



Exam Code: CKA

Exam Name: Certified Kubernetes Administrator

Website: www.VCEplus.io

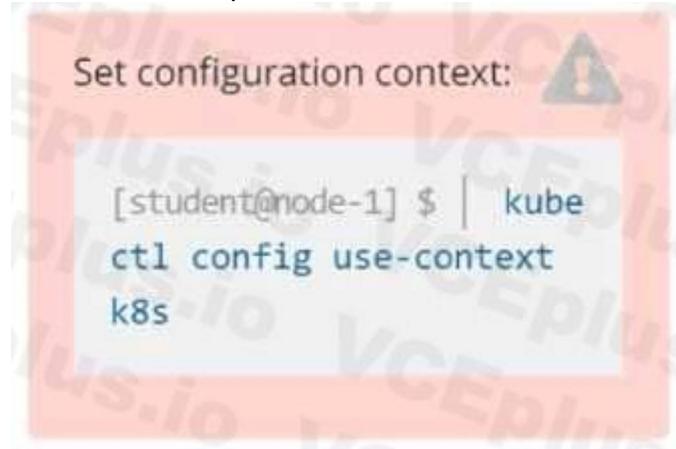
Twitter: www.twitter.com/VCE_Plus

Exam A**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



Set configuration context:

```
[student@node-1] $ | kube
ctl config use-context
k8s
```

A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

www.VCEplus.io

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

www.VCEplus.io

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.

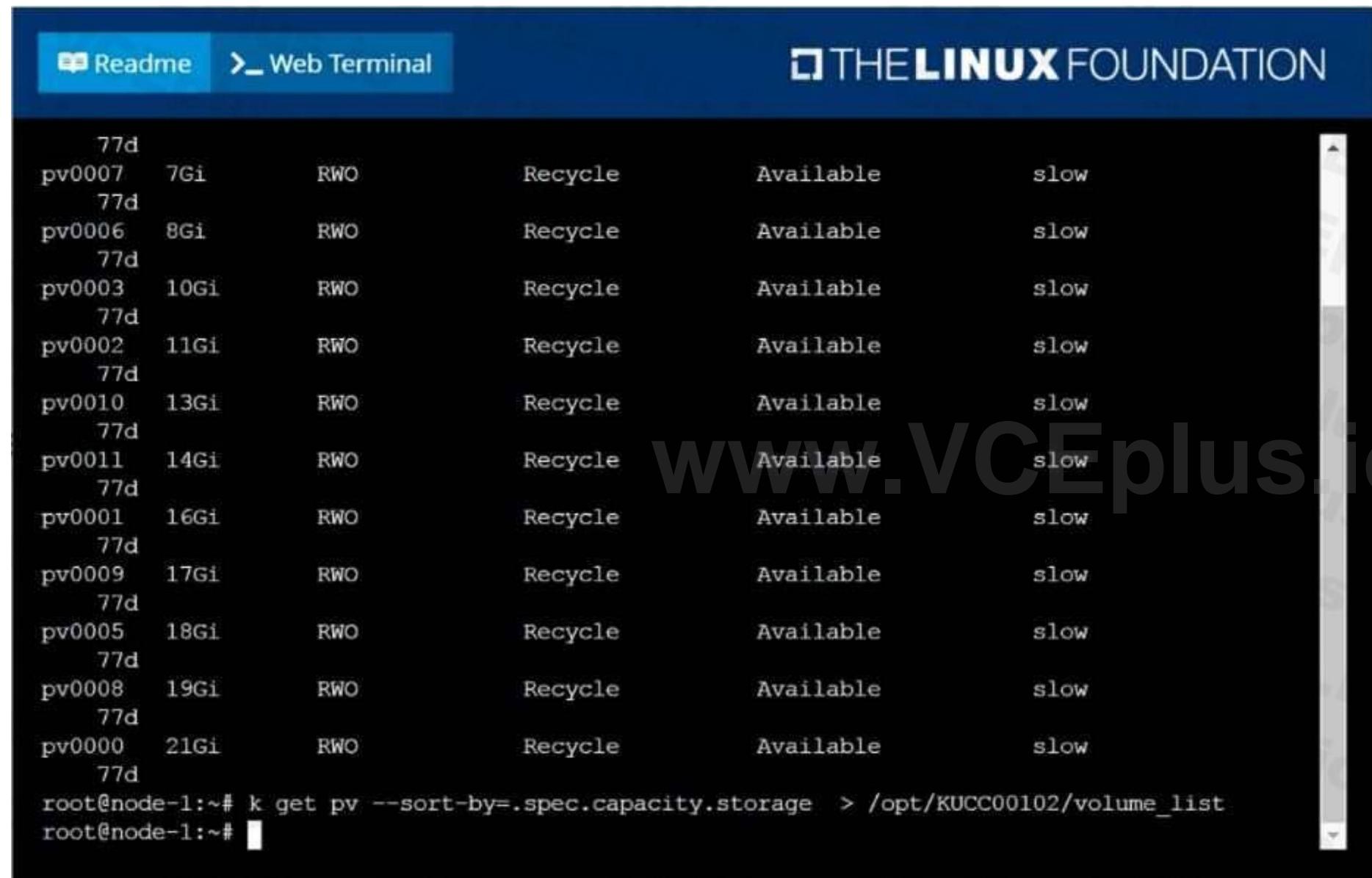
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window titled "Web Terminal" under "Readme". The title bar also features "THE LINUX FOUNDATION". The terminal displays the following command and its output:

```
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
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.

A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

www.VCEplus.io

Readme Web Terminal

THE LINUX FOUNDATION

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

www.VCEplus.io

Readme Web Terminal

THE LINUX FOUNDATION

```
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:
```

VCEUP VCEplus

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
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:~#
```

QUESTION 4

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

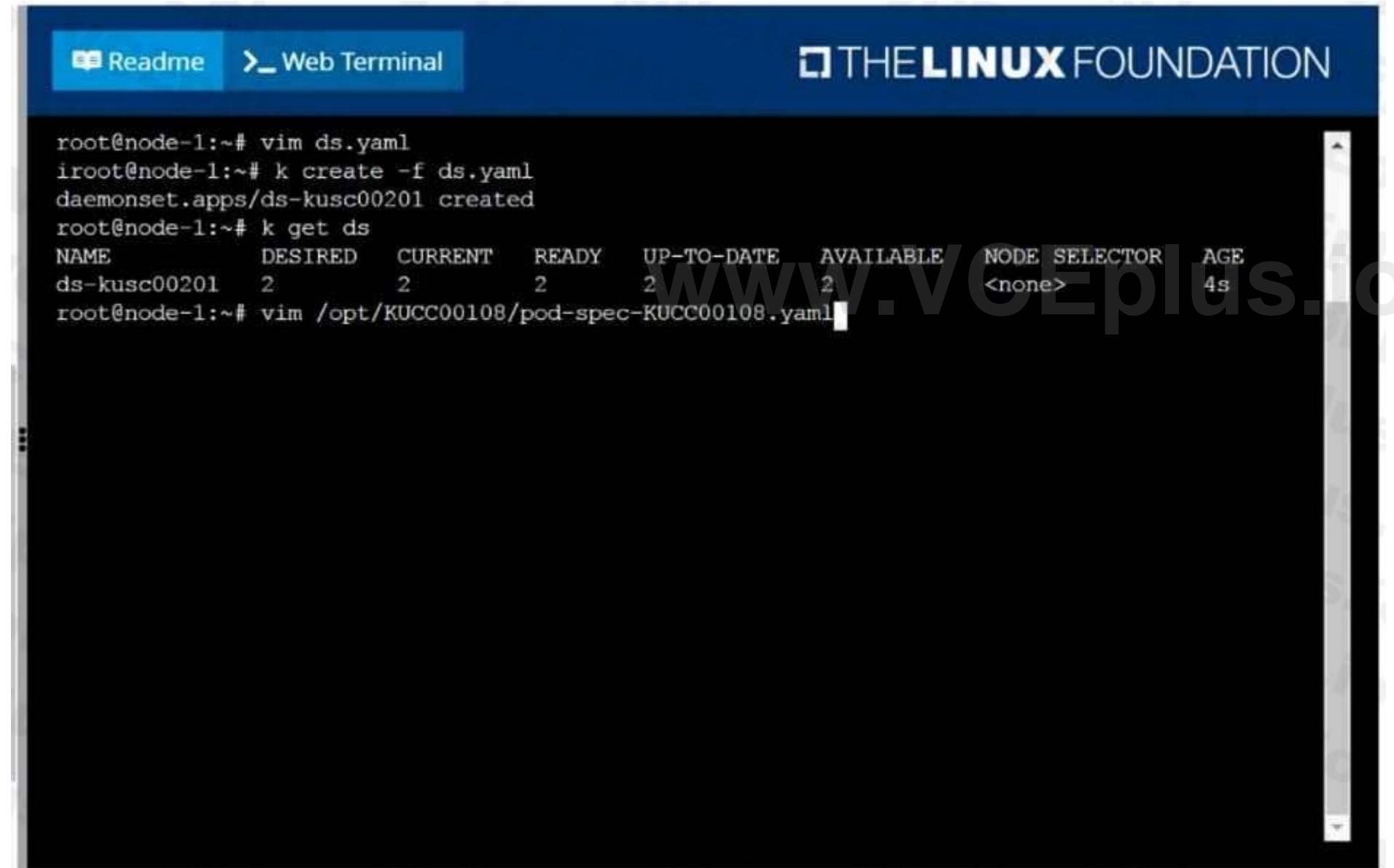
A. See the solution below.

