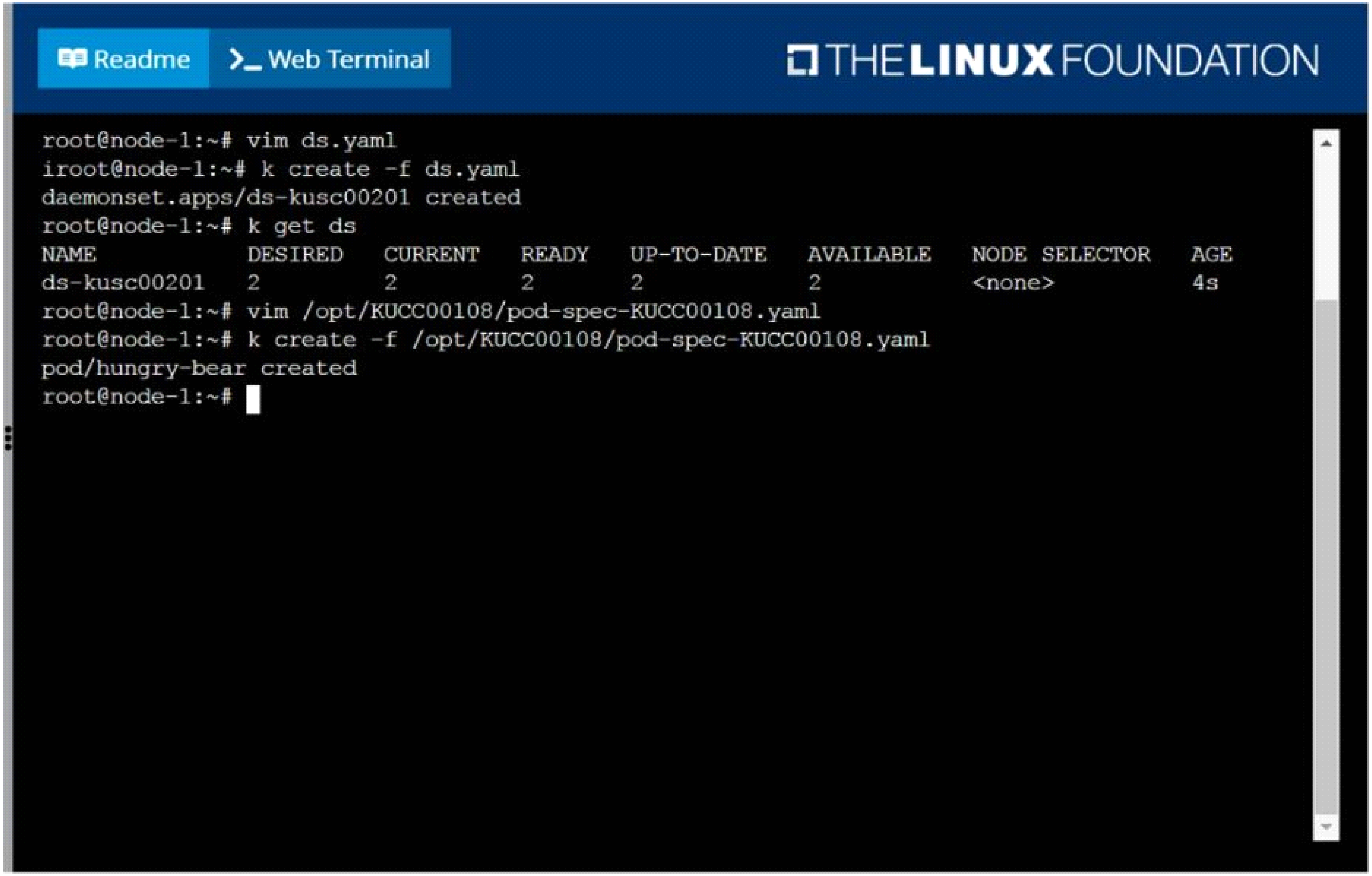
**B)** solution



```
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
```



```
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
```



```
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:~#
```

### Answer:

B

# Question 5

**Question Type:** **MultipleChoice**

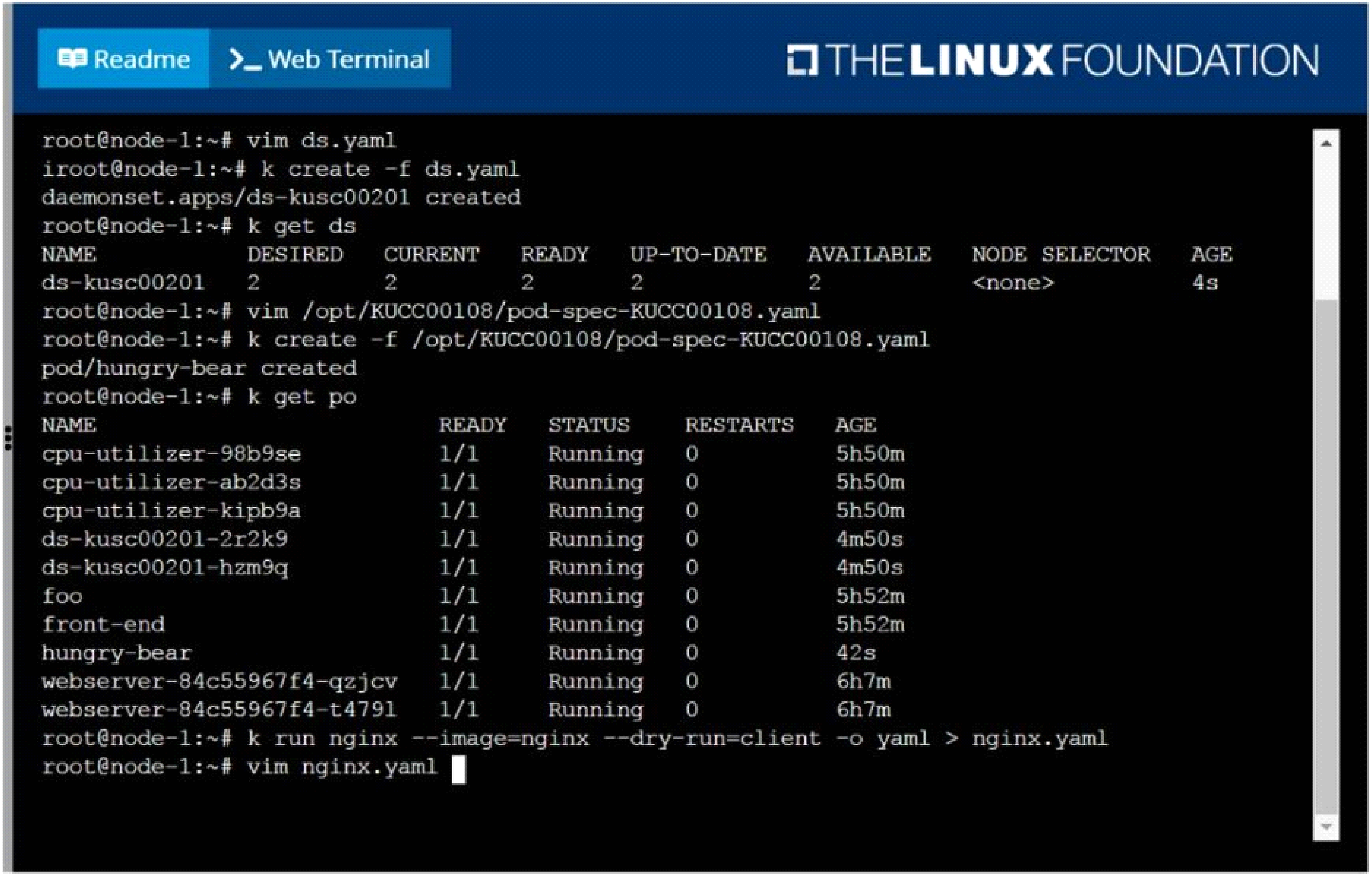
SIMULATION

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):

