Question #:1

Monitor the logs of pod foo and:

- **Extract log lines correspondingto error unable-to-access-website**
- **Write them to/opt/KULM00201/foo**

Set configuration context:

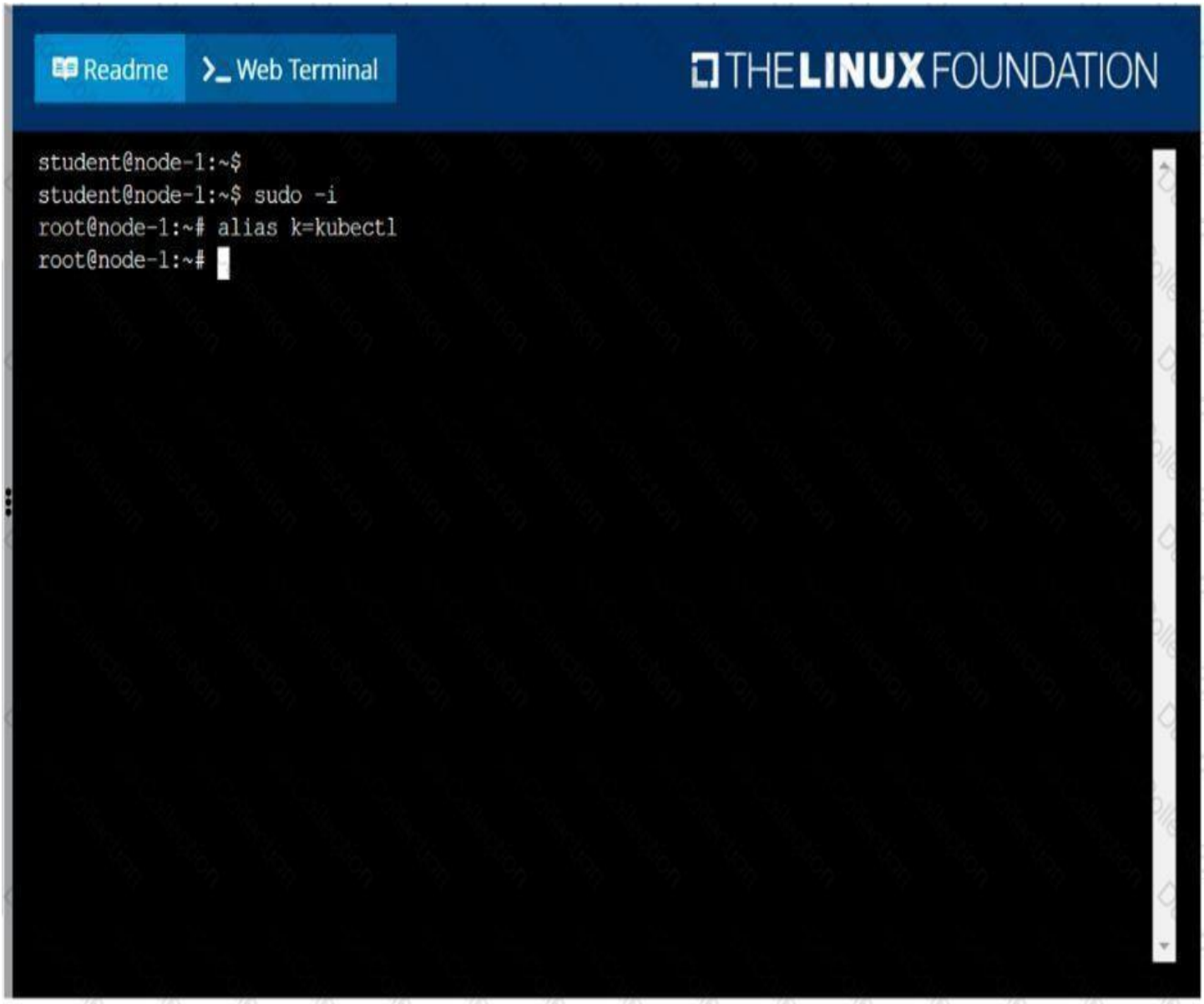
```
[student@node-1] $ kube
ctl config use-context
k8s
```

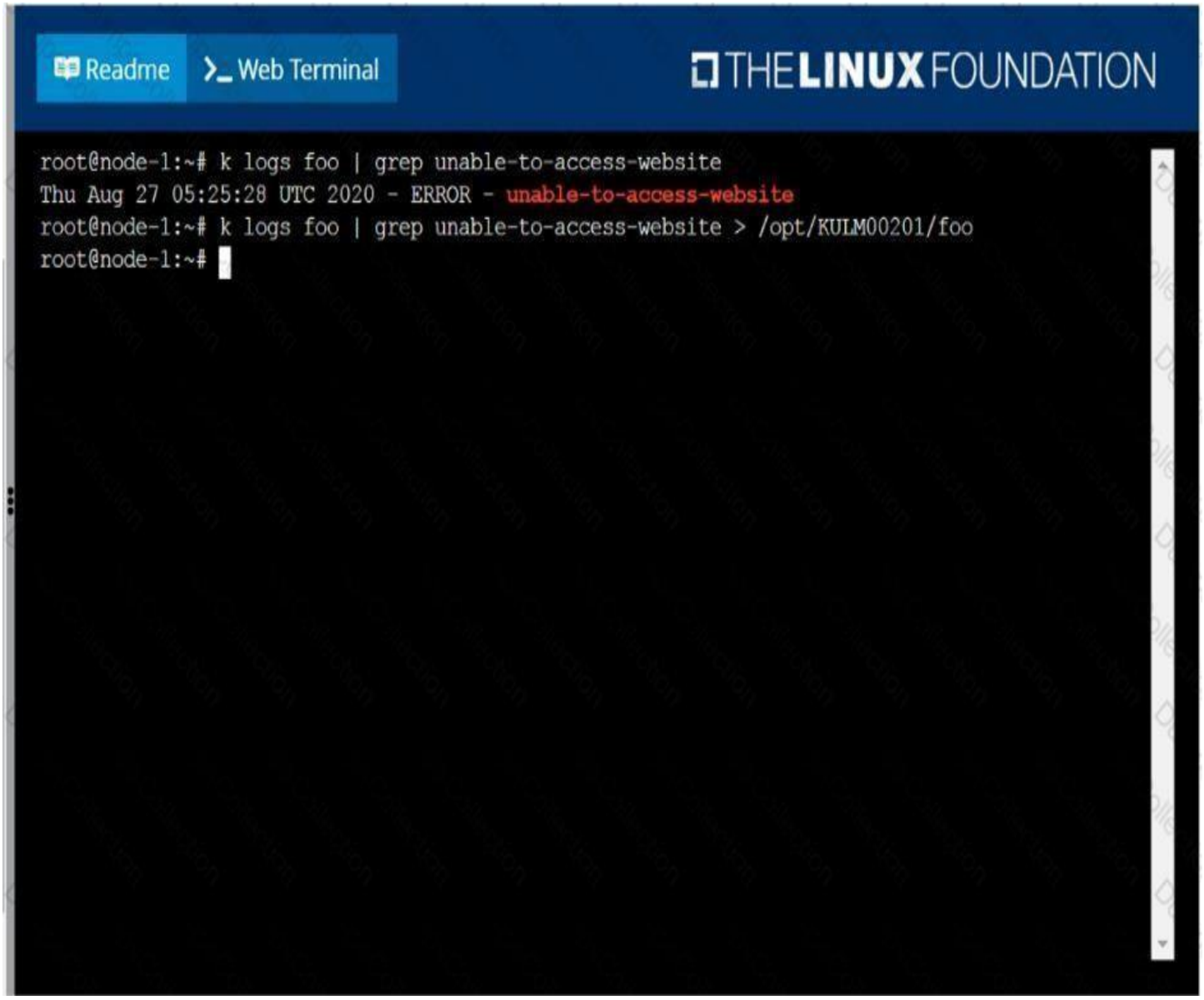
See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\1 B.JPG


Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$
student@node-1:~$ sudo -i
root@node-1:~# alias k=kubectl
root@node-1:~#



Question #:2

Get IP address of the pod – “nginx-dev”

See the solution below.

Explanation

Kubect1 get po -o wide

Using JsonPath

kubect1 get pods -o=jsonpath='{range

items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'

Question #:3

Create a deployment spec file thatwill:

- 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 thatyou produced during this task.

See the solution below.