Correct Answer: A

Section:

Explanation:

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
```

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 ];\n        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
: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:~# 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.

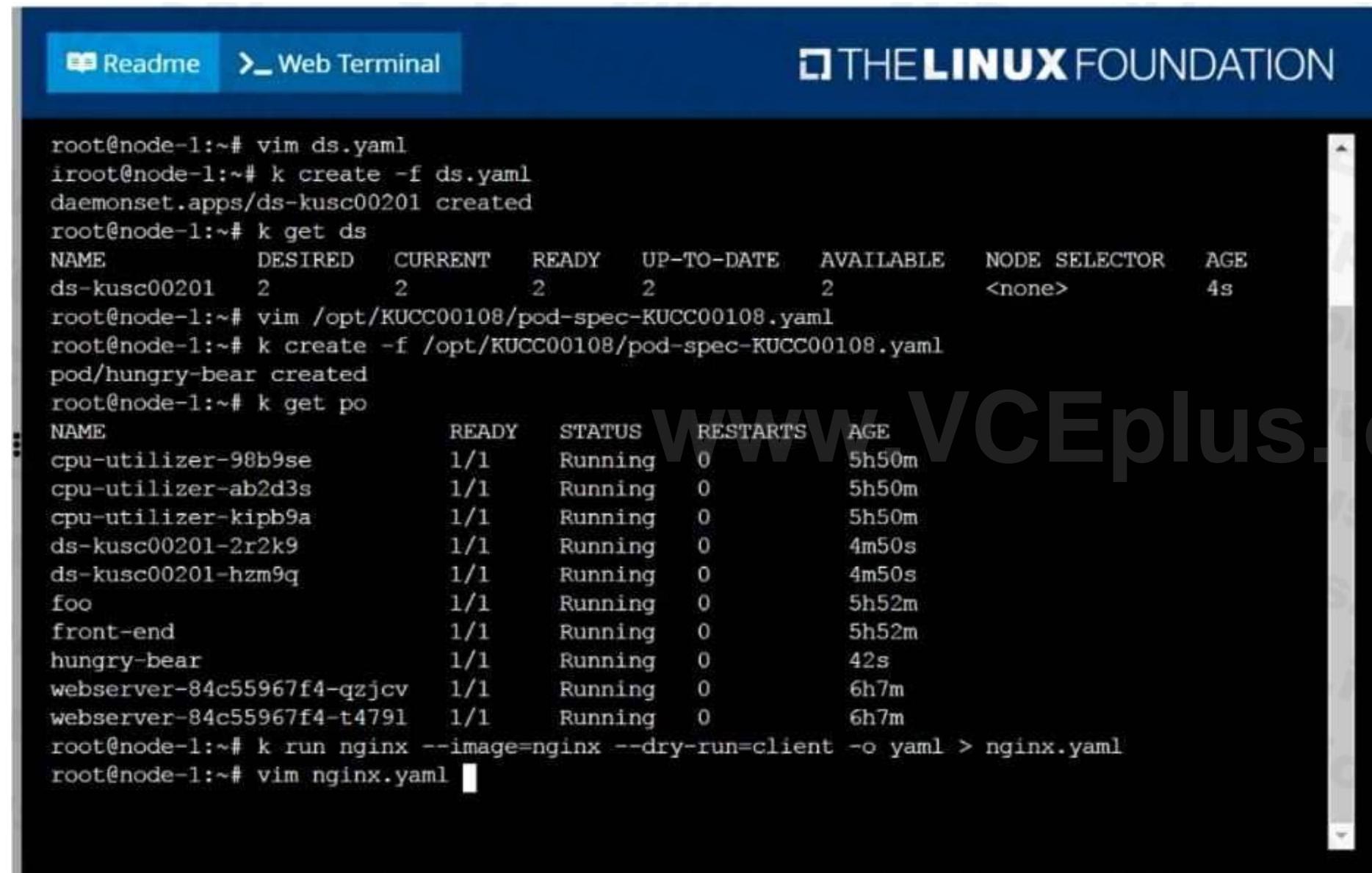
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window titled "Web Terminal" with the "THE LINUX FOUNDATION" logo at the top. The terminal content is as follows:

```
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
```

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
```

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

:w

www.VCEplus.io

Readme > Web Terminal

THE LINUX FOUNDATION

| NAME | READY | STATUS | RESTARTS | AGE |
|----------------------------|-------|-------------------|----------|-------|
| 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 | | | | |

QUESTION 6

Schedule a pod as follows:

Name: nginx-kusc00101

Image: nginx

Node selector: disk=ssd

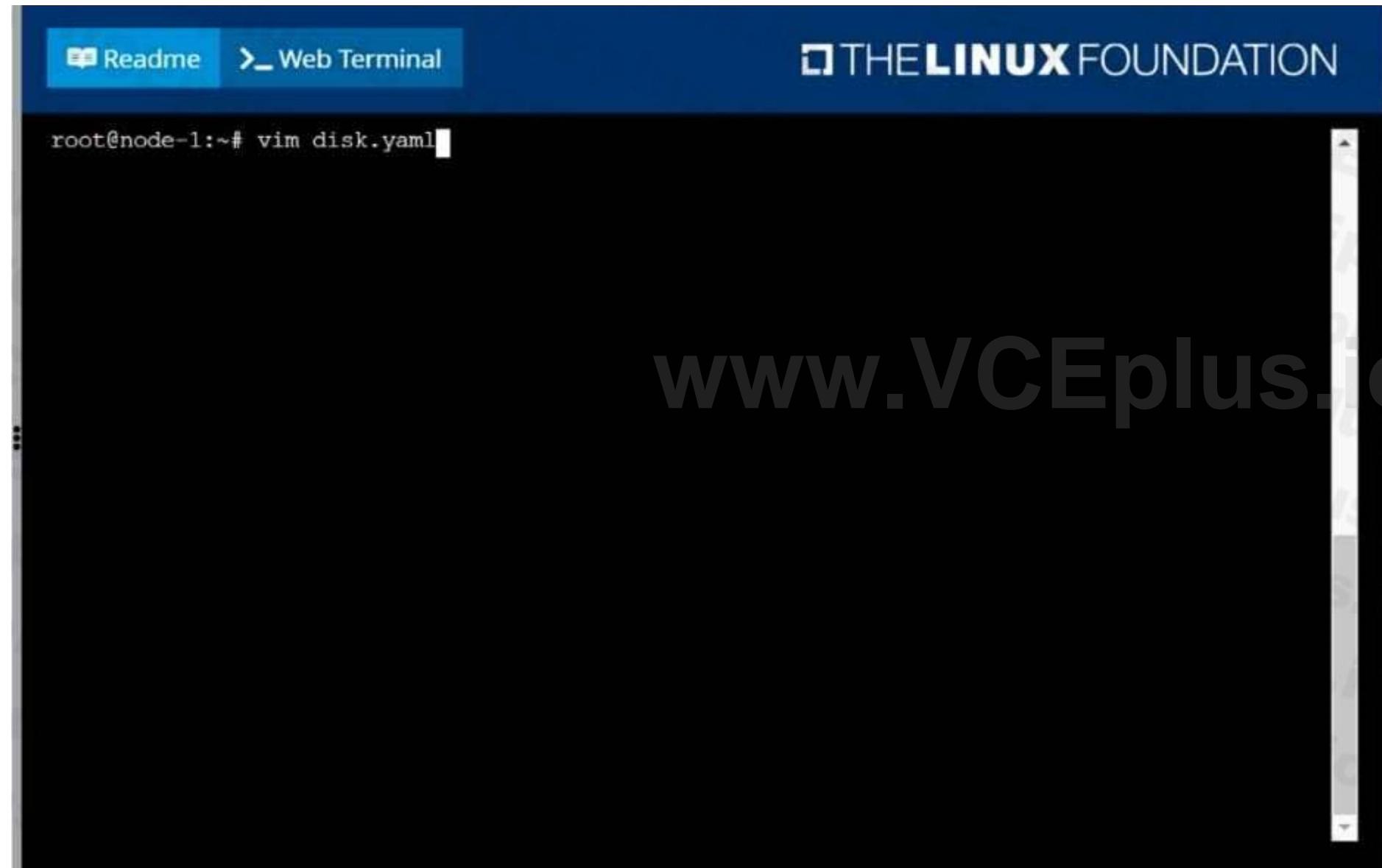
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window titled "Web Terminal". At the top, there are two tabs: "Readme" and "Web Terminal". The main area of the terminal shows the command "root@node-1:~# vim disk.yaml" entered by the user. The background of the terminal window features the "THE LINUX FOUNDATION" logo.

Readme > Web Terminal THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kusc00101
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    disk: ssd

"disk.yaml" [New] 11L, 176C written
```

Readme > Web Terminal THE LINUX FOUNDATION

```
root@node-1:~# vim disk.yaml
root@node-1:~# k create -f disk.yaml
pod/nginx-kusc00101 created
root@node-1:~# k get po
NAME          READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se   1/1     Running   0          5h59m
cpu-utilizer-ab2d3s   1/1     Running   0          5h59m
cpu-utilizer-kipb9a   1/1     Running   0          5h59m
ds-kusc00201-2r2k9   1/1     Running   0          13m
ds-kusc00201-hzm9q   1/1     Running   0          13m
foo-front-end        1/1     Running   0          6h1m
foo-hungry-bear       1/1     Running   0          6h1m
foo-hungry-bear       1/1     Running   0          9m37s
```

QUESTION 7

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.

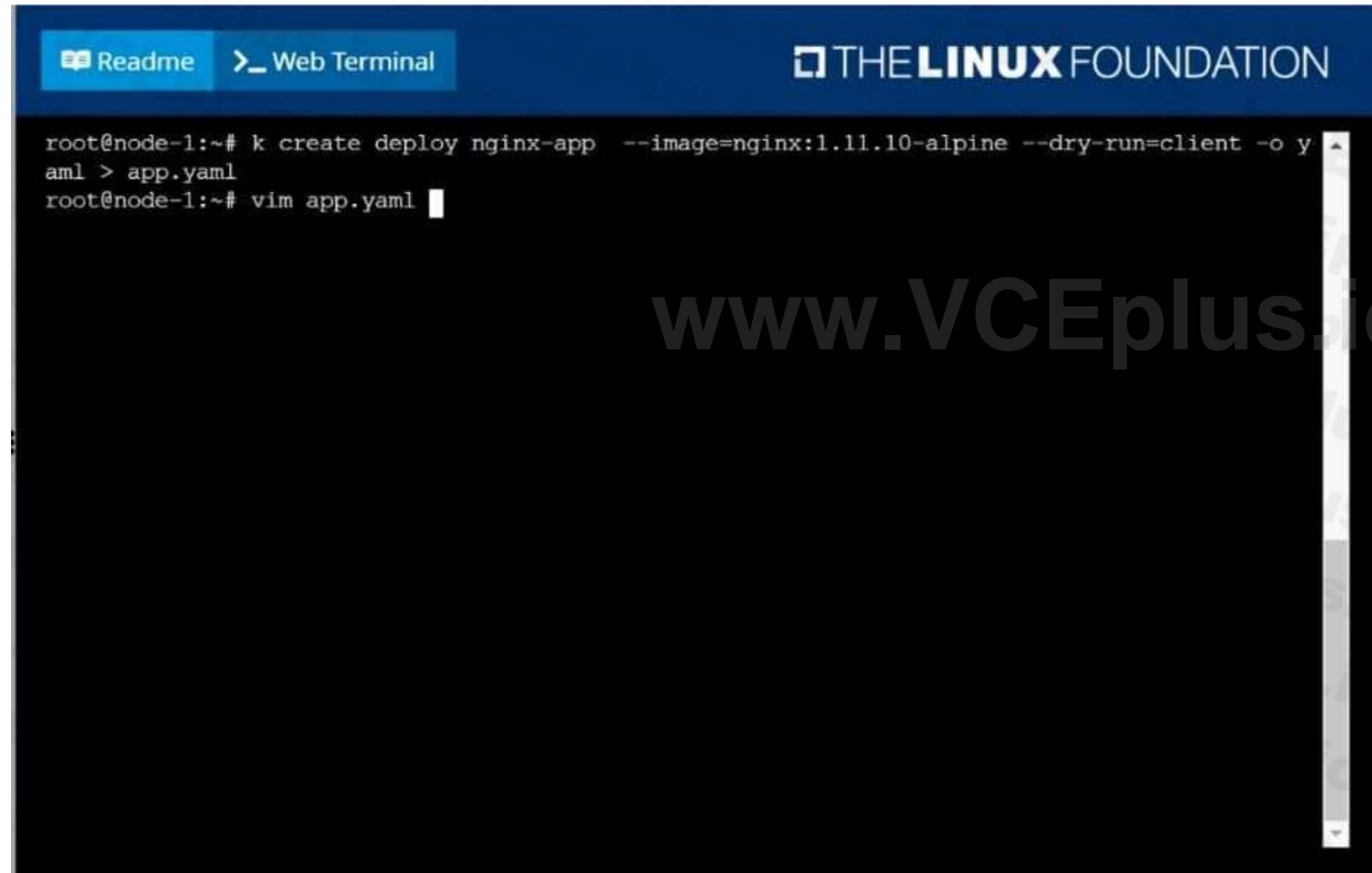
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window with a blue header bar. On the left of the header bar are two buttons: "Readme" and "Web Terminal". To the right of the header bar is the text "THE LINUX FOUNDATION". The main area of the terminal contains the following text:

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

www.VCEplus.io

Readme > Web Terminal

THE LINUX FOUNDATION