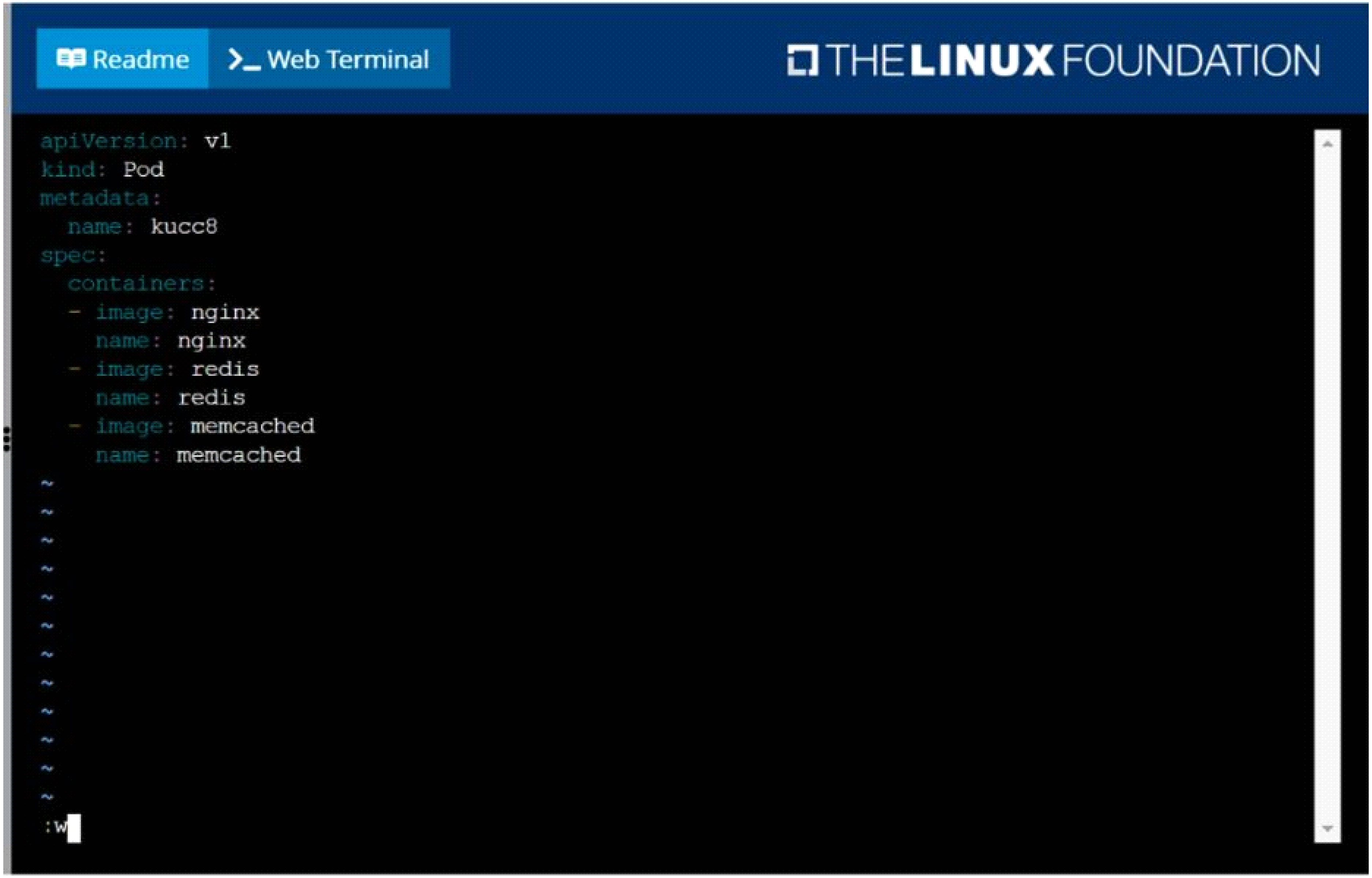
nginx + redis + memcached.

### Options:

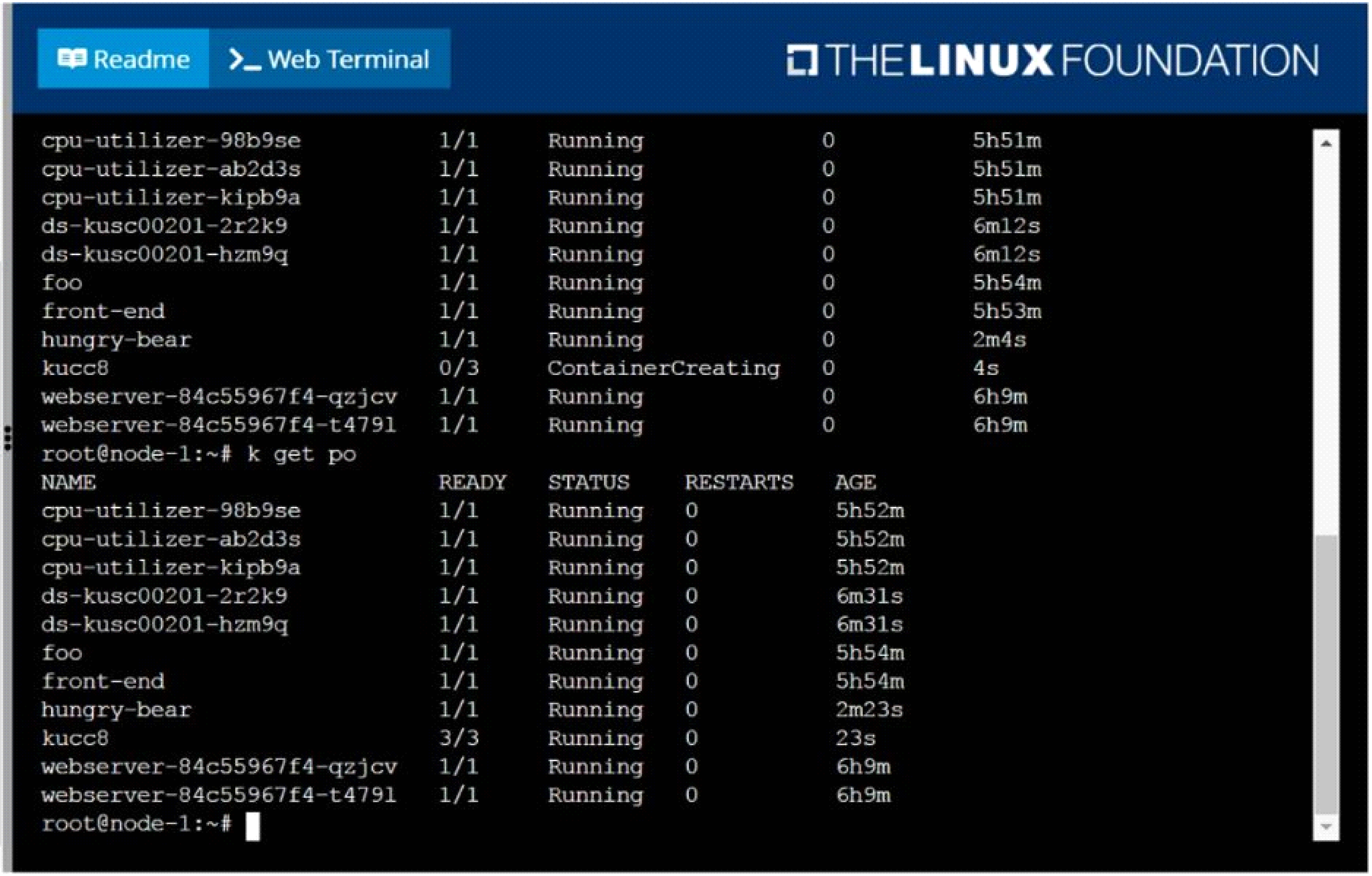
**A)** solution



```
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-t4791     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
```