Explanation

solution

Readme Web Terminal THE LINUX FOUNDATION

```
root@node-1:~# k create deploy kual00201  --image=nginx --dry-run=client -o yaml > /opt/KUAL
00201/spec_deployment.yaml
root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml
```

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:
      labels:
        app_runtime_stage: dev
    spec:
      containers:
      - image: nginx
        name: nginx
~
~
~
~
~
"/opt/KUAL00201/spec_deployment.yaml" 19L, 320C written
```

Question #:4

List pod logs named “frontend” and search for the pattern “started” and write it to a file “/opt/error-logs”

See the solution below.

Explanation

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

Question #:5

Create a Kubernetes secret asfollows:

- Name: super-secret
- password: bob

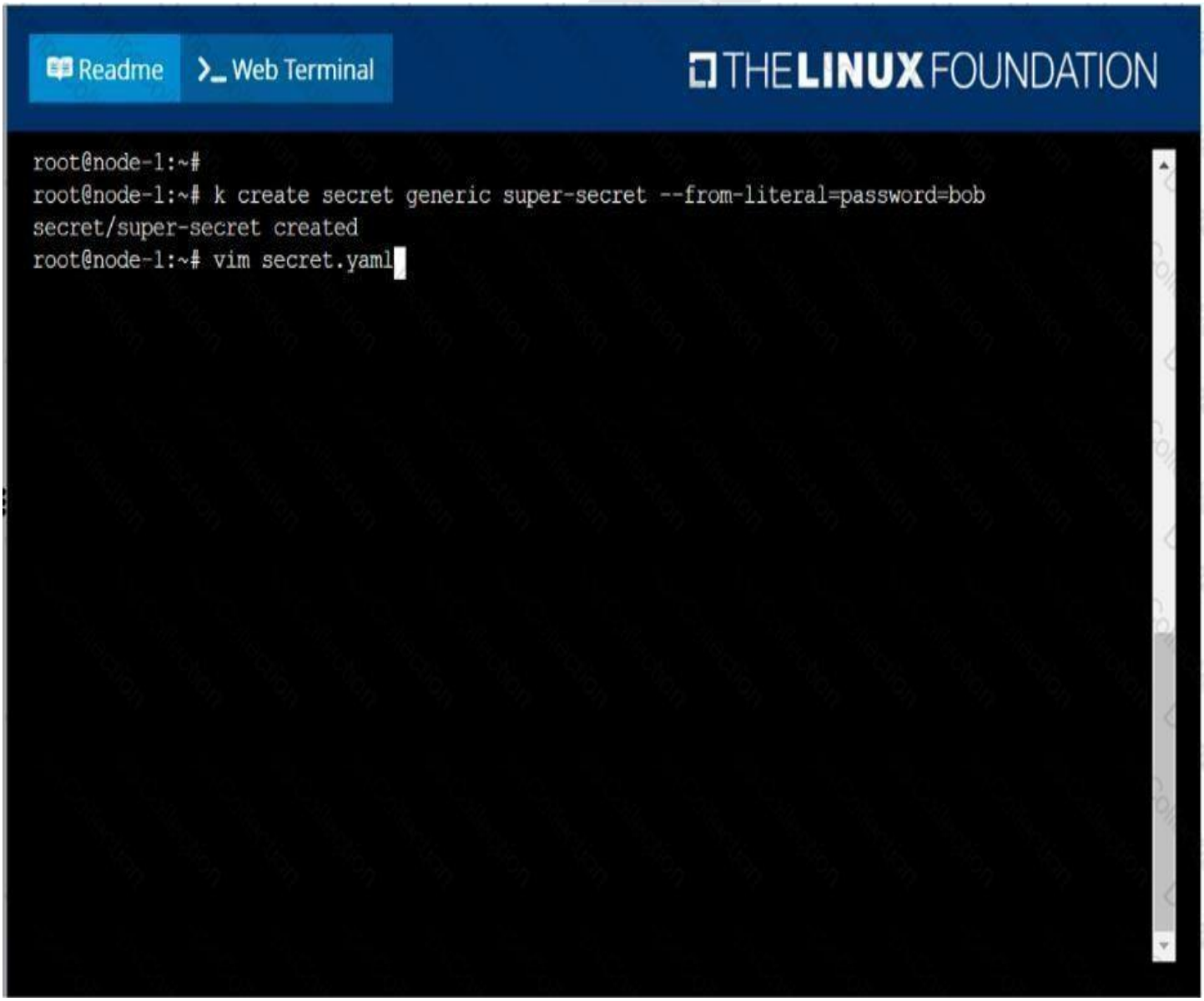
Create a pod named*pod-secrets-via-file*, using the*redis*Image, which mounts a secret named*super-secret*at */secrets*.

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

See the solution below.

Explanation

solution



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
```

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
front-end                   1/1     Running   0          6h27m
hungry-bear                 1/1     Running   0          36m
kucc8                       3/3     Running   0          34m
nginx-app-848cfcf495-9prjh  1/1     Running   0          19m
nginx-app-848cfcf495-gl2kh  1/1     Running   0          19m
nginx-app-848cfcf495-pg2c8  1/1     Running   0          19m
nginx-kusc00101             1/1     Running   0          26m
pod-secrets-via-env         1/1     Running   0          4s
pod-secrets-via-file        1/1     Running   0          106s
webserver-84c55967f4-qzjcv  1/1     Running   0          6h43m
webserver-84c55967f4-t4791  1/1     Running   0          6h43m
root@node-1:~#
```

Question #:6

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

See the solution below.

Explanation

kubectlrun nginx --image=nginx --restart=Never --port=80

Question #:7

List the nginx pod with custom columns POD_NAME and POD_STATUS

See the solution below.

Explanation

kubectl get po -o=custom-columns="POD_NAME:.metadata.name,

POD_STATUS:.status.containerStatuses[].state"

Question #:8

Create a pod as follows:

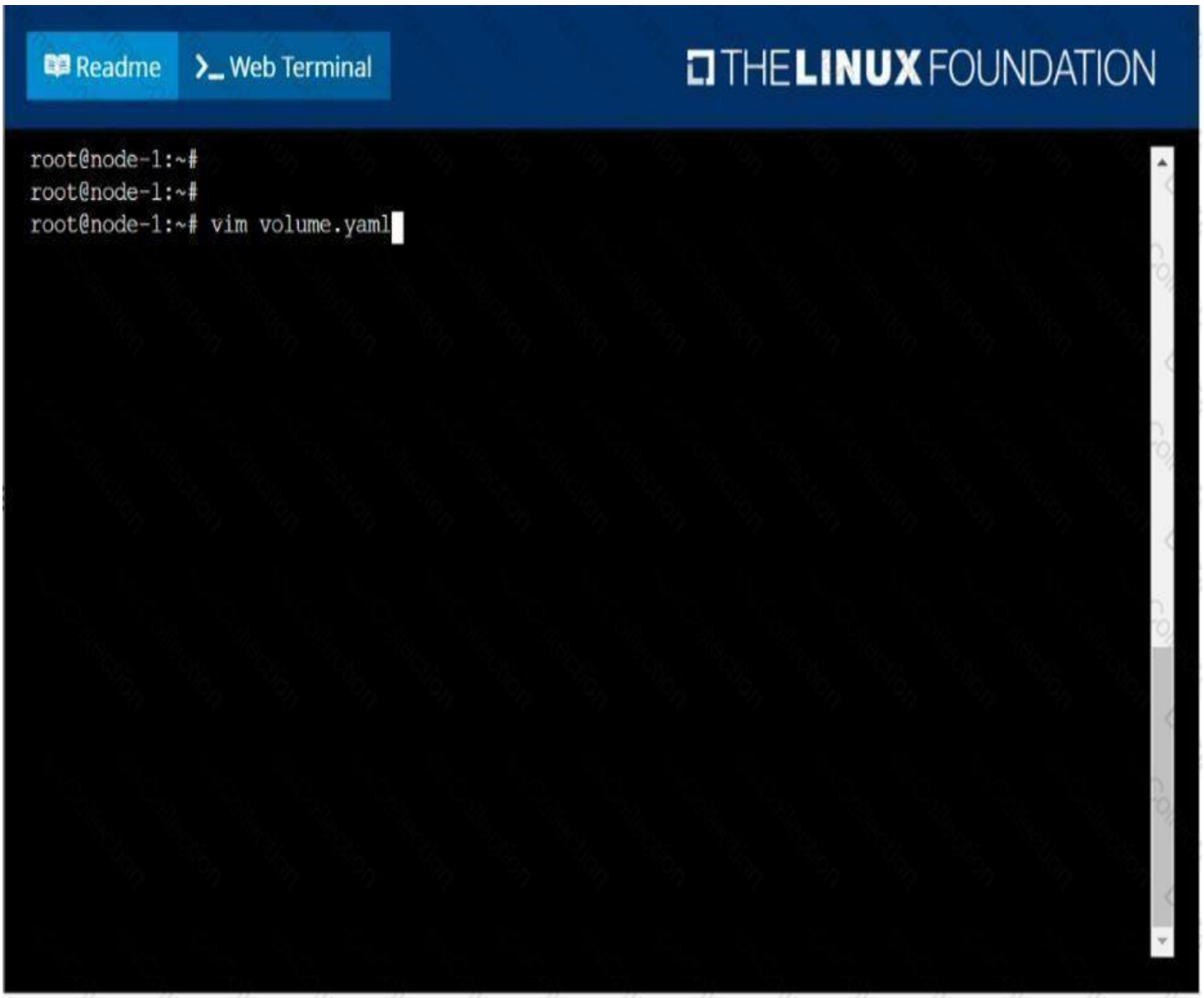
- Name:*non-persistent-redis*
- container Image:*redis*
- Volume with name:*cache-control*
- Mount path:*/data/redis*

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

See the solution below.

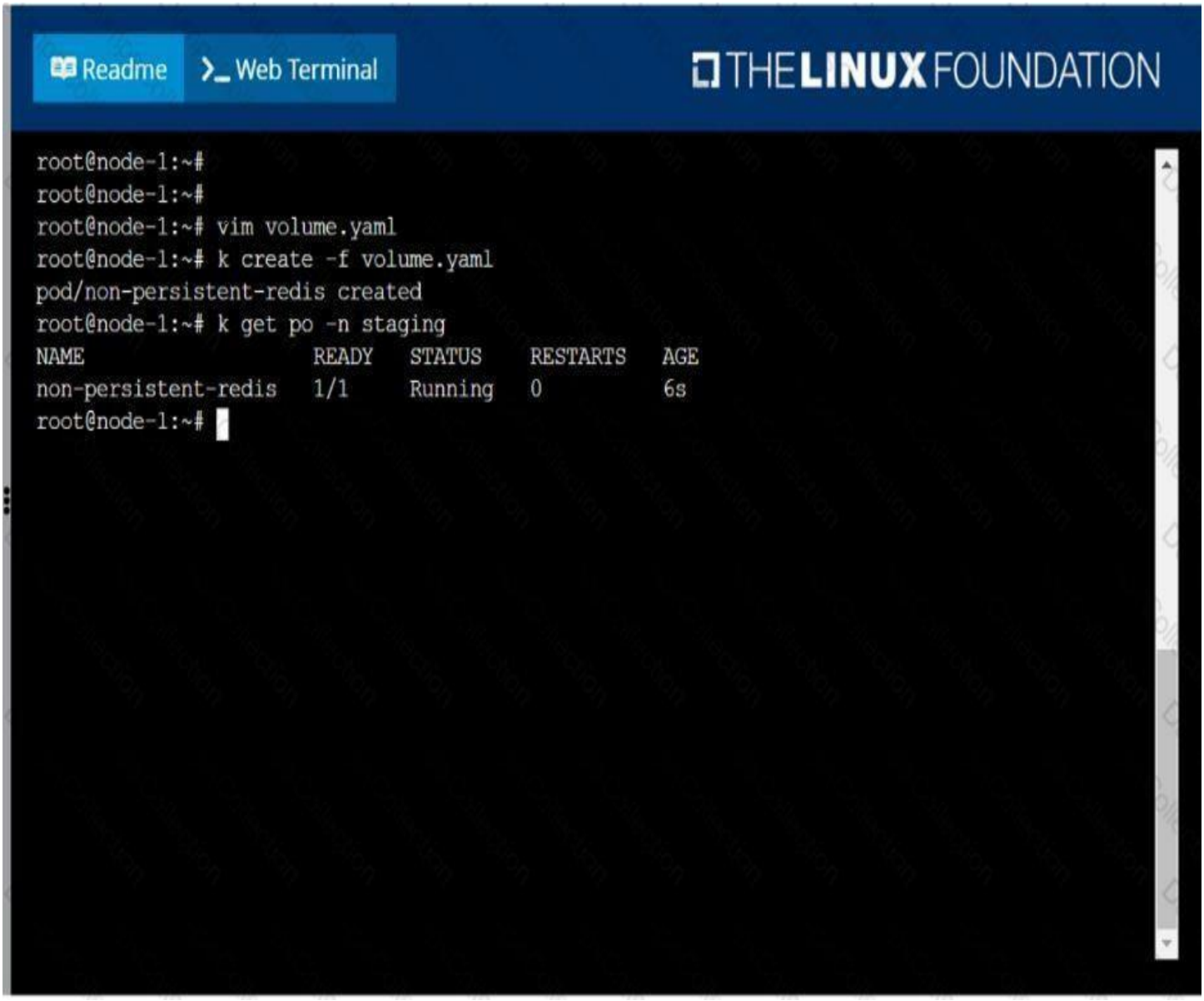
Explanation

solution


Readme
Web Terminal
THE LINUX FOUNDATION
root@node-1:~#
root@node-1:~#
root@node-1:~# vim volume.yaml

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
```