```
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

Readme > Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~# k create deploy nginx-app --image=nginx:1.11.10-alpine --dry-run=client -o yaml > 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
root@node-1:~# k rollout undo deploy nginx-app
deployment.apps/nginx-app rolled back
root@node-1:~#
```

QUESTION 8

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

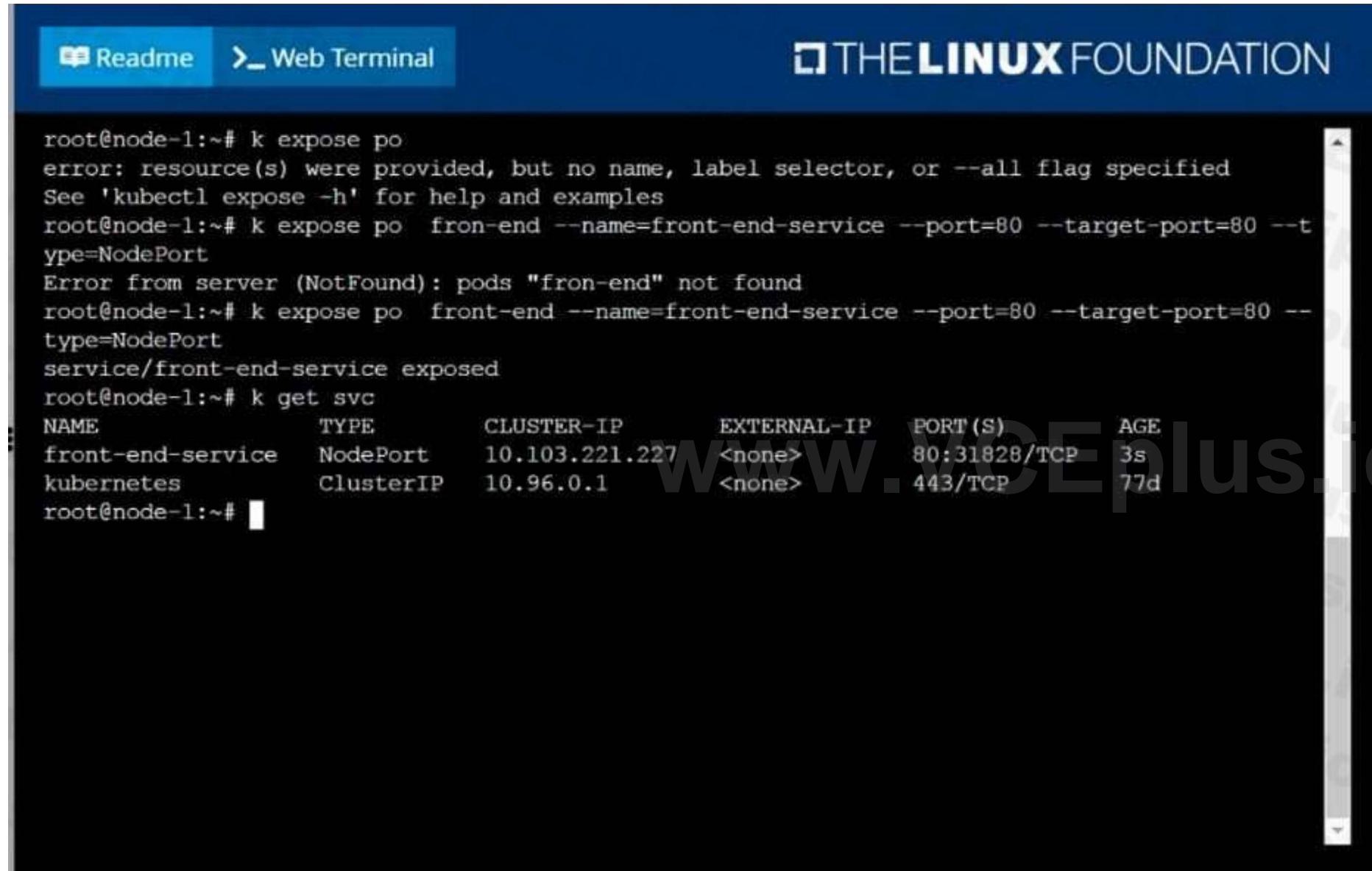
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window with two tabs: "Readme" and "Web Terminal". The "Web Terminal" tab is active and displays the following terminal session:

```
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 front-end --name=front-end-service --port=80 --target-port=80 --type=NodePort
Error from server (NotFound): pods "front-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:~#
```

QUESTION 9

Create a pod as follows:

Name: mongo

Using Image: mongo

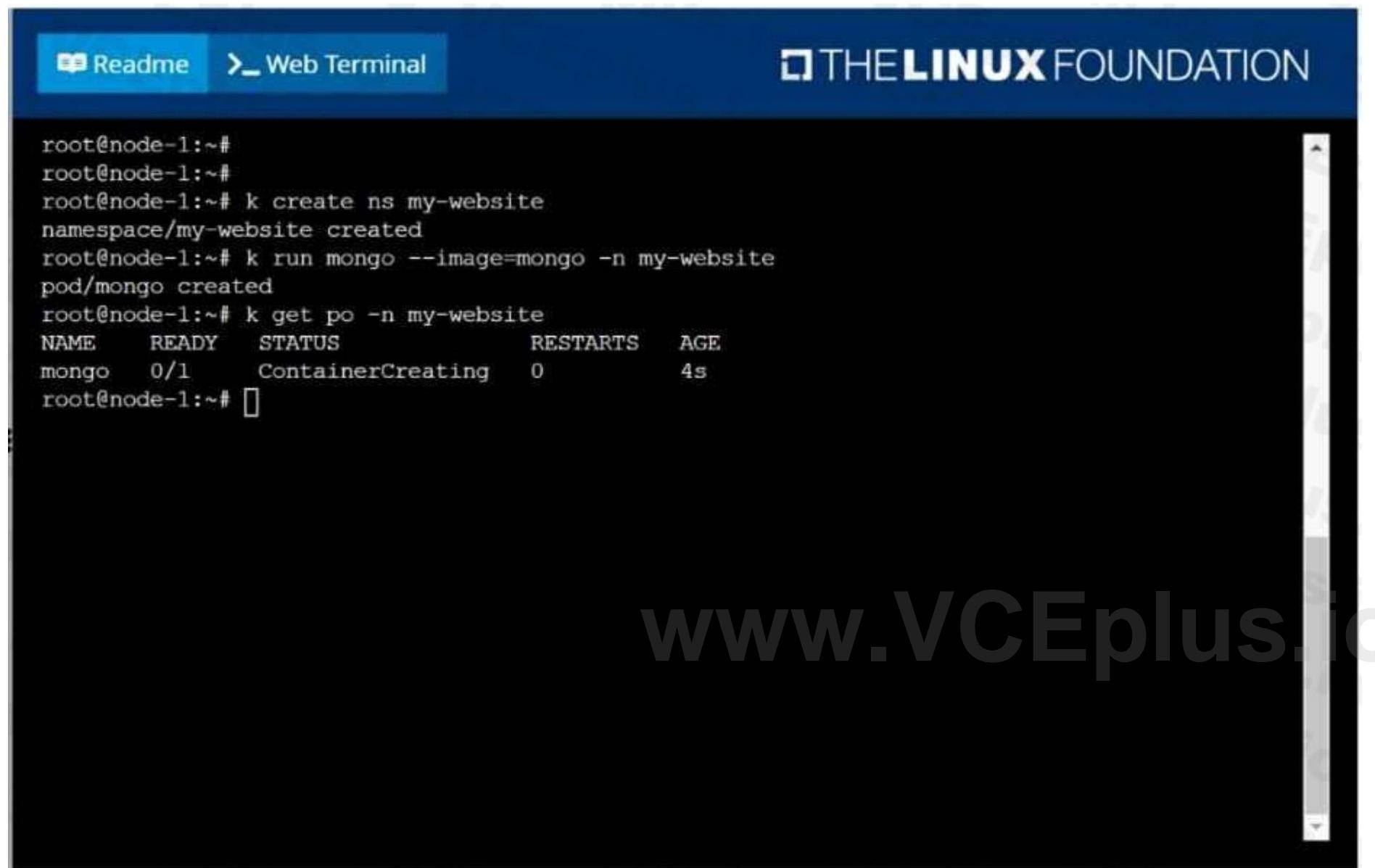
In a new Kubernetes namespace named: my-website

A. See the solution below.

Correct Answer: A

Section:

Explanation:
solution



The screenshot shows a terminal window with a blue header containing the 'Readme' and 'Web Terminal' tabs, and the 'THE LINUX FOUNDATION' logo. The terminal output is as follows:

```
root@node-1:~# k create ns my-website
namespace/my-website created
root@node-1:~# k run mongo --image=mongo -n my-website
pod/mongo created
root@node-1:~# k get po -n my-website
NAME      READY   STATUS        RESTARTS   AGE
mongo    0/1     ContainerCreating   0          4s
root@node-1:~# []
```

QUESTION 10

Create a deployment spec file that will:

Launch 7 replicas of the nginx Image with the label app_runtime_stage=dev deployment name: kual00201

Save a copy of this spec file to /opt/KUAL00201/spec_deployment.yaml

(or /opt/KUAL00201/spec_deployment.json).

When you are done, clean up (delete) any new Kubernetes API object that you produced during this task.

A. See the solution below.

Correct Answer: A

Section:

Explanation:
solution

Readme > Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~# k create deploy kual00201 --image=nginx --dry-run=client -o yaml > /opt/KUAL00201/spec_deployment.yaml
root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml
```

www.VCEplus.io

Readme > Web Terminal

THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app_runtime_stage: dev
  name: kual00201
spec:
  replicas: 7
  selector:
    matchLabels:
      app_runtime_stage: dev
  template:
    metadata:
```

VCEUp  VCEplus

QUESTION 11

Create a file:

/opt/KUCC00302/kucc00302.txt that lists all pods that implement service baz in namespace development.

The format of the file should be one pod name per line.

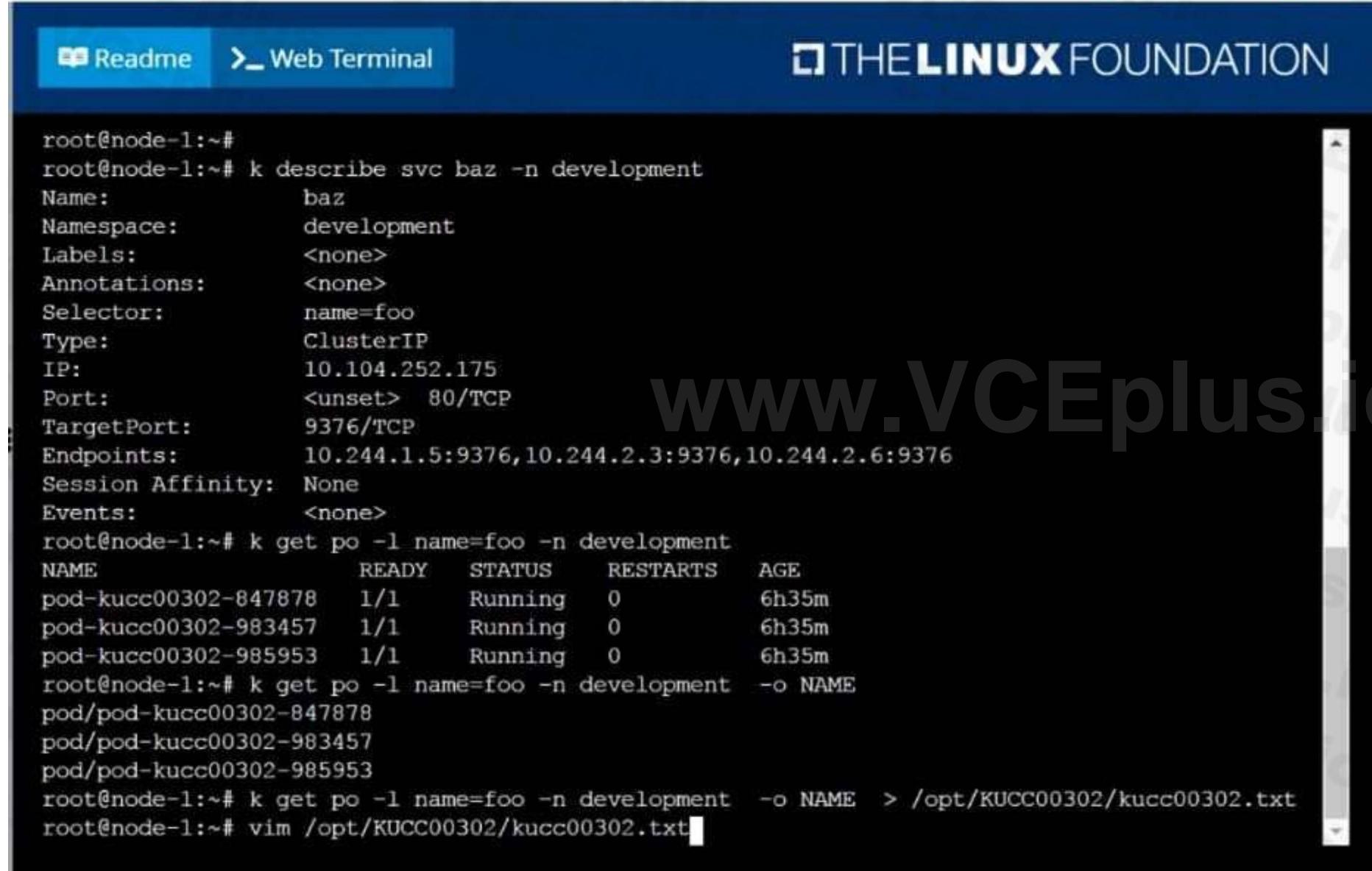
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window with the title bar "THE LINUX FOUNDATION". The window has two tabs: "Readme" (selected) and "Web Terminal". The terminal content is as follows:

```
root@node-1:~# k describe svc baz -n development
Name:          baz
Namespace:     development
Labels:        <none>
Annotations:   <none>
Selector:      name=foo
Type:          ClusterIP
IP:            10.104.252.175
Port:          <unset>  80/TCP
TargetPort:    9376/TCP
Endpoints:    10.244.1.5:9376,10.244.2.3:9376,10.244.2.6:9376
Session Affinity: None
Events:        <none>
root@node-1:~# k get po -l name=foo -n development
NAME          READY   STATUS    RESTARTS   AGE
pod-kucc00302-847878  1/1     Running   0          6h35m
pod-kucc00302-983457  1/1     Running   0          6h35m
pod-kucc00302-985953  1/1     Running   0          6h35m
root@node-1:~# k get po -l name=foo -n development -o NAME
pod/pod-kucc00302-847878
pod/pod-kucc00302-983457
pod/pod-kucc00302-985953
root@node-1:~# k get po -l name=foo -n development -o NAME > /opt/KUCC00302/kucc00302.txt
root@node-1:~# vim /opt/KUCC00302/kucc00302.txt
```

A screenshot of a terminal window with a dark background. At the top, there are two blue buttons: 'Readme' on the left and 'Web Terminal' on the right. In the center, the text 'THE LINUX FOUNDATION' is displayed next to its logo. The main area of the terminal shows a list of pod names: 'pod-kucc00302-847878', 'pod-kucc00302-983457', and 'pod-kucc00302-985953'. Below these, there are several small, illegible entries that appear to be truncated. A large watermark 'www.VCEplus.io' is overlaid across the bottom half of the screen.

A screenshot of a terminal window with a blue header bar. The header bar contains two buttons: 'Readme' and 'Web Terminal'. To the right of the header is the 'THE LINUX FOUNDATION' logo. The main content area of the terminal shows the following output:

```
Name: baz
Namespace: development
Labels: <none>
Annotations: <none>
Selector: name=foo
Type: ClusterIP
IP: 10.104.252.175
Port: <unset> 80/TCP
TargetPort: 9376/TCP
Endpoints: 10.244.1.5:9376,10.244.2.3:9376,10.244.2.6:9376
Session Affinity: None
Events: <none>
```

At the bottom of the terminal, the command used to retrieve the pod information is visible: `root@node-1:~# k get po -l name=foo -n development`.

QUESTION 12

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 supersecret at /secrets.

Create a second pod named pod-secrets-via-env, using the redis Image, which exports password as

CONFIDENTIAL

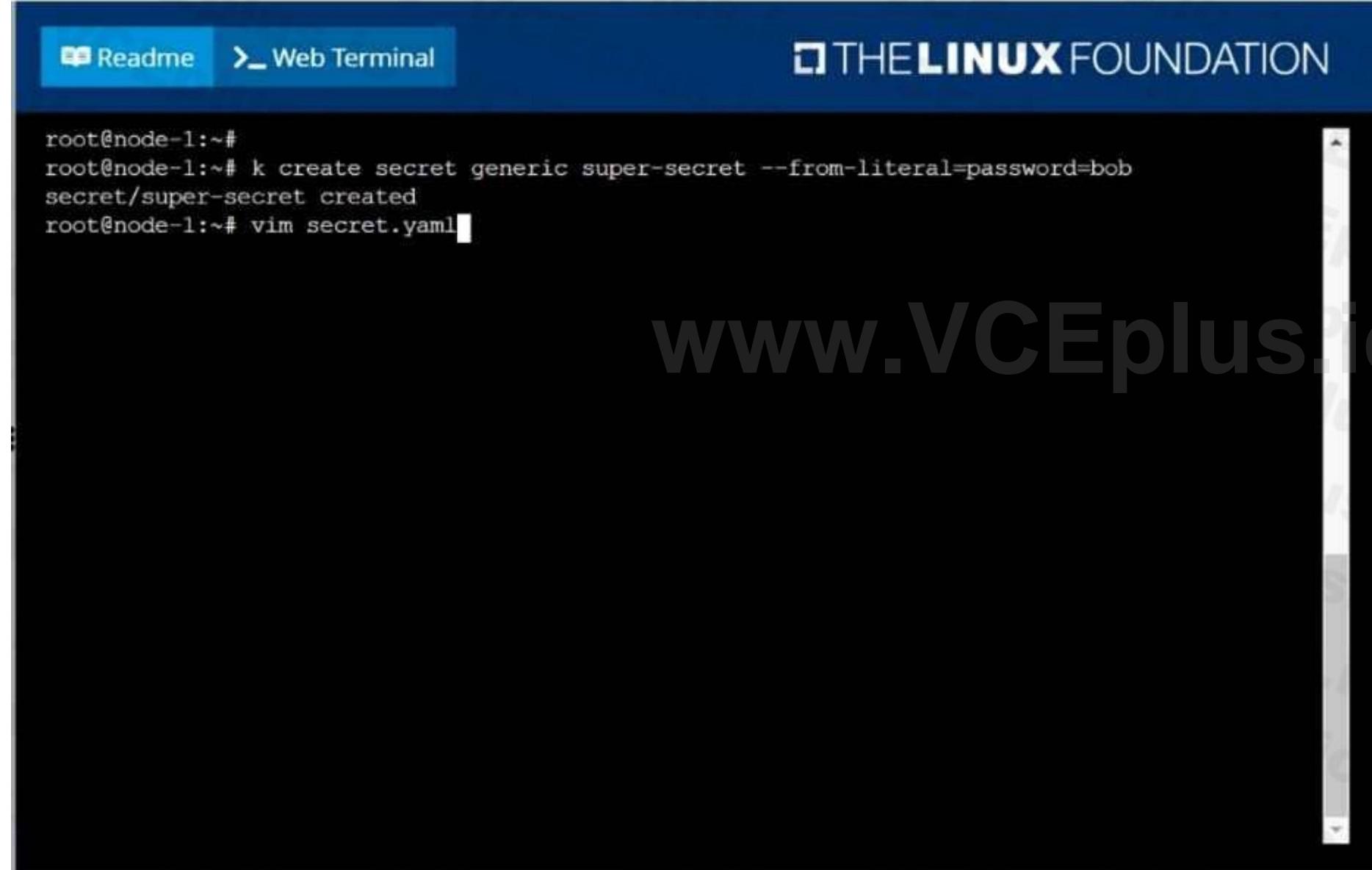
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window with a blue header bar. On the left of the header bar are two buttons: "Readme" and "Web Terminal". To the right of the buttons is the "THE LINUX FOUNDATION" logo. The main area of the terminal is black and contains white text representing a root shell session on a node. The text shows the creation of a Kubernetes secret named "super-secret" with the password "bob". It also shows the command to edit a YAML file named "secret.yaml".