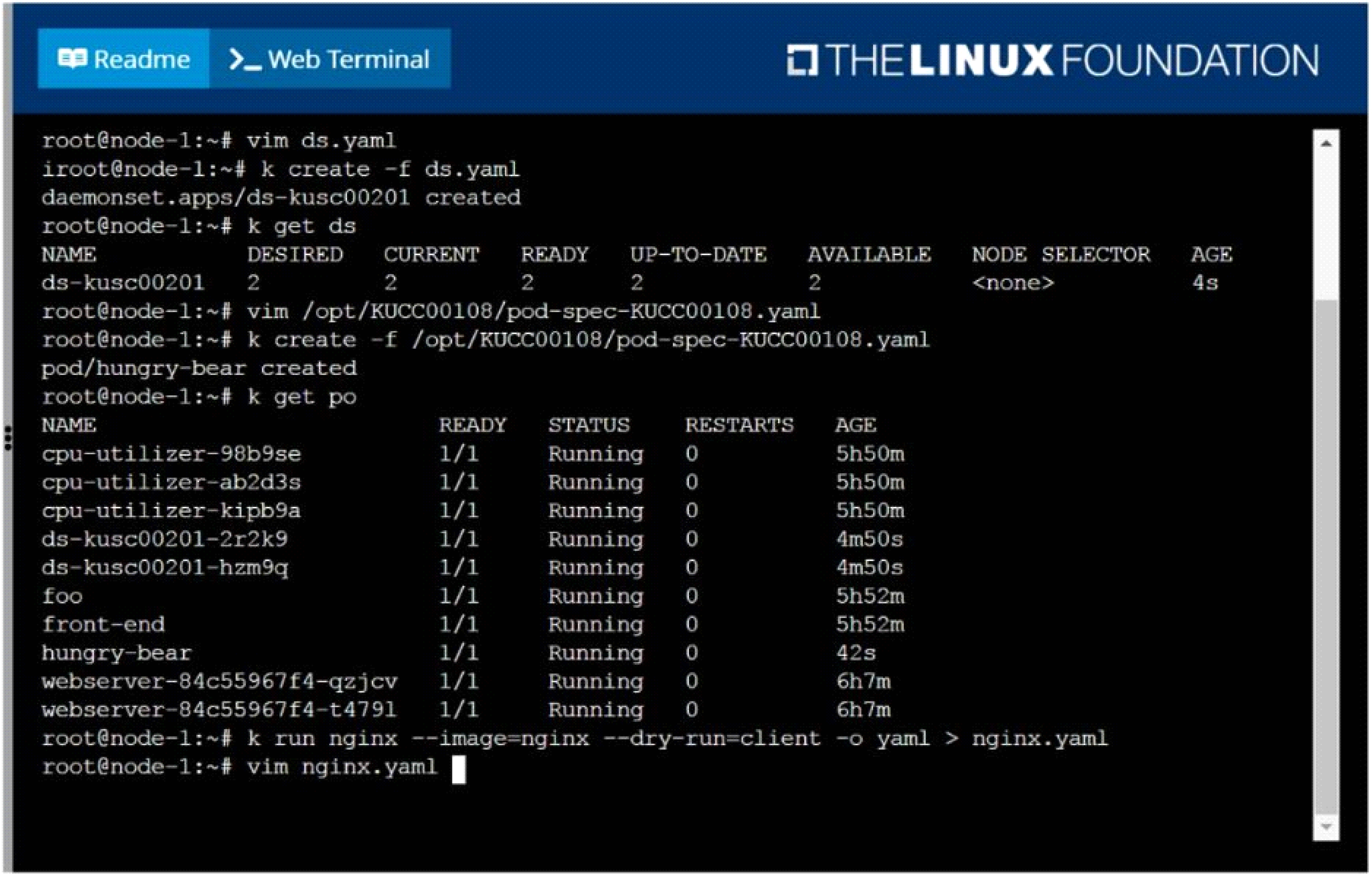


```
apiVersion: v1
kind: Pod
metadata:
  name: kucc8
spec:
  containers:
  - image: nginx
    name: nginx
  - image: redis
    name: redis
  - image: memcached
    name: memcached
~
~
~
~
~
~
~
~
~
~
~
~
:w
```

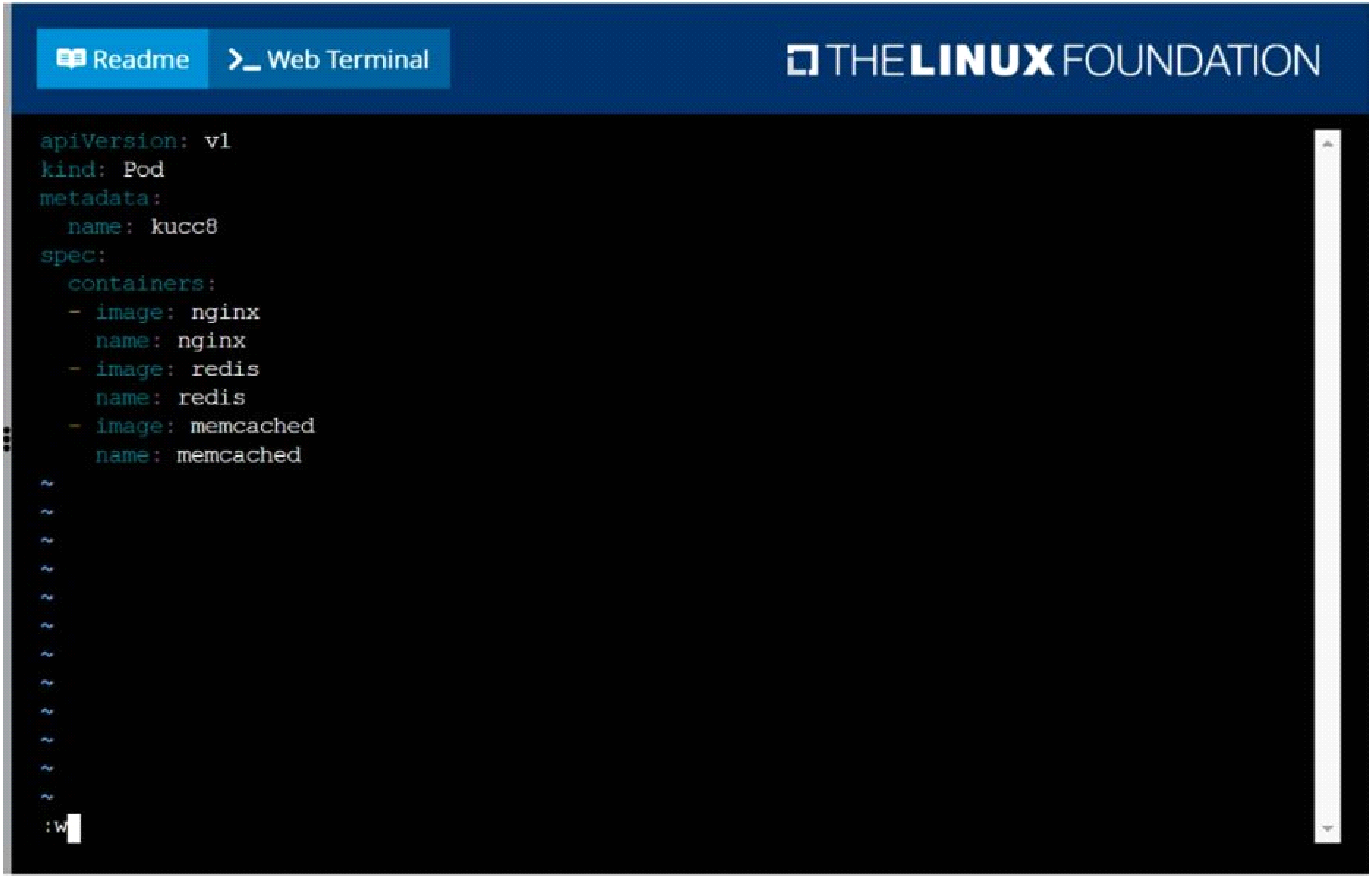


```
cpu-utilizer-98b9se          1/1     Running             0          5h51m
cpu-utilizer-ab2d3s          1/1     Running             0          5h51m
cpu-utilizer-kipb9a          1/1     Running             0          5h51m
ds-kusc00201-2r2k9           1/1     Running             0          6m12s
ds-kusc00201-hzm9q           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:~#
```