Question #:9

Create a deployment as follows:

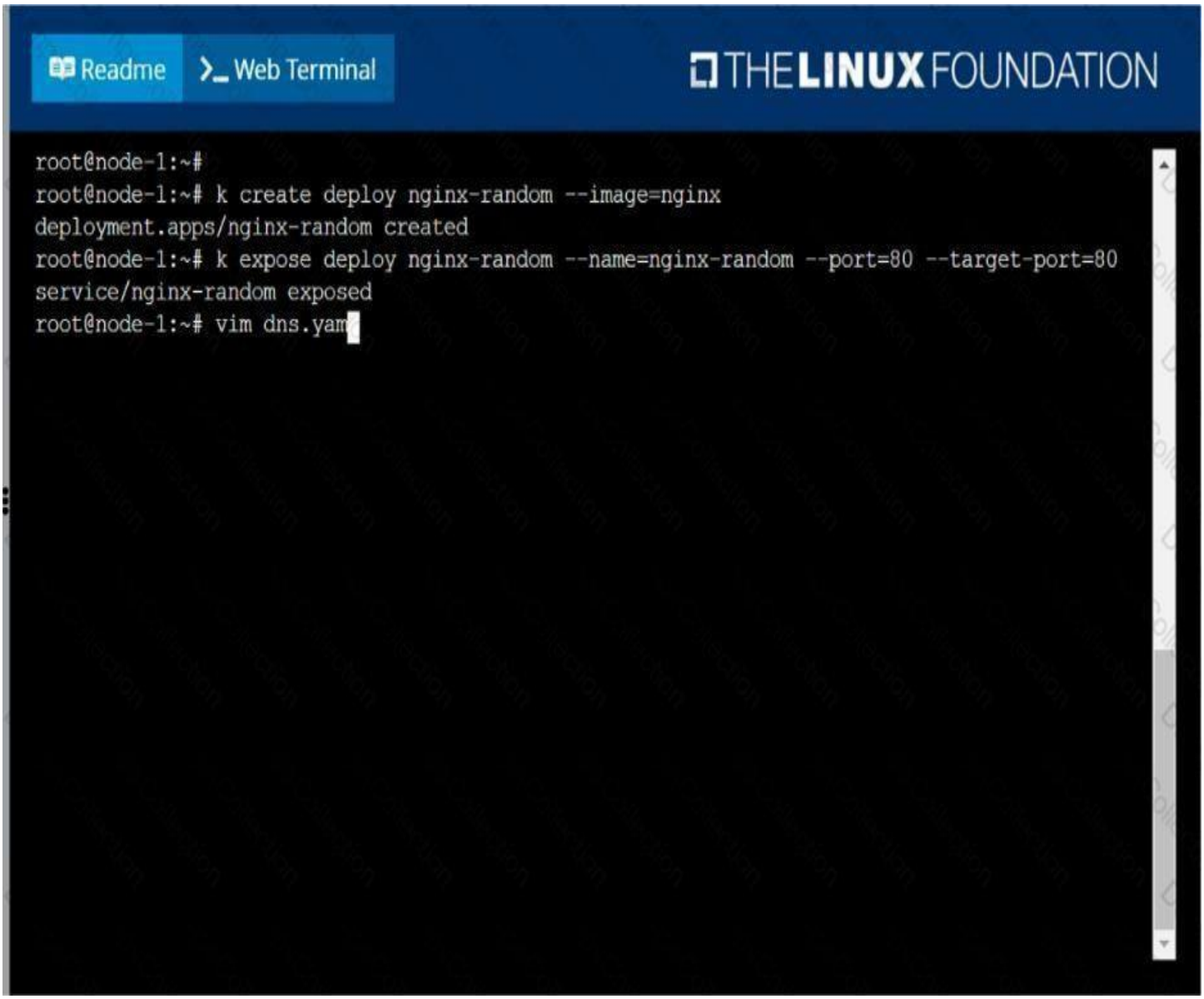
- Name:*nginx-random*
- Exposed via a service*nginx-random*
- Ensure that the service & podare accessible via theirrespective DNS records
- The container(s) within anypod(s) running as a part of thisdeployment should use the*nginx*Image

Next, use the utility*nslookup*to lookup the DNS records of the service &pod and write the output to */opt/KUNW00601/service.dns*and*/opt/KUNW00601/pod.dns*respectively.

See the solution below.

Explanation

Solution:



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 #:10

List all the pods sorted by name

See the solution below.