```
root@node-1:~# kubectl create secret generic super-secret --from-literal=password=bob
secret/super-secret created
root@node-1:~# vim secret.yaml
```

www.VCEplus.io

Readme Web Terminal

THE LINUX FOUNDATION

```
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

:w
```

www.VCEplus.io

Readme Web Terminal

THE LINUX FOUNDATION

```
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
```

QUESTION 13

Create a pod as follows:

Name: non-persistent-redis container Image: redis

Volume with name: cache-control

Mount path: /data/redis

The pod should launch in the staging namespace and the volume must not be persistent.

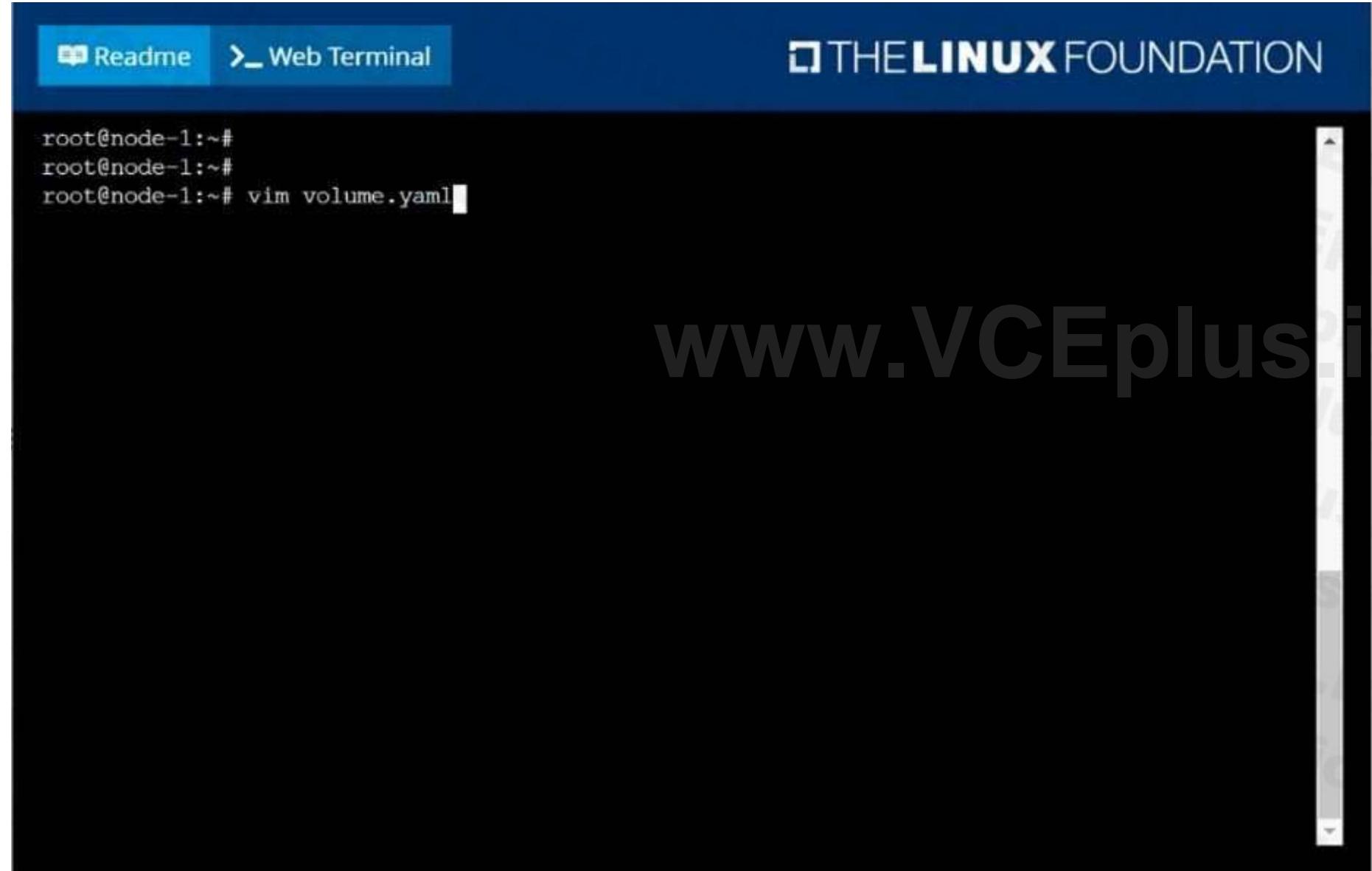
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a web-based terminal interface from 'THE LINUX FOUNDATION'. At the top, there are buttons for 'Readme' and 'Web Terminal'. The terminal window is dark with white text. It displays a root shell on 'node-1':

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

The file 'volume.yaml' is open in the vim editor. The URL 'www.VCEplus.io' is visible at the bottom of the browser window.

Readme > Web Terminal

THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: non-persistent-redis
  namespace: staging
spec:
  containers:
  - name: redis
    image: redis
    volumeMounts:
    - name: cache-control
      mountPath: /data/redis
  volumes:
  - name: cache-control
    emptyDir: {}
```

:w

Readme > Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml
root@node-1:~# k create -f volume.yaml
pod/non-persistent-redis created
root@node-1:~# k get po -n staging
NAME           READY   STATUS    RESTARTS   AGE
non-persistent-redis  1/1     Running   0          6s
root@node-1:~#
```

QUESTION 14

Scale the deployment webserver to 6 pods.

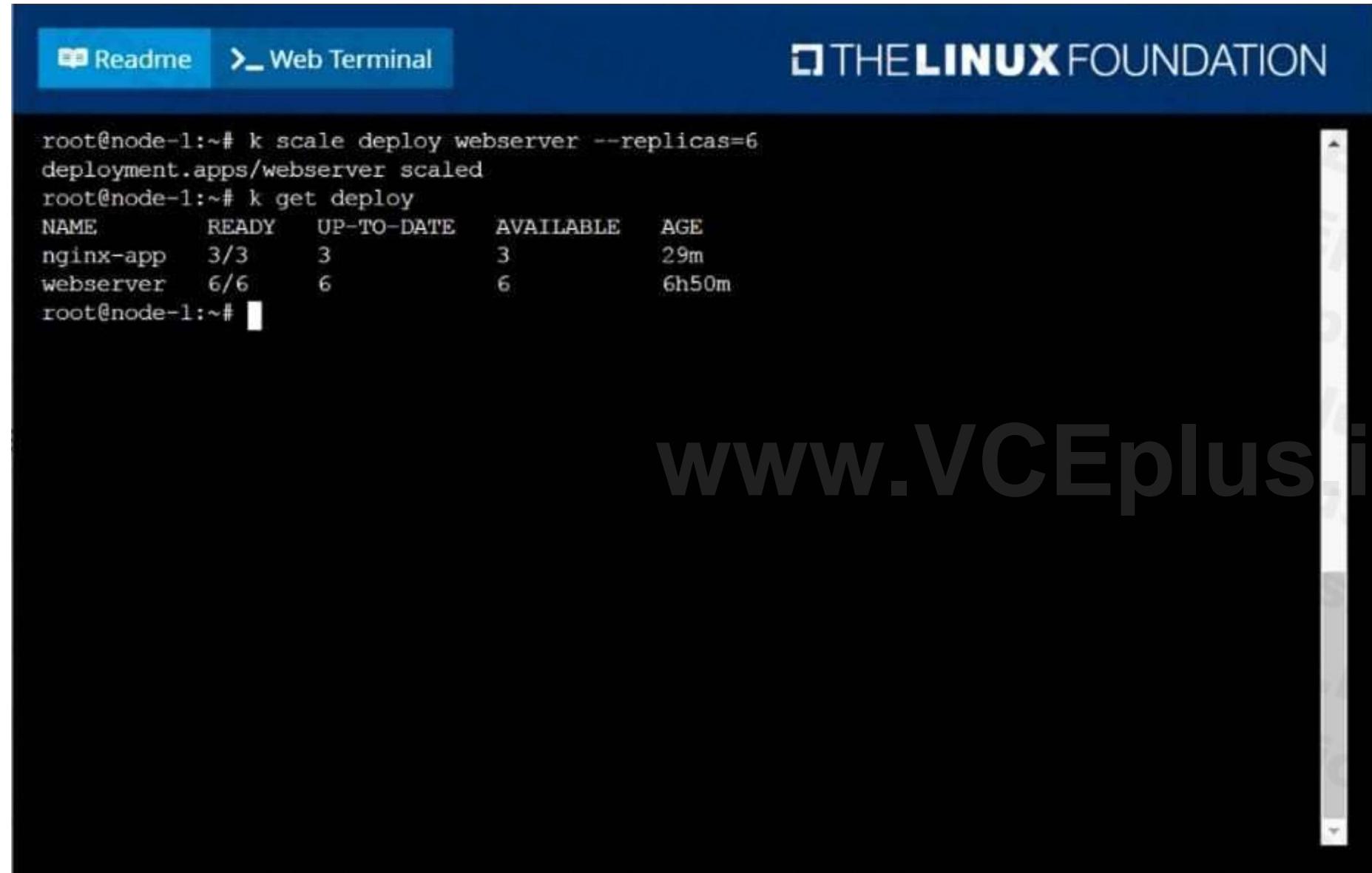
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window with a blue header bar. On the left of the header bar are two buttons: 'Readme' and 'Web Terminal'. To the right of the header bar is the text 'THE LINUX FOUNDATION'. The main area of the terminal contains the following text:

```
root@node-1:~# k scale deploy webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app 3/3     3           3           29m
webserver 6/6     6           6           6h50m
root@node-1:~#
```

A large watermark 'www.VCEplus.io' is overlaid across the center of the terminal window.

QUESTION 15

Check to see how many worker nodes are ready (not including nodes tainted NoSchedule) and write the number to /opt/KUCC00104/kucc00104.txt.

A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

Readme > Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~# k scale deploy webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app   3/3     3           3           29m
webserver   6/6     6           6           6h50m
root@node-1:~#
root@node-1:~# k get nodes
NAME            STATUS   ROLES      AGE    VERSION
k8s-master-0   Ready    master     77d    v1.18.2
k8s-node-0     Ready    <none>    77d    v1.18.2
k8s-node-1     Ready    <none>    77d    v1.18.2
root@node-1:~# vim /opt/KUCC00104/kucc00104.txt
```

www.VCEplus.io

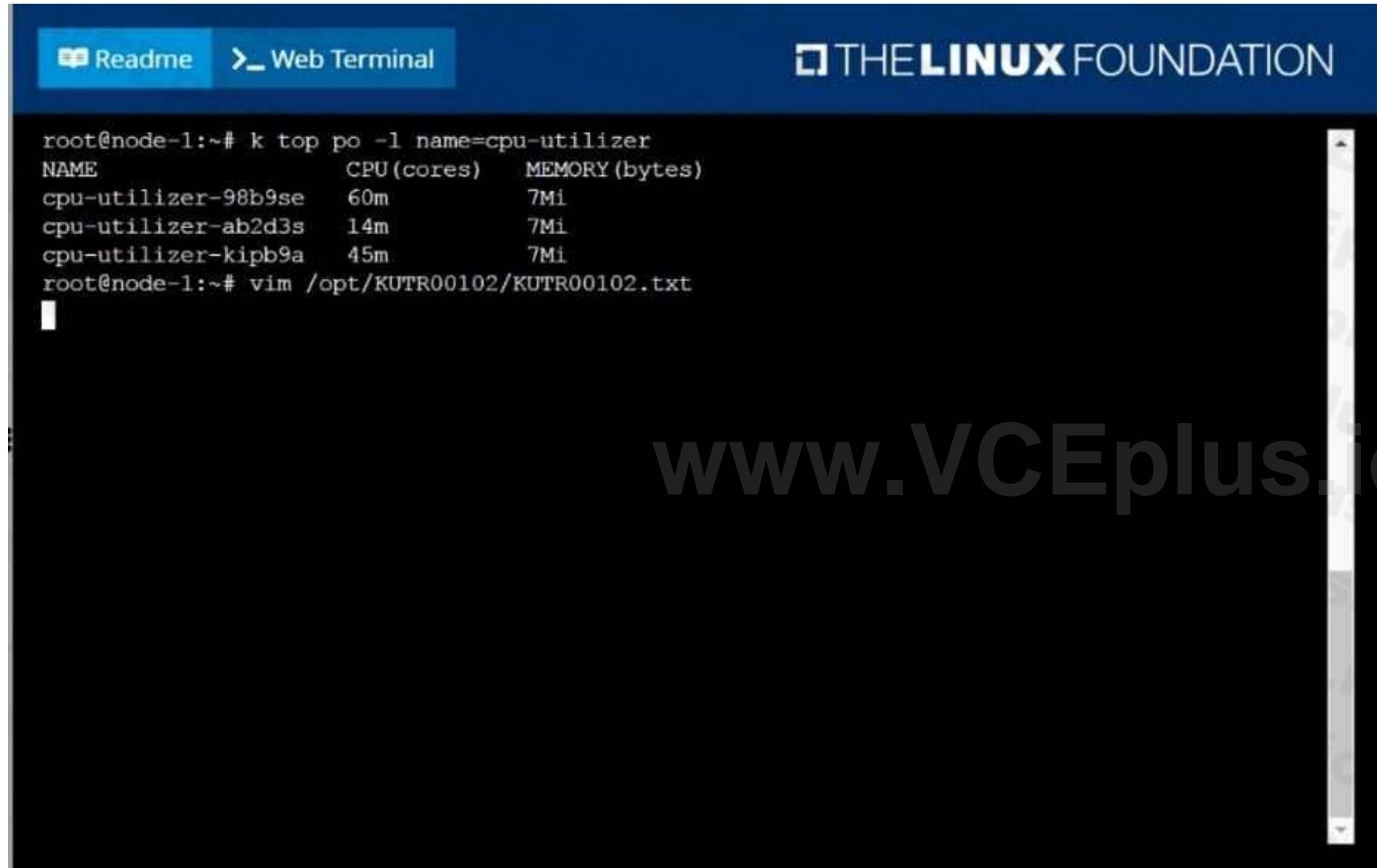
VCEup

QUESTION 16

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. See the solution below.

Correct Answer: A**Section:****Explanation:**

solution



The screenshot shows a terminal window with two tabs: "Readme" and "Web Terminal". The "Web Terminal" tab is active, displaying the output of a command. The command "k top po -l name=cpu-utilizer" lists three pods: "cpu-utilizer-98b9se", "cpu-utilizer-ab2d3s", and "cpu-utilizer-kipb9a". The first pod, "cpu-utilizer-98b9se", has the highest CPU usage at 60m. The terminal then shows the command "root@node-1:~# vim /opt/KUTR00102/KUTR00102.txt", indicating that the user is about to edit the specified file.

```
root@node-1:~# k top po -l name=cpu-utilizer
NAME          CPU(cores)   MEMORY(bytes)
cpu-utilizer-98b9se    60m        7Mi
cpu-utilizer-ab2d3s    14m        7Mi
cpu-utilizer-kipb9a    45m        7Mi
root@node-1:~# vim /opt/KUTR00102/KUTR00102.txt
```

The image is a screenshot of a GitHub repository page. At the top, there are two blue buttons: 'Readme' and 'Web Terminal'. Below these buttons, the repository name 'cpu-utilizer-98b9se' is displayed. The main content area is very dark, almost black, with only a few small, faint white characters visible on the left side, which appear to be part of a terminal session. A large, semi-transparent watermark with the URL 'www.VCEplus.io' is overlaid across the bottom half of the page. The overall appearance is that of a GitHub interface with a dark theme.

QUESTION 17

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. See the solution below.

Correct Answer: A

Section: Five

Explanation: Solution

Solution:

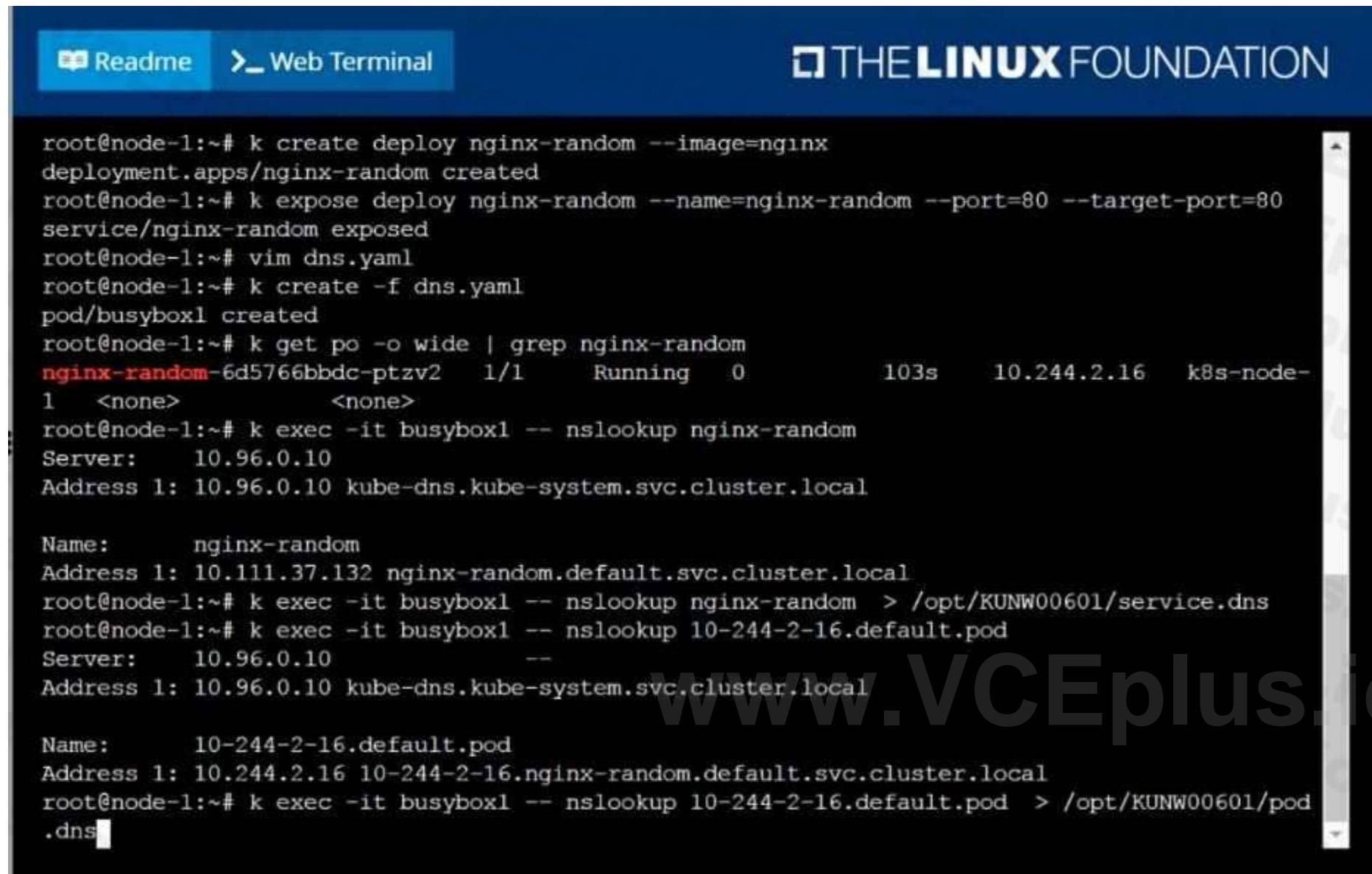
Readme Web Terminal THE LINUX FOUNDATION

```
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
```

www.VCEplus.io

Readme Web Terminal THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox1
  labels:
    name: busybox
spec:
  containers:
  - image: busybox:1.28
    command:
      sleep "3600"
    name: busybox
```