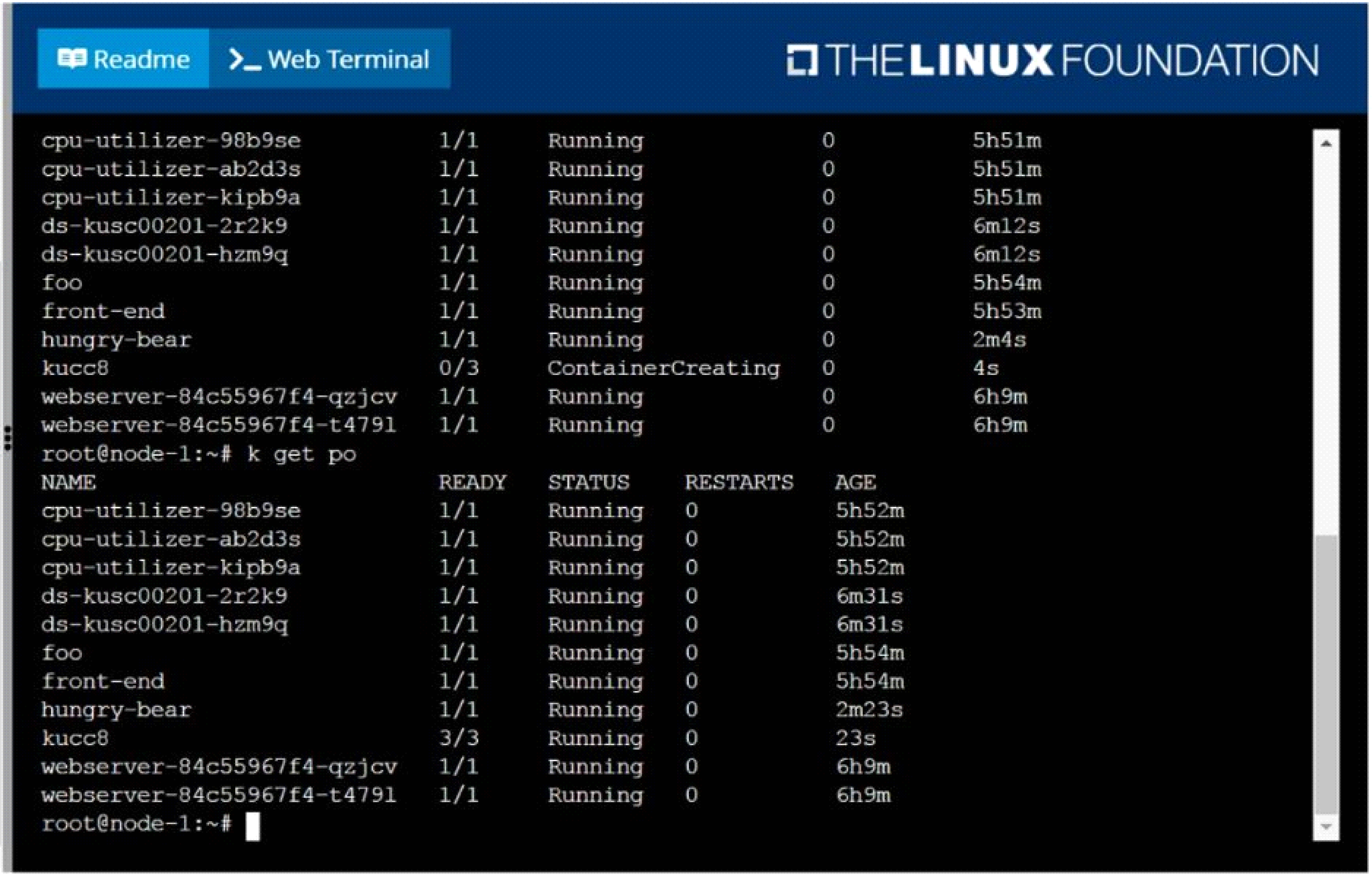
**B)** solution



```
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-t4791   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
```



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



```
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:~#
```