Explanation

kubectl get pods --sort-by=.metadata.name

Question #:11

Create a deployment as follows:

- Name:*nginx-app*
- Using container*nginx*withversion 1.11.10-alpine
- The deployment should contain*3*replicas

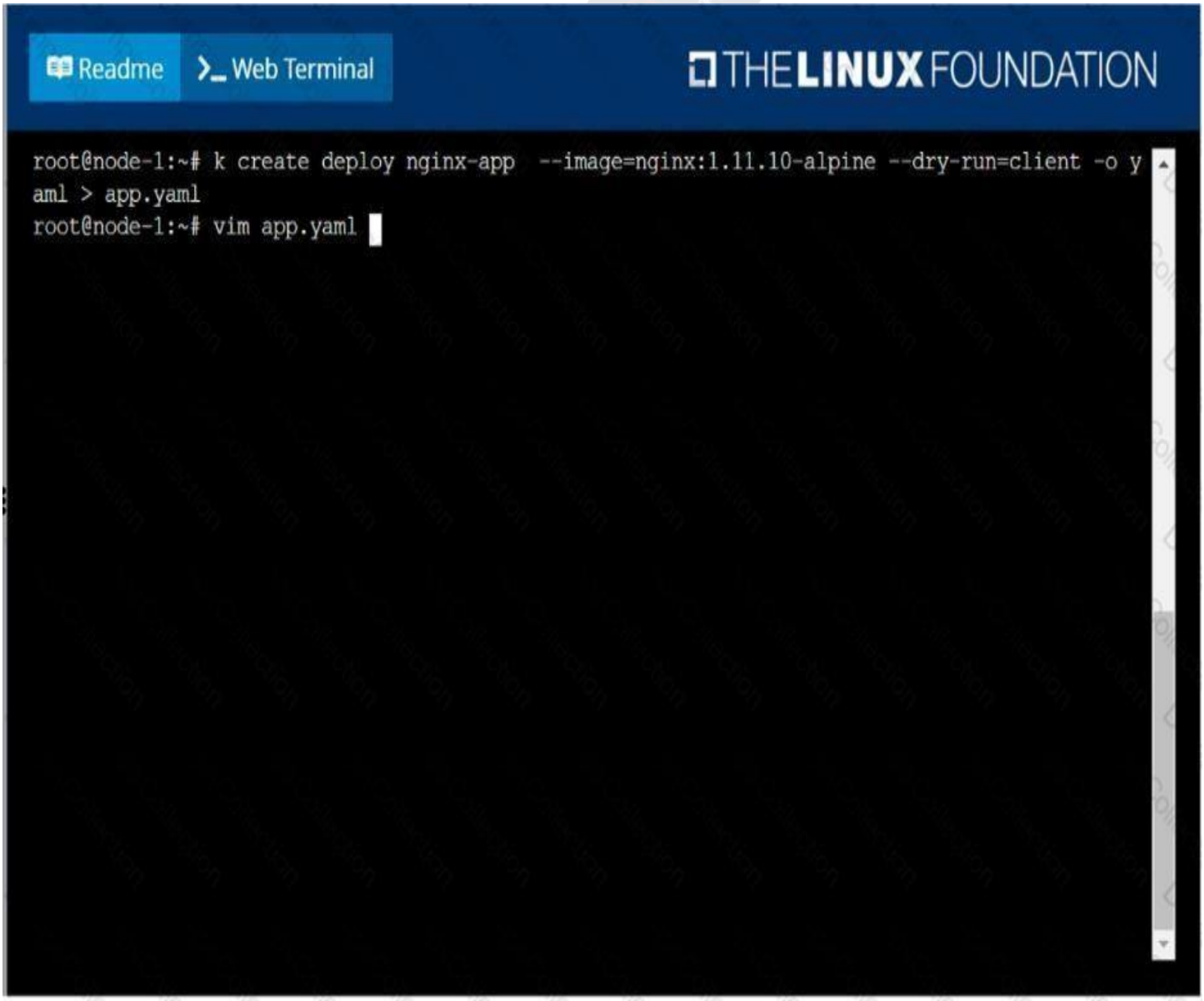
Next, deploy the application with newversion*1.11.13-alpine*, byperforming a rolling update.

Finally, rollback that update to theprevious version*1.11.10-alpine*.

See the solution below.