```
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
```

QUESTION 18

Create a snapshot of the etcd instance running at <https://127.0.0.1:2379>, saving the snapshot to the file path /srv/data/etcd-snapshot.db.

The following TLS certificates/key are supplied for connecting to the server with etcdctl:

CA certificate: /opt/KUCM00302/ca.crt

Client certificate: /opt/KUCM00302/etcd-client.crt

Client key: /opt/KUCM00302/etcd-client.key

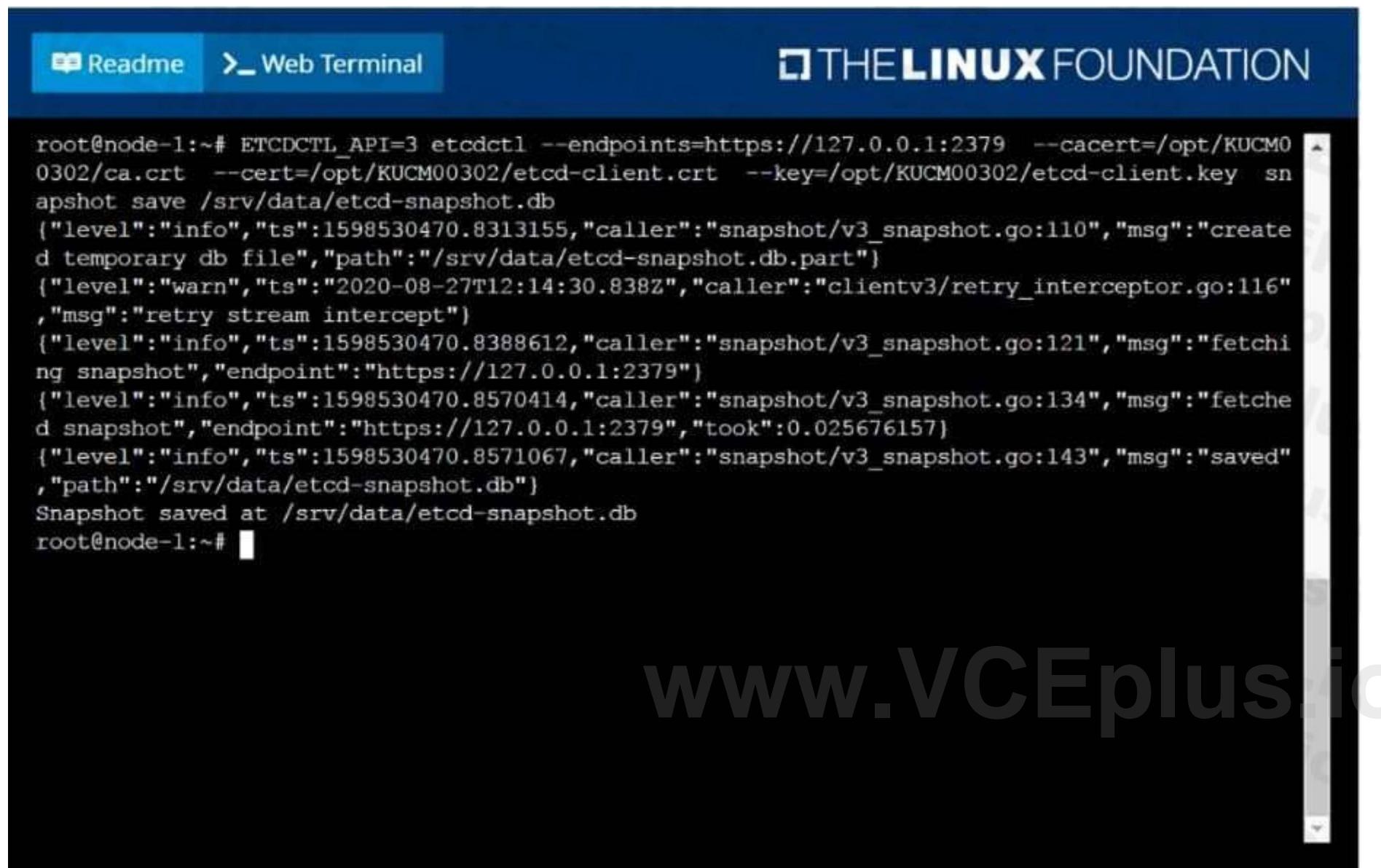
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window with two tabs: "Readme" and "Web Terminal". The "Web Terminal" tab is active, displaying the output of a command. The command is:

```
root@node-1:~# ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/opt/KUCM00302/ca.crt --cert=/opt/KUCM00302/etcd-client.crt --key=/opt/KUCM00302/etcd-client.key snapshot save /srv/data/etcd-snapshot.db
```

The output shows the process of creating a temporary db file, fetching a snapshot, and saving it to /srv/data/etcd-snapshot.db. The log entries include:

- {"level": "info", "ts": 1598530470.8313155, "caller": "snapshot/v3_snapshot.go:110", "msg": "create temporary db file", "path": "/srv/data/etcd-snapshot.db.part"}
- {"level": "warn", "ts": "2020-08-27T12:14:30.838Z", "caller": "clientv3/retry_interceptor.go:116", "msg": "retry stream intercept"}
- {"level": "info", "ts": 1598530470.8388612, "caller": "snapshot/v3_snapshot.go:121", "msg": "fetching snapshot", "endpoint": "https://127.0.0.1:2379"}
- {"level": "info", "ts": 1598530470.8570414, "caller": "snapshot/v3_snapshot.go:134", "msg": "fetched snapshot", "endpoint": "https://127.0.0.1:2379", "took": 0.025676157}
- {"level": "info", "ts": 1598530470.8571067, "caller": "snapshot/v3_snapshot.go:143", "msg": "saved", "path": "/srv/data/etcd-snapshot.db"}

Snapshot saved at /srv/data/etcd-snapshot.db

root@node-1:~#

www.VCEplus.io

QUESTION 19

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

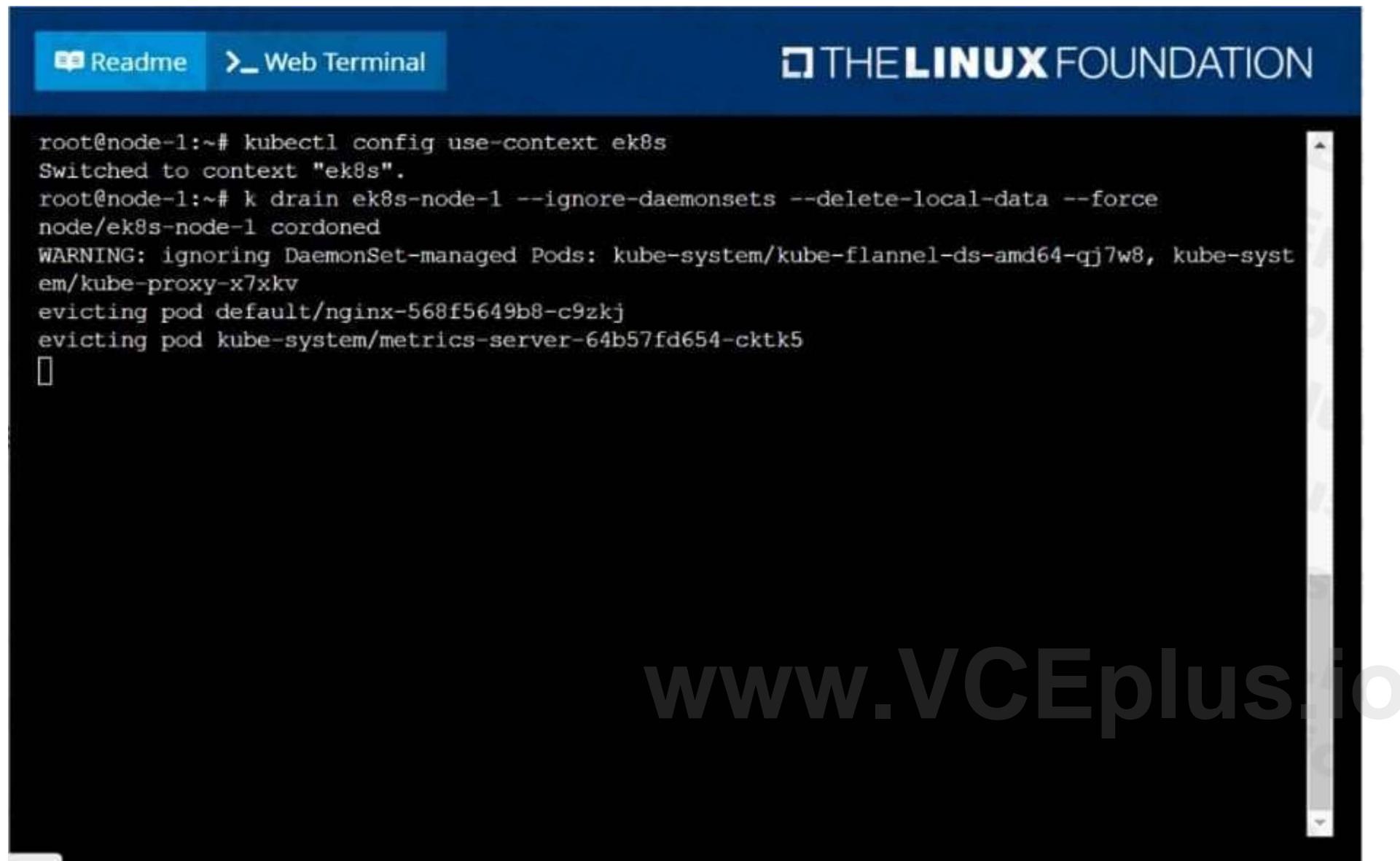
A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution



The screenshot shows a terminal window titled "THE LINUX FOUNDATION". The terminal content is as follows:

```
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
[]
```

QUESTION 20

A Kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a Ready state, ensuring that any changes are made permanent.

You can ssh to the failed node using:

[student@node-1] \$ | ssh Wk8s-node-0

You can assume elevated privileges on the node with the following command:

[student@w8ks-node-0] \$ | sudo -i

A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

Readme Web Terminal THE LINUX FOUNDATION

```
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# k get nodes
NAME        STATUS   ROLES    AGE     VERSION
wk8s-master-0 Ready    master   77d    v1.18.2
wk8s-node-0  NotReady <none>   77d    v1.18.2
wk8s-node-1  Ready    <none>   77d    v1.18.2
root@node-1:~# ssh wk8s-node-0
[
```

www.VCEplus.io

Readme Web Terminal THE LINUX FOUNDATION

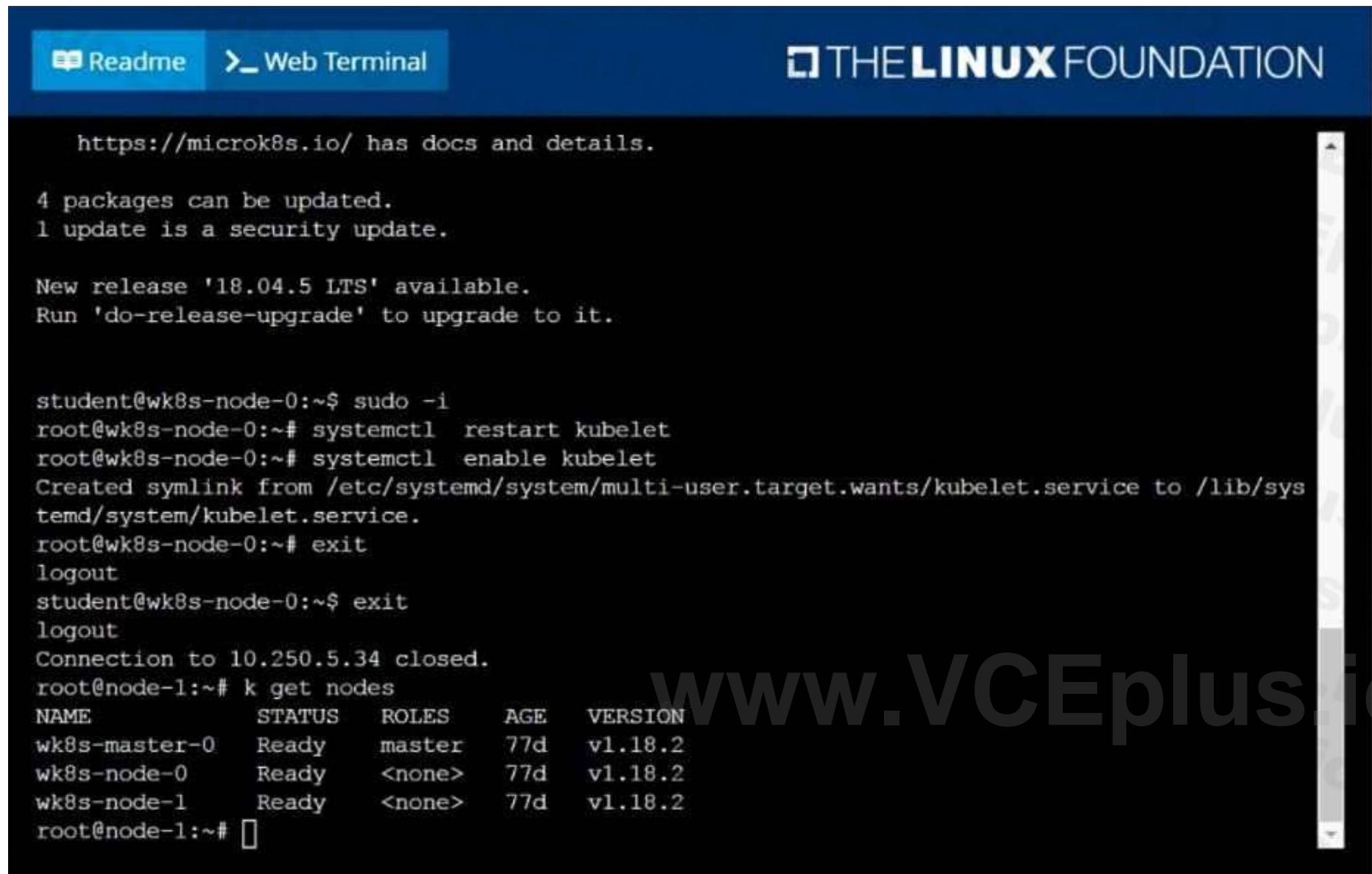
```
wk8s-node-0  NotReady  <none>  77d  v1.18.2
wk8s-node-1  Ready     <none>  77d  v1.18.2
root@node-1:~# ssh wk8s-node-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