## Answer:

B

# Question 6

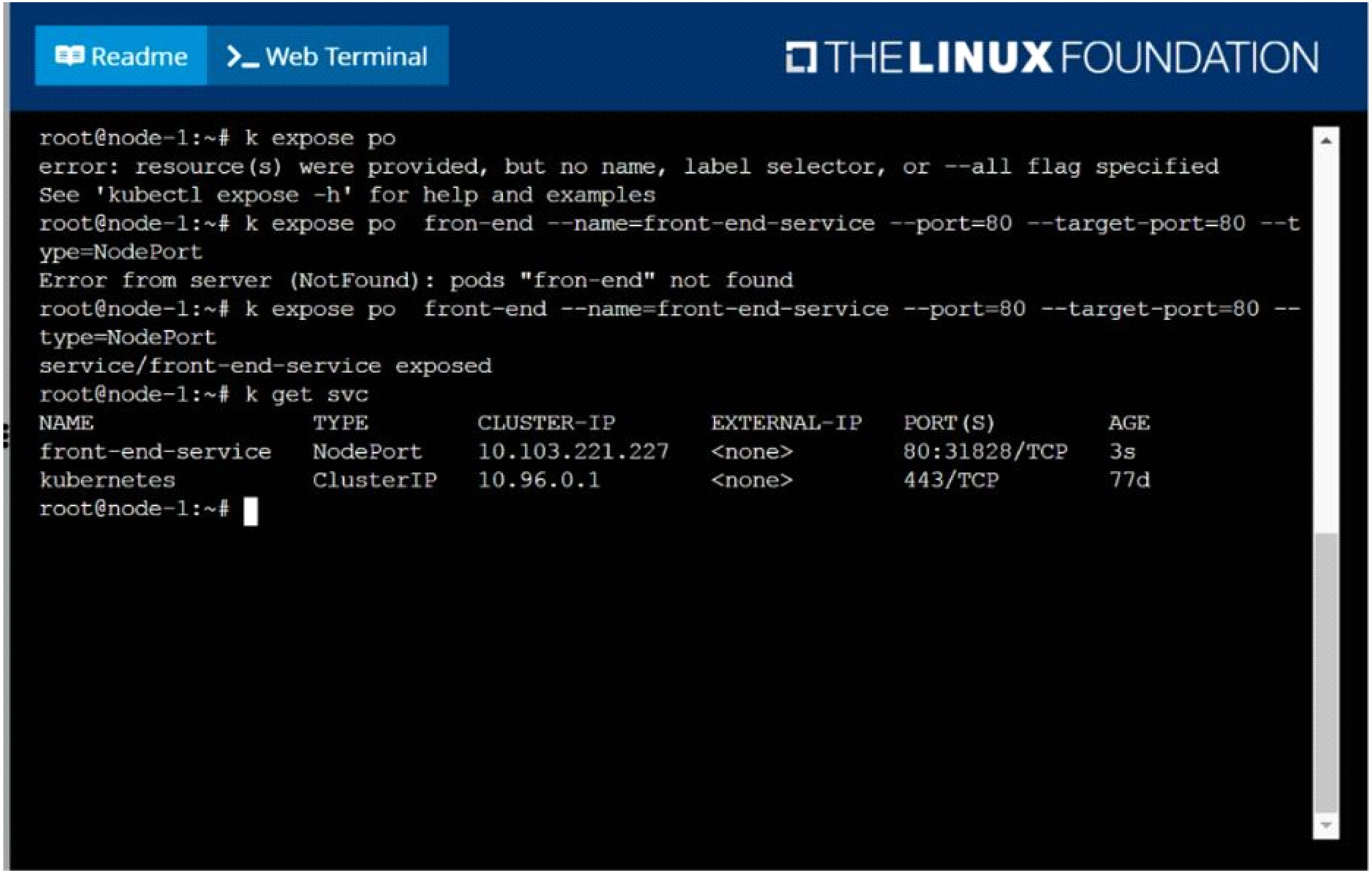
**Question Type: MultipleChoice**

SIMULATION

Create and configure the service front-end-service so it's accessible through NodePort and routes to the existing pod named front-end.

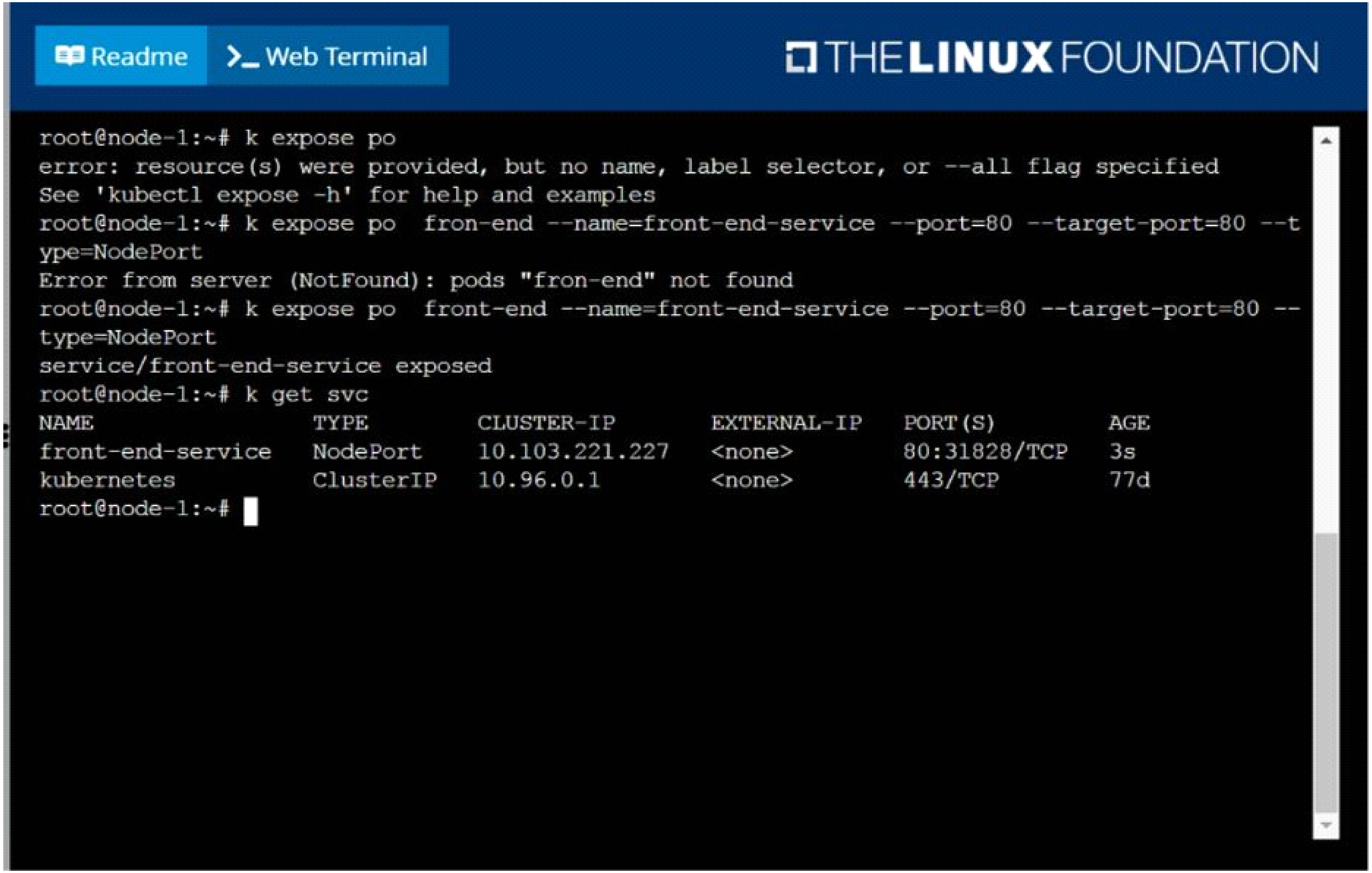
## Options:

**A)** solution



```
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po  fron-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "fron-end" not found
root@node-1:~# k expose po  front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
front-end-service   NodePort    10.103.221.227   <none>        80:31828/TCP   3s
kubernetes          ClusterIP   10.96.0.1        <none>        443/TCP        77d
root@node-1:~#
```

B) solution



```
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po  fron-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "fron-end" not found
root@node-1:~# k expose po  front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
front-end-service   NodePort    10.103.221.227   <none>        80:31828/TCP   3s
kubernetes          ClusterIP   10.96.0.1        <none>        443/TCP        77d
root@node-1:~# 
```

## Answer:

B

# Question 7

**Question Type: MultipleChoice**

SIMULATION

Create a deployment as follows:

* Name: nginx-app

* Using container nginx with version 1.11.10-alpine

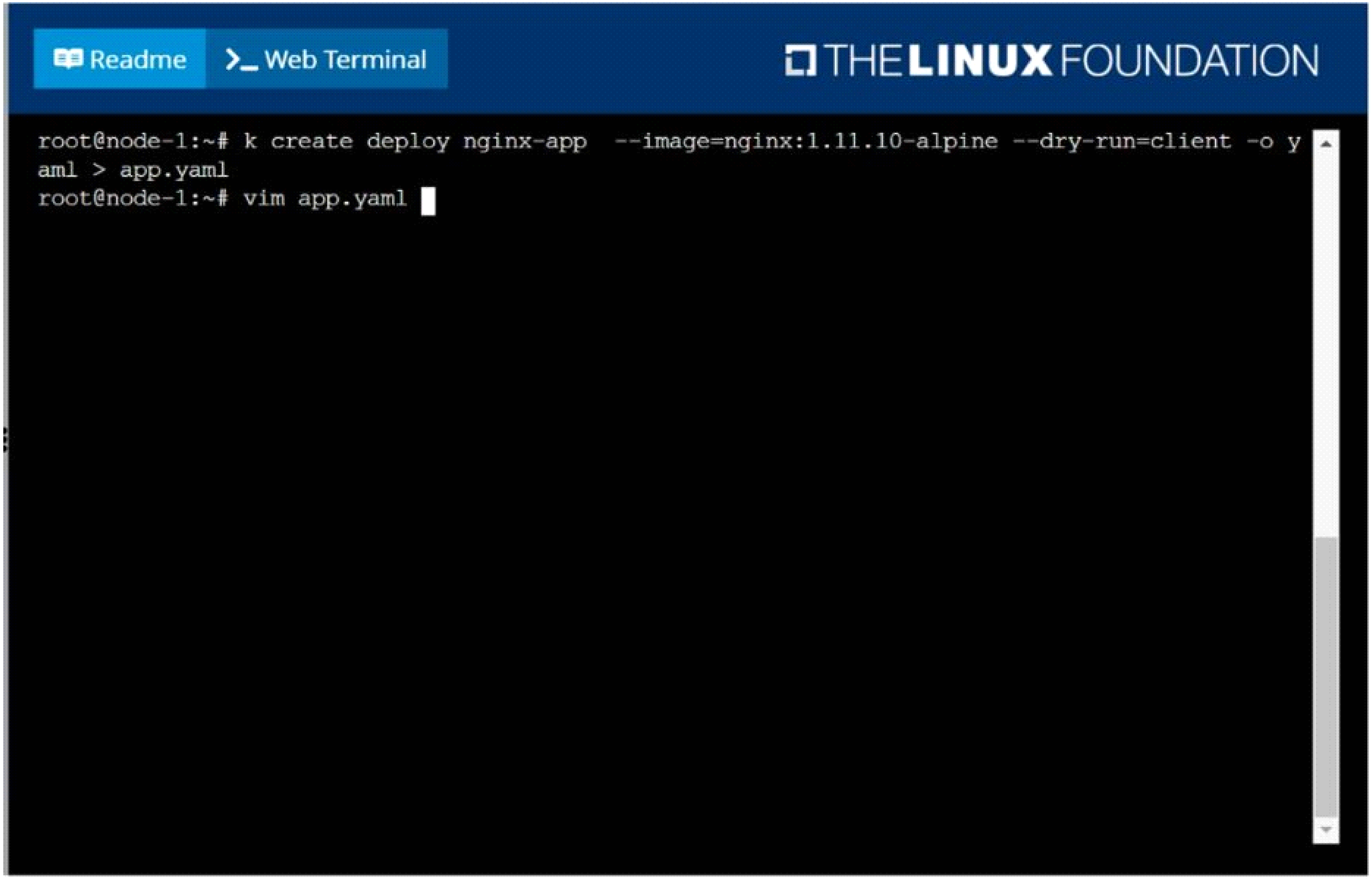
* The deployment should contain 3 replicas

Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update.

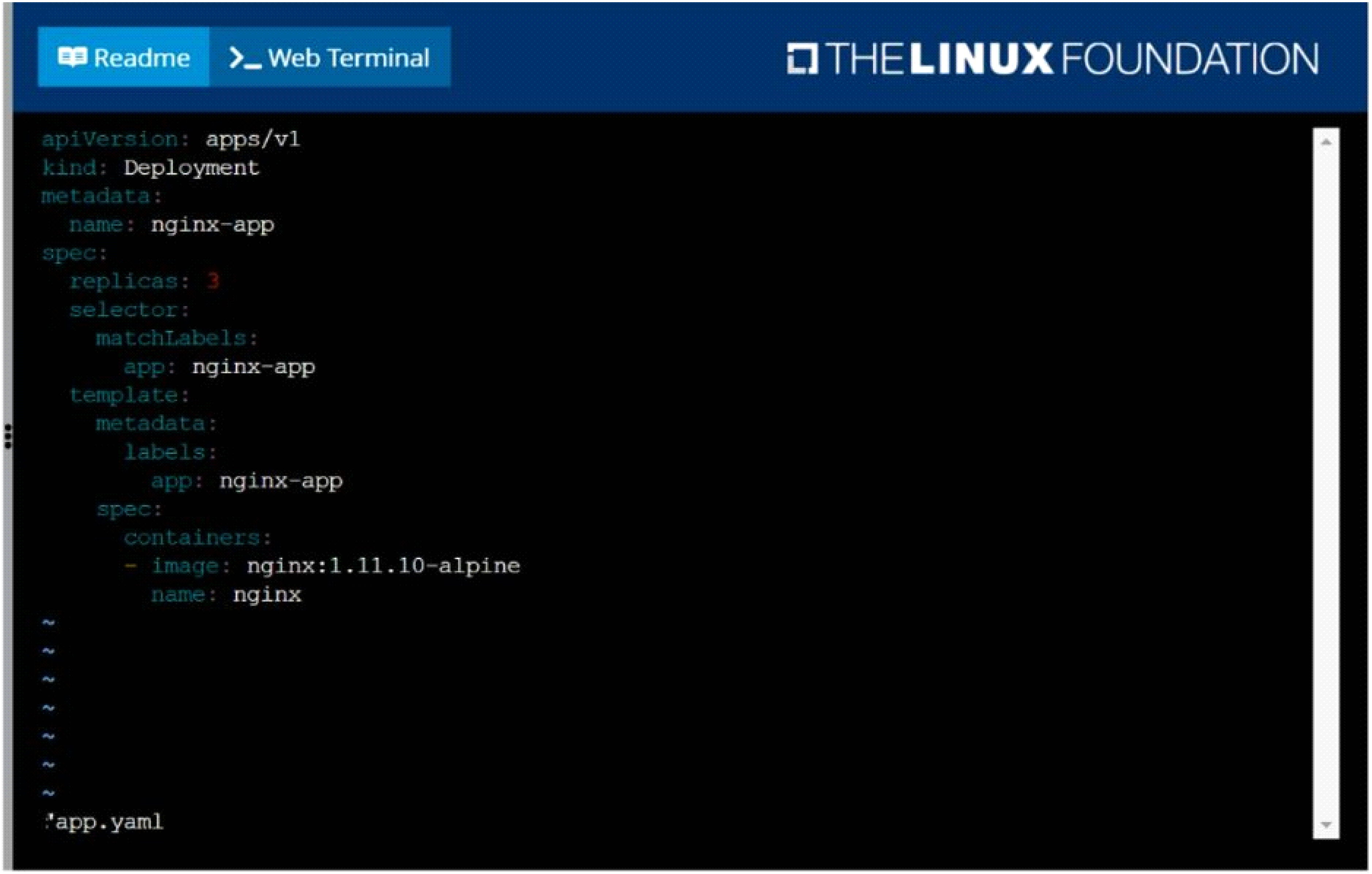
Finally, rollback that update to the previous version 1.11.10-alpine.