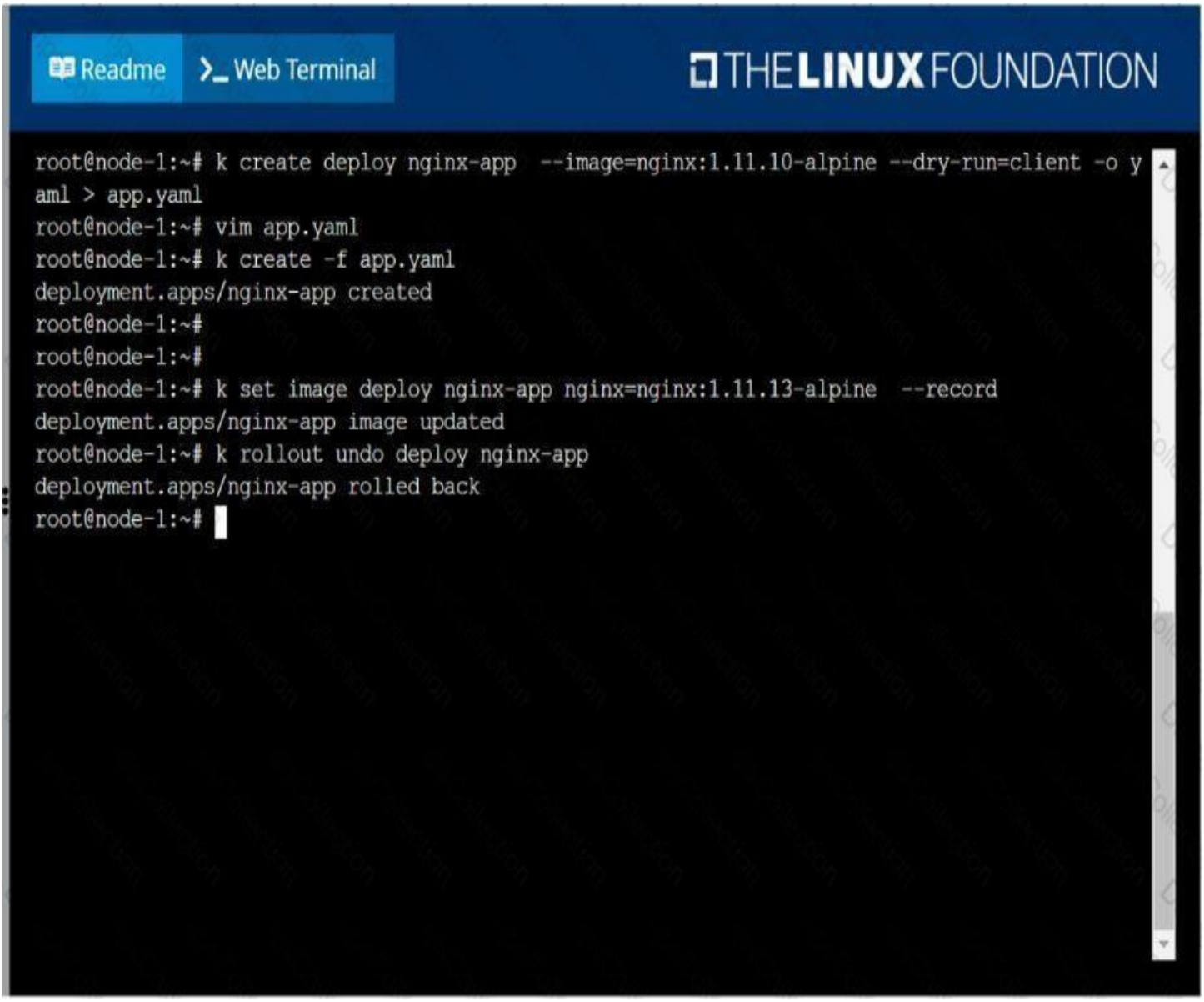
Explanation

solution



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
```



Question #:12

From the pod label*name=cpu-utilizer*, find podsrunning high CPU workloads and

write the name of the pod consumingmost CPU to the*file/opt/KUTR00102/KUTR00102.txt*(which already exists).

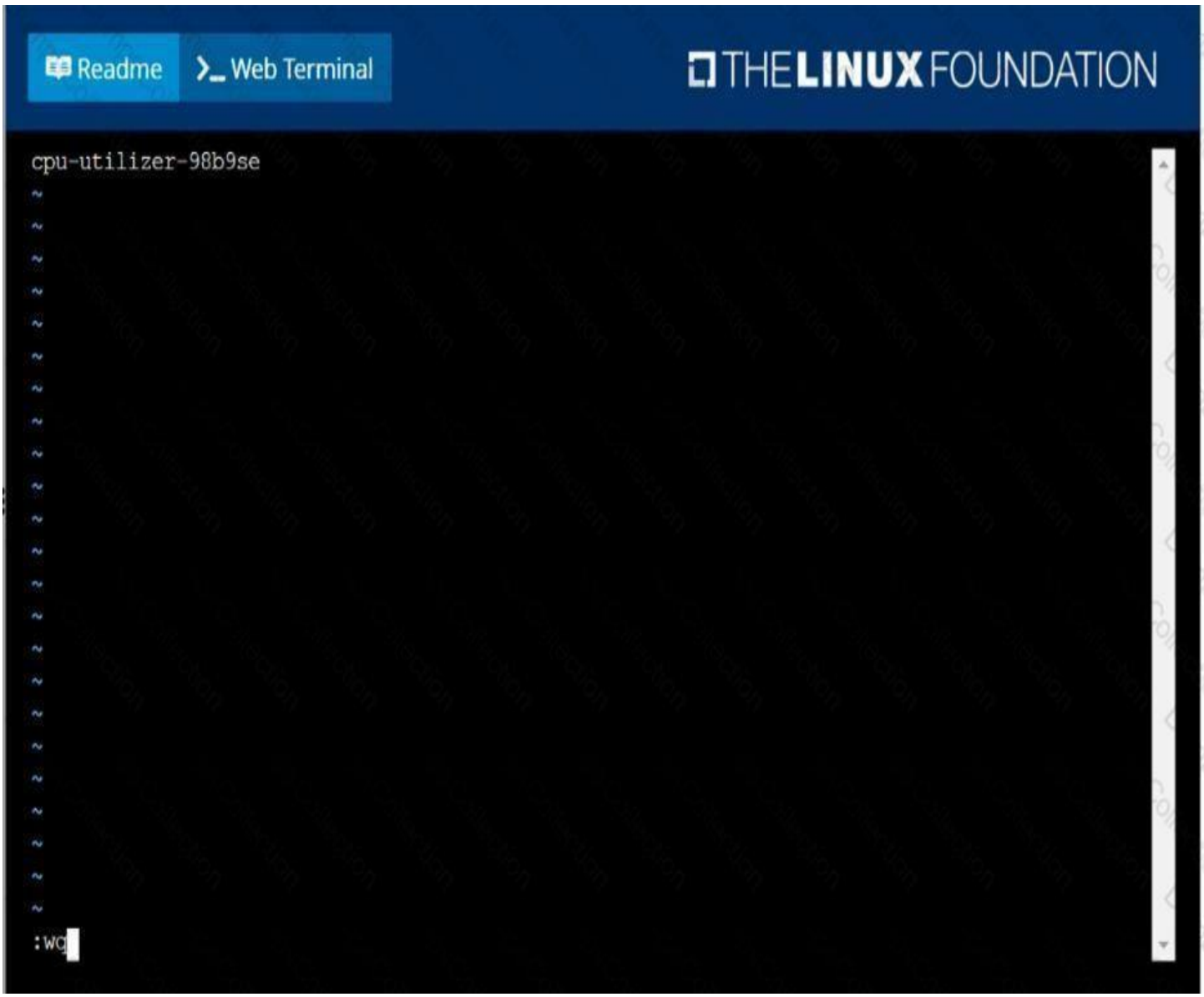
See the solution below.

Explanation

solution

Readme Web Terminal THE LINUX FOUNDATION

```
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
```



Question #:13

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

The following TLScertificates/key are suppliedfor connecting to the server with*etcdctl:*

- CA certificate:*/opt/KUCM00302/ca.crt*
- Client certificate:*/opt/KUCM00302/etcd-client.crt*
- Client key:*Topt/KUCM00302/etcd-client.key*

See the solution below.

Explanation

solution

Readme Web Terminal THE LINUX FOUNDATION

```
root@node-1:~# ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379  --cacert=/opt/KUCM0
0302/ca.crt  --cert=/opt/KUCM00302/etcd-client.crt  --key=/opt/KUCM00302/etcd-client.key  sn
apshot save /srv/data/etcd-snapshot.db
{"level":"info","ts":1598530470.8313155,"caller":"snapshot/v3_snapshot.go:110","msg":"create
d 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":"fetchi
ng snapshot","endpoint":"https://127.0.0.1:2379"}
{"level":"info","ts":1598530470.8570414,"caller":"snapshot/v3_snapshot.go:134","msg":"fetche
d 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:~#
```

Question #:14

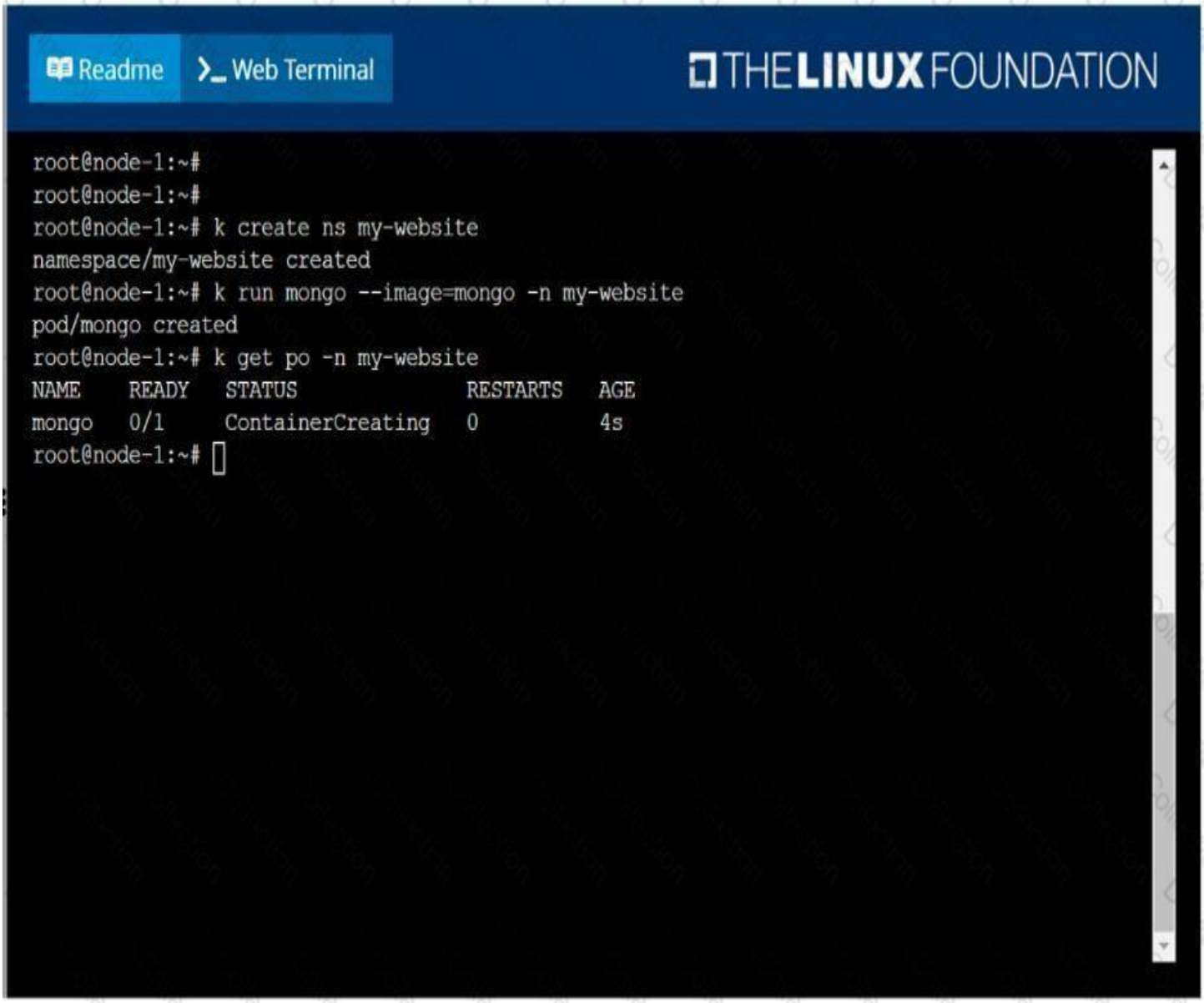
Create a pod as follows:

- Name:*mongo*
- Using Image:*mongo*
- In anew Kubernetes namespacenamed:*my-website*

See the solution below.

Explanation

solution



Question #:15

Create 2 nginx image pods in which one of them is labelled with env=prod and another one labelled with env=dev and verify the same.

See the solution below.

Explanation

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml >

nginx-prodpod.yaml Now, edit nginx-prod-pod.yaml file and remove entries like “creationTimestamp: null” “dnsPolicy: ClusterFirst”

vim nginx-prod-pod.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

env: prod

name: nginx-prod

spec:

containers:

- image: nginx

name: nginx-prod

restartPolicy: Always

kubectl create -f nginx-prod-pod.yaml

kubectl run --generator=run-pod/v1 --image=nginx --

labels=env=dev nginx-dev --dry-run -o yaml > nginx-dev-pod.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

env: dev

name: nginx-dev

spec:

containers:

- image: nginx

name: nginx-dev

restartPolicy: Always

kubectl create -f nginx-prod-dev.yaml

Verify :

kubectl get po --show-labels

kubectl get po -l env=prod

kubectl get po -l env=dev

Question #:16

Schedule a pod as follows:

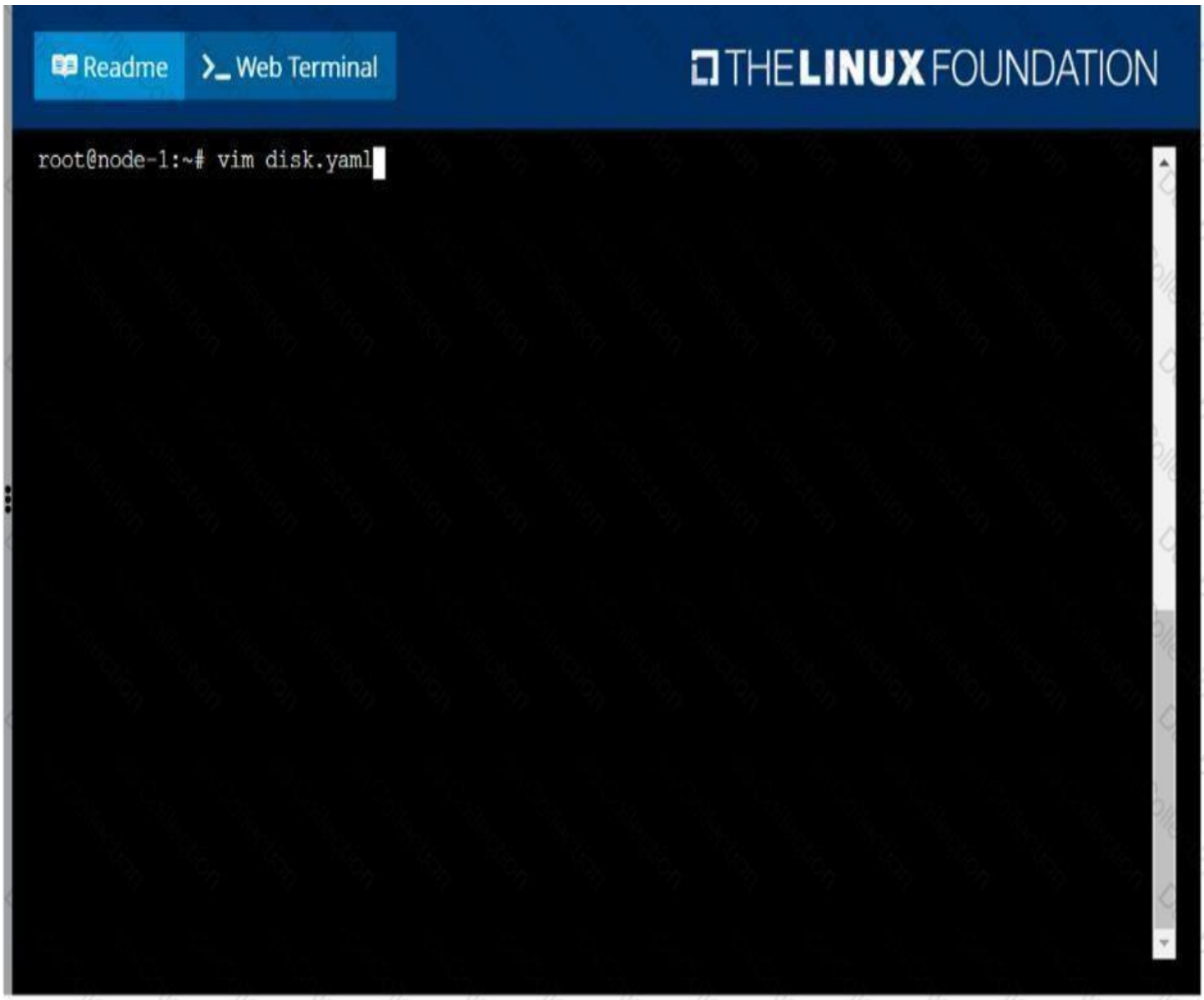
- **Name: nginx-kusc00101**
- **Image: nginx**
- **Node selector: disk=ssd**

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\6 B.JPG


Readme
Web Terminal
THE LINUX FOUNDATION
root@node-1:~# vim disk.yaml

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\6 D.JPG

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                         1/1     Running   0          6h1m
front-end                   1/1     Running   0          6h1m
hungry-bear                 1/1     Running   0          9m37s
kucc8                       3/3     Running   0          7m37s
nginx-kusc00101             1/1     Running   0          9s
webserver-84c55967f4-qzjcv  1/1     Running   0          6h16m
webserver-84c55967f4-t479l  1/1     Running   0          6h16m
root@node-1:~#
```