 * 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
```

VCEUP VCEplus IT Certification Exams - Questions & Answers | VCEplus.io <https://microk8s.io/> has docs and details.

VCEplus VCEUP



The screenshot shows a terminal window with the following content:

```
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@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/sys
temd/system/kubelet.service.
root@wk8s-node-0:~# exit
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.5.34 closed.
root@node-1:~# k get nodes
NAME      STATUS  ROLES   AGE    VERSION
wk8s-master-0  Ready   master  77d   v1.18.2
wk8s-node-0    Ready   <none>  77d   v1.18.2
wk8s-node-1    Ready   <none>  77d   v1.18.2
root@node-1:~# 
```

QUESTION 21

Configure the kubelet systemd- managed service, on the node labelled with name=wk8s-node-1, to launch a pod containing a single container of Image httpd named webtool automatically. Any spec files required should be placed in the /etc/kubernetes/manifests directory on the node.

You can ssh to the appropriate node using:

```
[student@node-1] $ ssh wk8s-node-1
```

You can assume elevated privileges on the node with the following command:

```
[student@wk8s-node-1] $ | sudo -i
```

A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

Readme > Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~#
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# ssh wk8s-node-1
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@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
```

Readme > Web Terminal

THE LINUX FOUNDATION

```
clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
failureFrequency: 0s
healthBindAddress: 127.0.0.1
healthPort: 10949
```

Readme > Web Terminal

THE LINUX FOUNDATION

```
root@node-1:~# ssh wk8s-node-1
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.

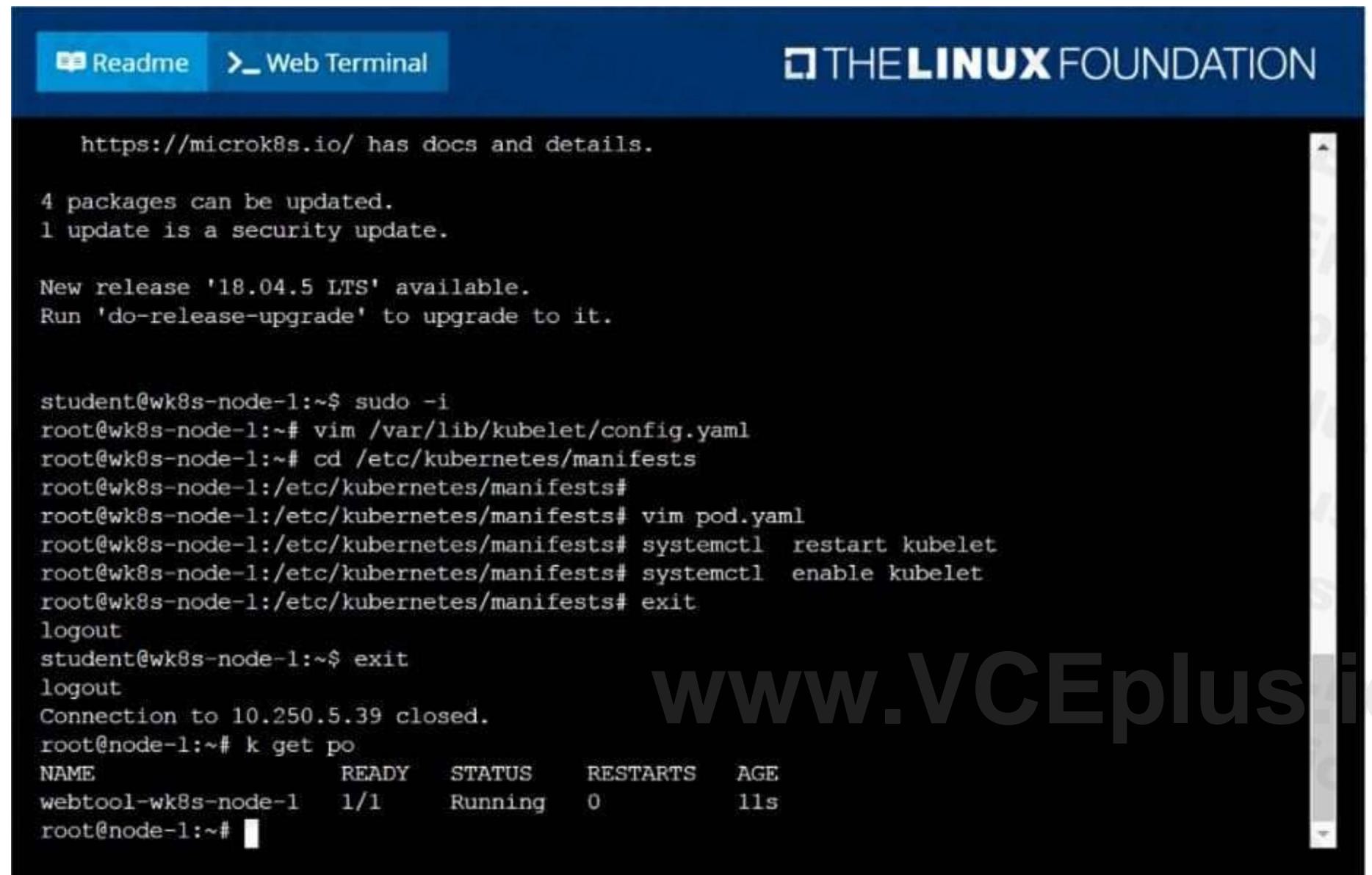
student@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests#
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml
```

www.VCEplus.io

Readme > Web Terminal

THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  name: webtool
spec:
  containers:
  - name: webtool
    image: httpd
```



The screenshot shows a terminal window with the following content:

```
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@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests#
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml
root@wk8s-node-1:/etc/kubernetes/manifests# systemctl restart kubelet
root@wk8s-node-1:/etc/kubernetes/manifests# systemctl enable kubelet
root@wk8s-node-1:/etc/kubernetes/manifests# exit
logout
student@wk8s-node-1:~$ exit
logout
Connection to 10.250.5.39 closed.
root@node-1:~# k get po
NAME        READY   STATUS    RESTARTS   AGE
webtool-wk8s-node-1   1/1     Running   0          11s
root@node-1:~#
```

QUESTION 22

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. See the solution below.

Correct Answer: A

Section:

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.

QUESTION 23

Given a partially-functioning Kubernetes cluster, identify symptoms of failure on the cluster.

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.

You can ssh to the relevant 1 nodes (bk8s-master-0 or bk8s-node-0) using:

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

You can assume elevated privileges on any node in the cluster with the following command:

[student@nodename] \$ | sudo -i

A. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

www.VCEplus.io

Readme ➔ Web Terminal

THE LINUX FOUNDATION

```
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
```

Readme ➔ Web Terminal

THE LINUX FOUNDATION

```
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
```

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:~#
```

QUESTION 24

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. See the solution below.

Correct Answer: A

Section:

Explanation:

solution

Persistent Volume

A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a clusterlevel 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 tying 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.
```

www.VCEplus.io

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /srv/app-data
  storageClassName: shared
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

www.VCEplus.io

“app-data.yaml” 12L, 194C

2. Save the file and create the persistent volume.

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl create -f pv.yaml
persistentvolume/pv created
```

3. View the persistent volume.

| (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: PersistentVolume

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

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.

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: v1

kind: Pod

metadata:

creationTimestamp: null

name: app-data

spec:

volumes:

- name:configpvc

```
persistenVolumeClaim:  
claimName: app-data  
containers:  
- image: nginx  
name: app  
volumeMounts:  
- mountPath: "/srv/app-data"  
name: configpvc
```

QUESTION 25

Create a namespace called 'development' and a pod with image nginx called nginx on this namespace.

A. See the solution below.

Correct Answer: A**Section:****Explanation:**

kubectl create namespace development

kubectl run nginx --image=nginx --restart=Never -n development

QUESTION 26

Create a nginx pod with label env=test in engineering namespace

A. See the solution below.

Correct Answer: A**Section:****Explanation:**

kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dryrun
-o yaml > nginx-pod.yaml

kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dryrun
-o yaml | kubectl create -n engineering -f -

YAML File:

```
apiVersion: v1  
kind: Pod  
metadata:  
name: nginx  
namespace: engineering  
labels:  
env: test  
spec:  
containers:  
- name: nginx  
image: nginx  
imagePullPolicy: IfNotPresent  
restartPolicy: Never  
kubectl create -f nginx-pod.yaml
```

QUESTION 27

Get list of all pods in all namespaces and write it to file "/opt/pods-list.yaml"

A. See the solution below.

Correct Answer: A

Section:

Explanation:

kubectl get po --all-namespaces > /opt/pods-list.yaml

QUESTION 28

Create a pod with image nginx called nginx and allow traffic on port 80

A. See the solution below.

Correct Answer: A

Section:

Explanation:

kubectl run nginx --image=nginx --restart=Never --port=80

QUESTION 29

Create a busybox pod that runs the command "env" and save the output to "envpod" file

A. See the solution below.

Correct Answer: A

Section:

Explanation:

kubectl run busybox --image=busybox --restart=Never --rm -it -- env > envpod.yaml

QUESTION 30

List pod logs named "frontend" and search for the pattern "started" and write it to a file "/opt/errorlogs"

A. See the solution below.

Correct Answer: A

Section:

Explanation:

Kubectl logs frontend | grep -i "started" > /opt/error-logs

www.VCEplus.io