## Options:

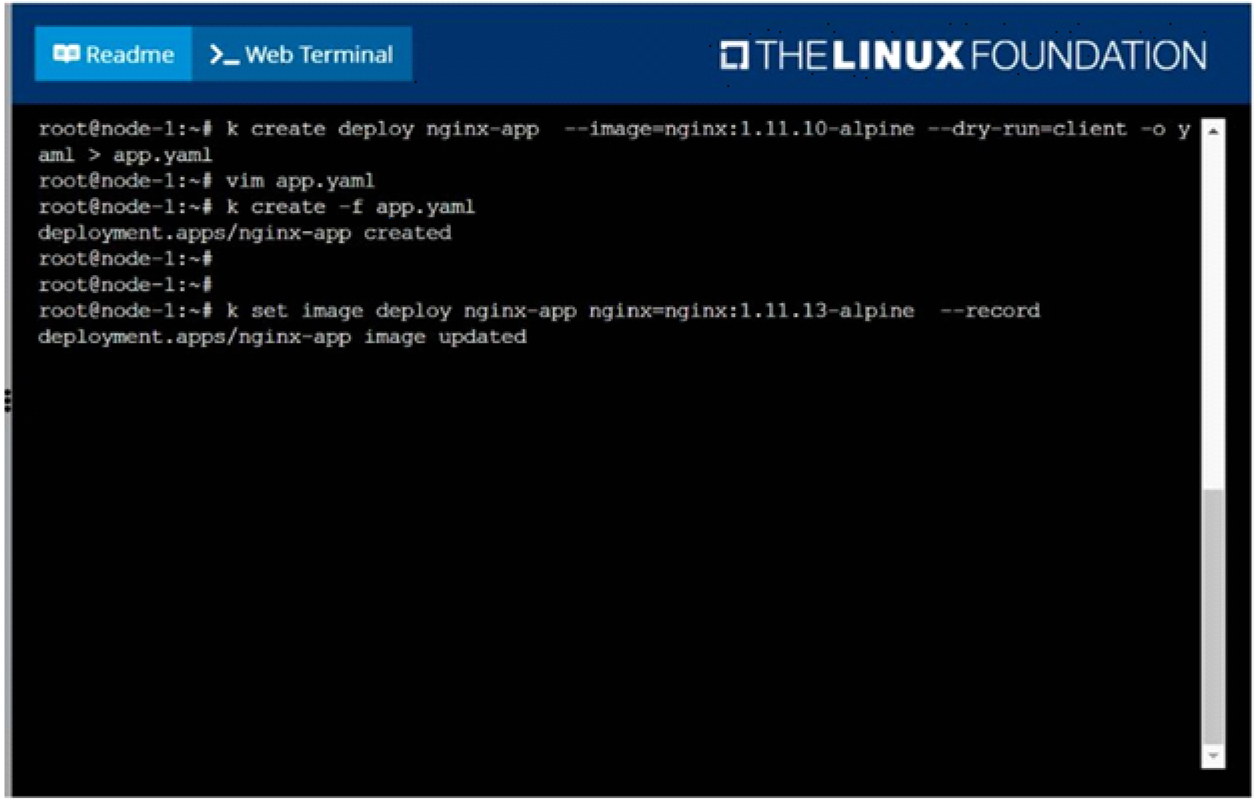
**A)** solution



```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
```