Question #:17

List all the pods showing name and namespace with a json path expression

See the solution below.

Explanation

kubectl get pods -o=jsonpath="{.items[*]['metadata.name',

'metadata.namespace']}"

Question #:18

A Kubernetes worker node, named*wk8s-node-0*is in state*NotReady*.Investigate why this is the case, andperform any appropriate steps tobring the node to a*Ready*state,ensuring that any changes are madepermanent.

You can*ssh*to the failednode using:

[student@node-1] $ | sshWk8s-node-0

You can assume elevatedprivileges on the node with thefollowing command:

[student@w8ks-node-0] $ |sudo –i

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\20 C.JPG

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\20 D.JPG

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

   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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\20 E.JPG

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 #:19

Create and configure the service*front-end-service*so it's accessiblethrough*NodePort*and routes to theexisting pod named*front-end.*

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\8 B.JPG

Readme Web Terminal THE LINUX FOUNDATION

```
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:~#
```

Question #:20

Create an nginx pod and list the pod with different levels of verbosity

See the solution below.

Explanation

// create a pod

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

// List the pod with different verbosity

kubectl get po nginx --v=7

kubectl get po nginx --v=8

kubectl get po nginx --v=9

Question #:21

List all the pods sorted by created timestamp

See the solution below.

Explanation

kubect1 get pods--sort-by=.metadata.creationTimestamp

Question #:22

Perform the following tasks:

- **Add an init container tohungry-bear(which has beendefined in spec file /opt/KUCC00108/pod-spec-KUCC00108.yaml)**
- **The init container should createan empty file named/workdir/calm.txt**
- **If/workdir/calm.txtis notdetected, the pod should exit**
- **Once the spec file has beenupdatedwith the init containerdefinition, the pod should becreated**

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\4 B.JPG

Readme Web Terminal THE LINUX FOUNDATION

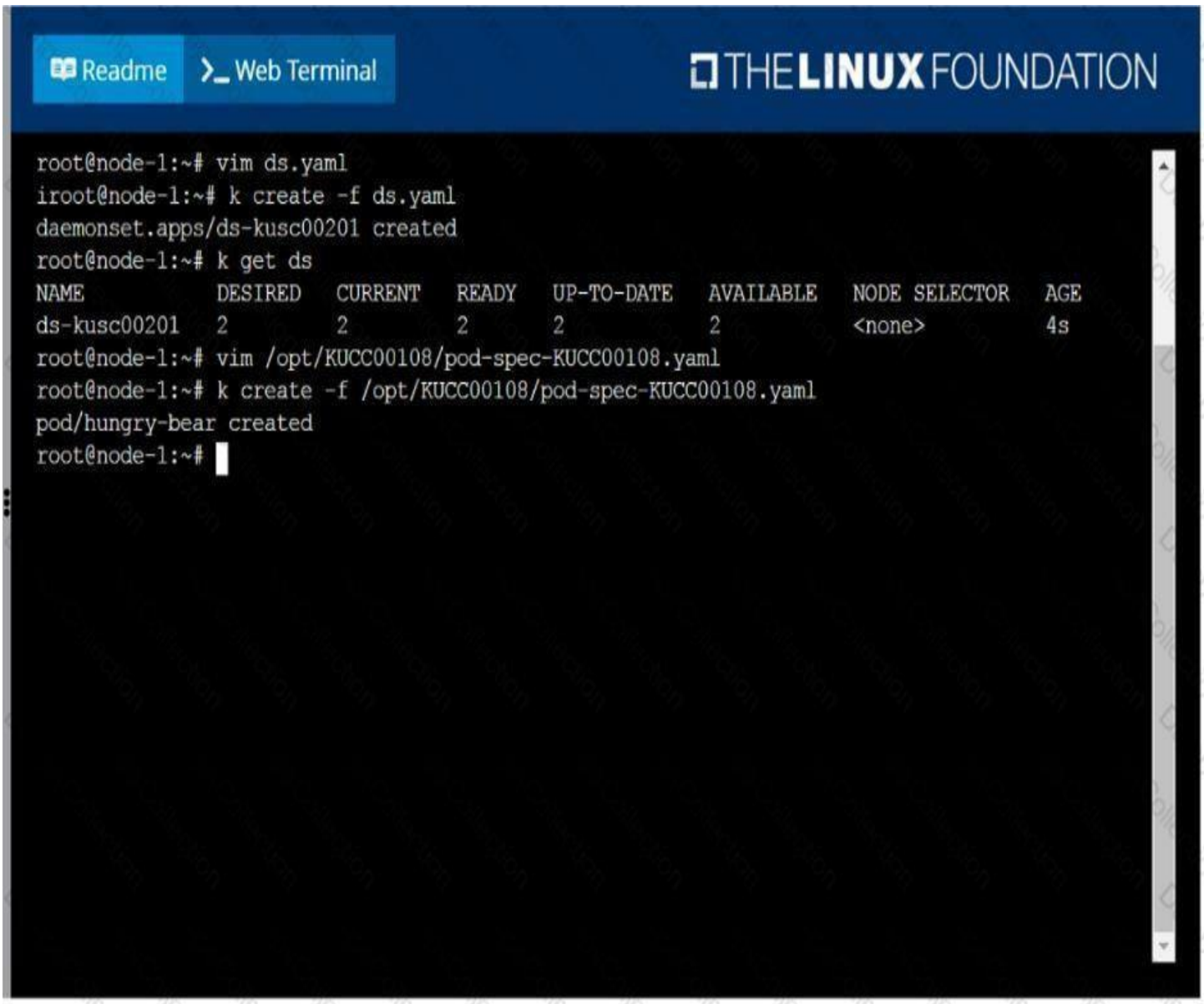
```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201  2         2         2       2            2           <none>          4s
root@node-1:~#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

F:\Work\Data Entry Work\Data Entry\abc\CKA\4 C.JPG

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
:wq
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\4 D.JPG



Question #:23

Check the image version in pod without the describe command

See the solution below.

Explanation

kubectl get po nginx -o

jsonpath='{.spec.containers[].image}{"\n"}'

Question #:24

Create a persistent volume with name*app-data*, of capacity*2Gi*andaccess mode*ReadWriteMany*. Thetype of volume is*hostPath*and itslocation is*/srv/app-data*.

See the solution below.

Explanation

solution

Persistent Volume

A persistent volume is a piece of storage in aKubernetes cluster. PersistentVolumes are a cluster-level 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 knowthe 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 PersistentVolume

kind: PersistentVolumeapiVersion: v1metadata:name:**app-data**spec:capacity: # defines the capacity of PV we are creatingstorage:**2Gi**#the amount of storage we are tying to claimaccessModes: # defines the rights of the volumewe 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**ReadWriteMan**y, storage classname *shared*,**2Gi**of storage capacity and the host path**/srv/app-data**.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
      path: /srv/app-data
  storageClassName: shared
~
~
~
~
~
~
~
~
~
~
~
~
"app-data.yaml" 12L, 194C
```

2. Save the file and create the persistent volume.

Image for post

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

3. View the persistent volume.

```
njerry191@cloudshell:~ (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 willchange 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. Ensurethat the Persistent Volume Claim has the same storageClassName as the persistentVolume you had previously created.

kind: PersistentVolumeapiVersion: v1metadata:name:**app-data**

spec:

accessModes:-**ReadWriteMany**resources:

requests:storage:**2Gi**

storageClassName:**shared**

2. Save and create the pvc

njerry191@cloudshell:~(extreme-clone-2654111)$ kubect1 create -f app-data.yaml

persistentvolumeclaim/app-data created

3. View the pvc

Image for post

```
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.

Image for post

```
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: v1kind: Podmetadata:creationTimestamp: nullname: app-dataspec:volumes:- name:congigpvcpersistenVolumeClaim:claimName: app-datacontainers:- image: nginxname: appvolumeMounts:- mountPath: "**/srv/app-data**"name: configpvc

Question #:25

List the nginx pod with custom columns POD_NAME and POD_STATUS

See the solution below.

Explanation

kubectl get po -o=custom-columns="POD_NAME:.metadata.name,

POD_STATUS:.status.containerStatuses[].state"

Question #:26

Create a pod with environment variables as var1=value1.Check the environment variable in pod

See the solution below.

Explanation

kubectl run nginx --image=nginx --restart=Never --env=var1=value1

then

kubectl exec -it nginx -- env

or

kubectl exec -it nginx -- sh -c 'echo $var1'

or

kubectl describe po nginx | grep value1

Question #:27

For this item, you will havetos*sh*to the nodes*ik8s-master-0*and*ik8s-node-0*and complete all tasks on thesenodes. Ensure that you return tothe base node (hostname:*node-1*) when you havecompleted this item.

Context

As an administrator of a smalldevelopment team, you have beenasked to set up a Kubernetes clusterto test the viability of a newapplication.

Task

You must use*kubeadm*to performthis task. Any*kubeadm*invocationswill require the use of the *--ignore-preflight-errors=all*option.

- Configure the*nodeik8s-master-O*as a masternode. .
- Join the node*ik8s-node-o*tothe cluster.

See the solution below.

Explanation

solution

You must use the*kubeadm*configuration file located at*/etc/kubeadm.conf*when initializingyour cluster.

You may use any CNI pluginto complete this task, but ifyou don't have your favouriteCNI plugin's manifest URL athand, Calico is one popularoption:*https://docs.projectcalico.org/v3.14/manifests/calico.yaml*

Docker is already installedon both nodes and*apt*hasbeen configured so that you caninstall the required tools.

Question #:28

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

See the solution below.

Explanation

kubectl create namespace development

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

Question #:29

List all the pods sorted by name

See the solution below.

Explanation

kubect1 get pods --sort-by=.metadata.name

Question #:30

List “nginx-dev” and “nginx-prod” pod and delete those pods

See the solution below.

Explanation

kubect1 get pods -o wide

kubectl delete po “nginx-dev”kubectl delete po “nginx-prod”

Question #:31

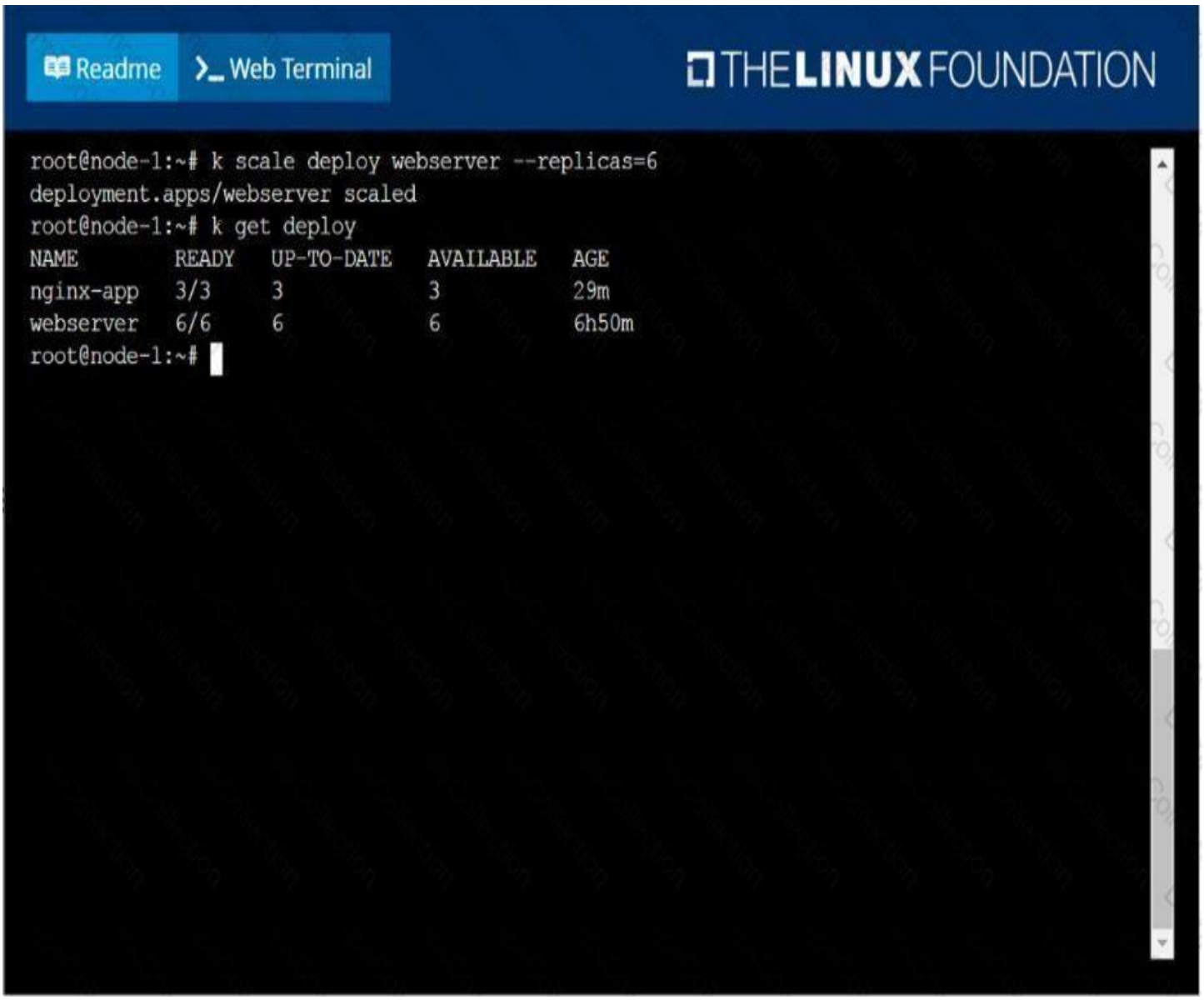
Scale the deployment*webserver*to*6*pods.

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\14 B.JPG



Question #:32

List all persistent volumes sorted bycapacity, saving the fullkubectloutput to /opt/KUCC00102/volume_list. Usekubectl 's own functionality forsorting the output, and do not manipulate it any further.

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\2 C.JPG

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 #:33

Create a pod namedkucc8with asingle app container for each of the

following images running inside(there may be between 1 and 4images specified):

nginx + redis + memcached.

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\5 B.JPG

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:~# 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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\5 C.JPG

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\5 D.JPG

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:~#
```

Question #:34

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

See the solution below.