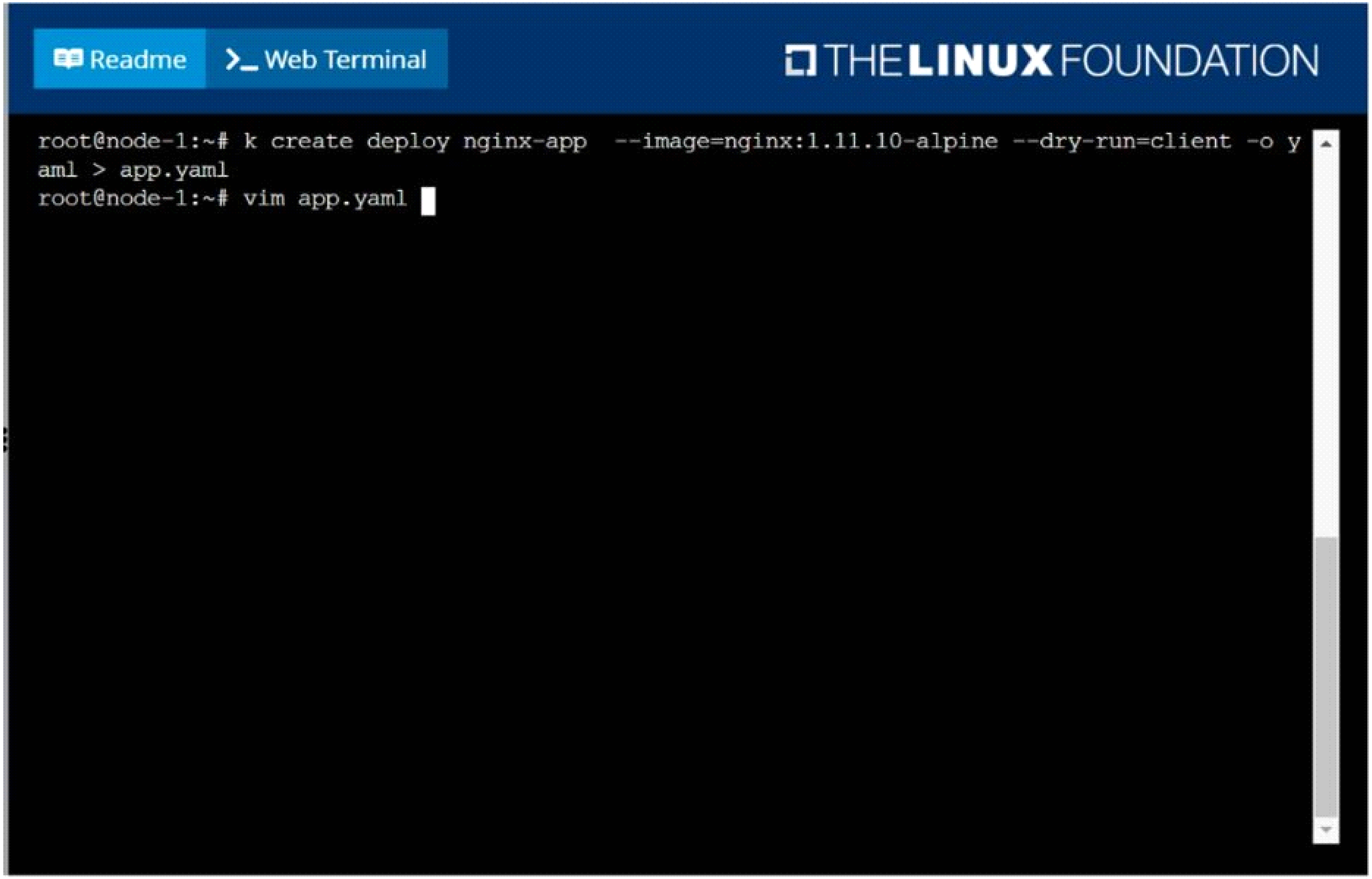


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      labels:
        app: nginx-app
    spec:
      containers:
      - image: nginx:1.11.10-alpine
        name: nginx
~
~
~
~
~
~
~
"app.yaml
```

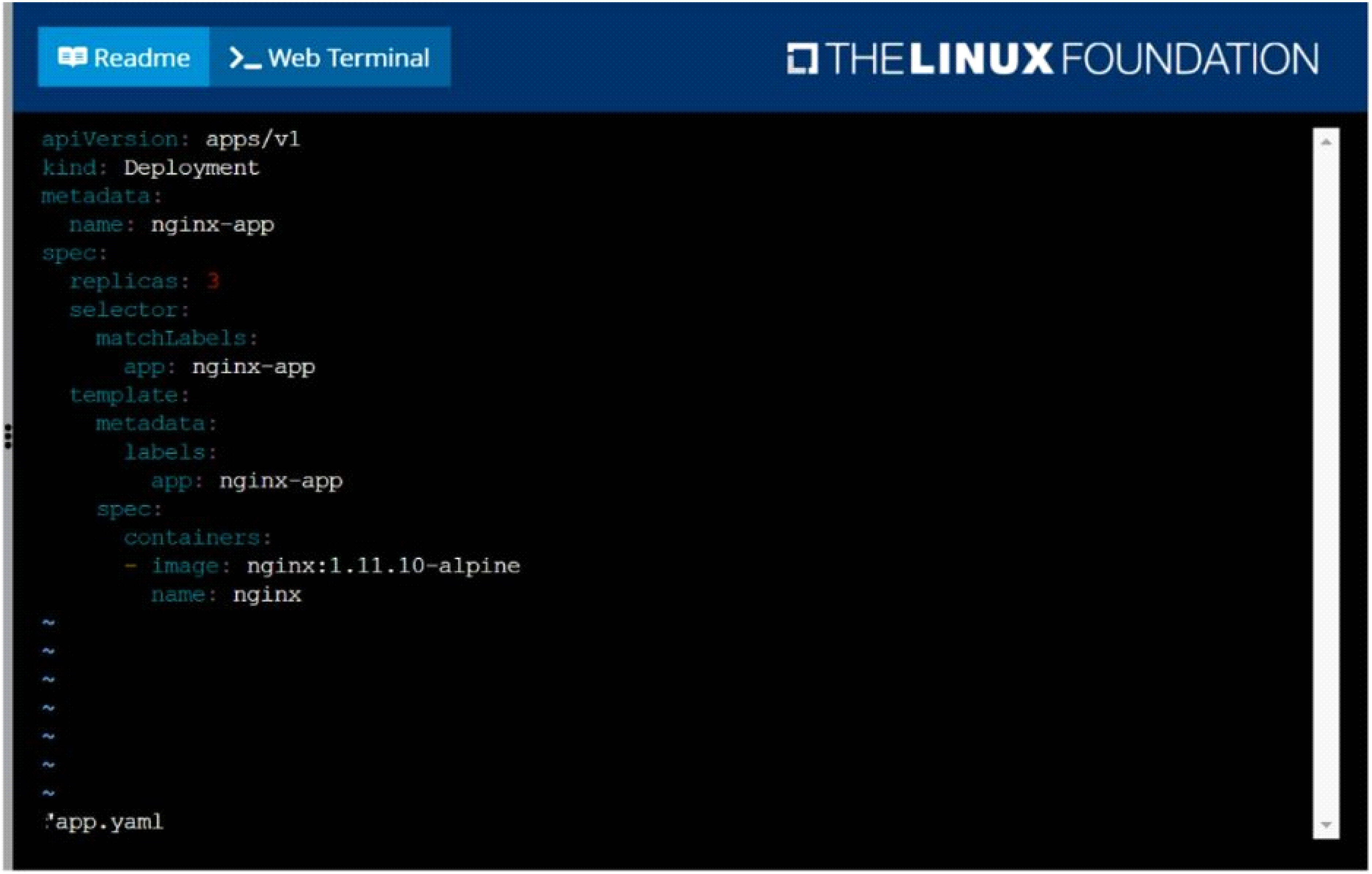


```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
root@node-1:~# k create -f app.yaml
deployment.apps/nginx-app created
root@node-1:~#
root@node-1:~#
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine  --record
deployment.apps/nginx-app image updated
```

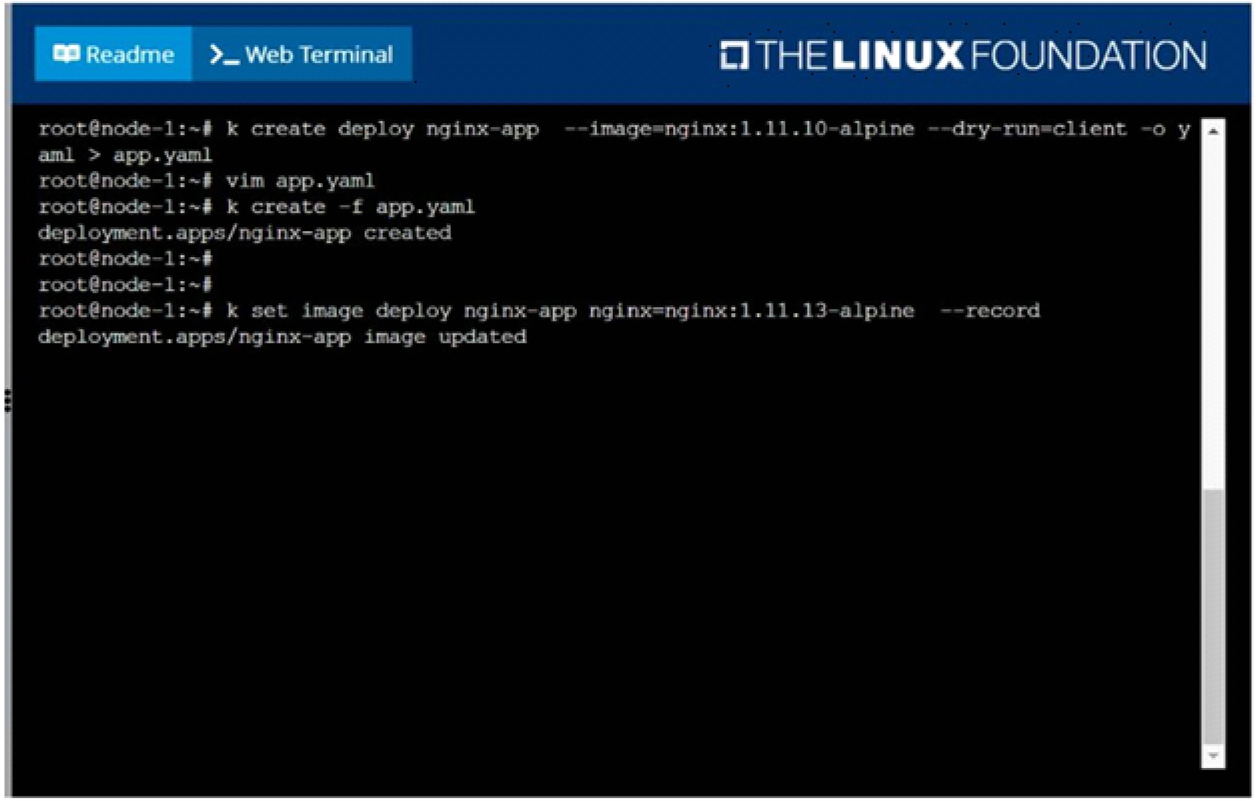
**B)** solution



```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
```



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      labels:
        app: nginx-app
    spec:
      containers:
      - image: nginx:1.11.10-alpine
        name: nginx
~
~
~
~
~
~
~
"app.yaml
```



```
root@node-1:~# k create deploy nginx-app  --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
root@node-1:~# k create -f app.yaml
deployment.apps/nginx-app created
root@node-1:~#
root@node-1:~#
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine  --record
deployment.apps/nginx-app image updated
```

## Answer:

A