Explanation

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

Question #:35

Ensure a single instance of podnginxis running on each node of theKubernetes cluster wherenginxalso represents the Image name whichhas to be used. Do not override anytaints currently in place.

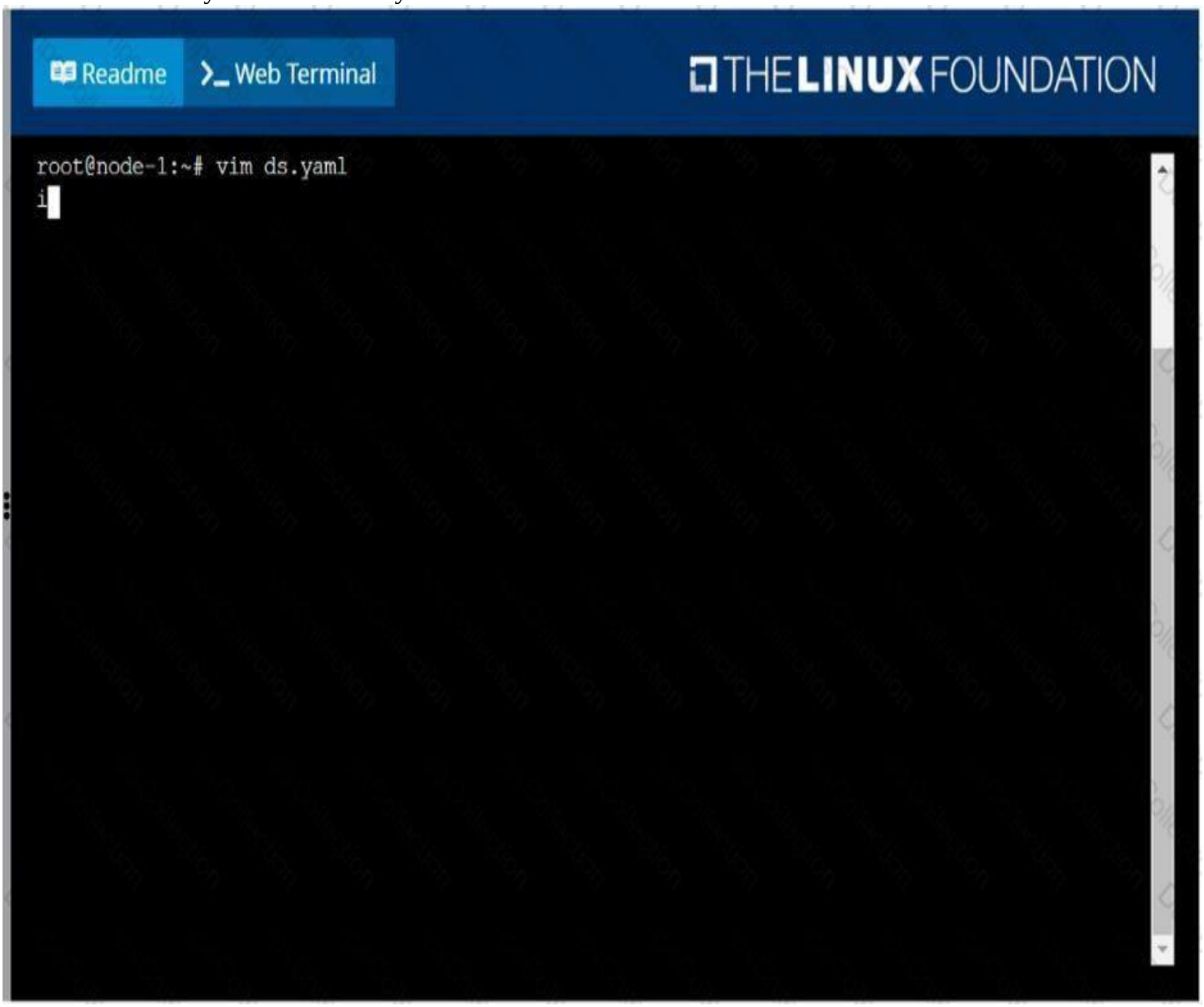
Use**DaemonSet**to complete thistask and use*ds-kusc00201*asDaemonSet name.

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\3 B.JPG



F:\Work\Data Entry Work\Data Entry\abc\CKA\3 C.JPG

Readme Web Terminal THE LINUX FOUNDATION

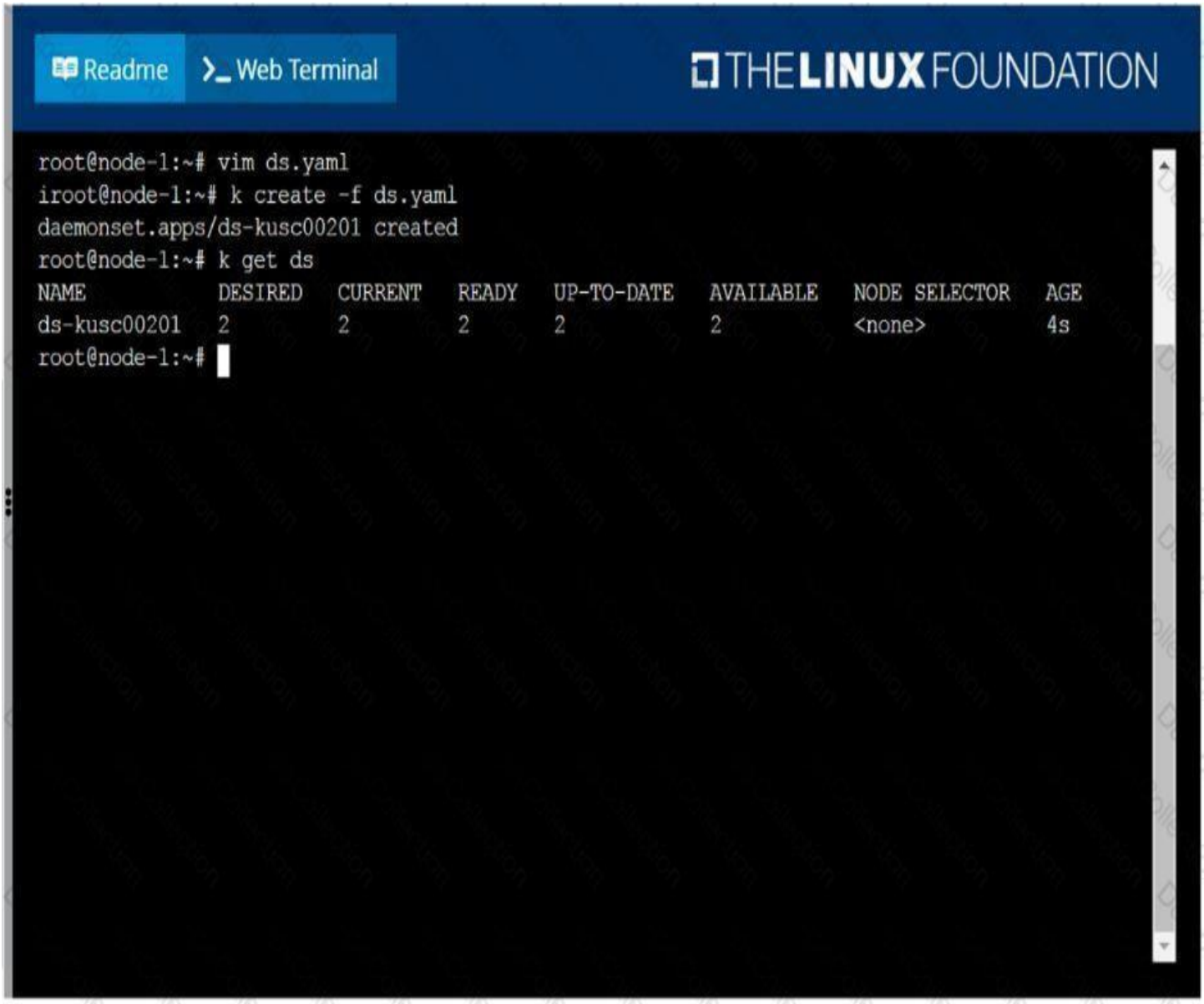
```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\3 D.JPG

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

F:\Work\Data Entry Work\Data Entry\abc\CKA\3 E.JPG



Question #:36

Print pod name and start time to "/opt/pod-status" file

See the solution below.

Explanation

kubect1 get pods -o=jsonpath='{range

items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'

Question #:37

Create a pod that having 3 containers in it? (Multi-Container)

See the solution below.

Explanation

image=nginx, image=redis, image=consul

Name nginx container as “nginx-container”

Name redis container as “redis-container”

Name consul container as “consul-container”

Create a pod manifest file for a container and append container

section for rest of the images

kubectl run multi-container --generator=run-pod/v1 --image=nginx --

dry-run -o yaml > multi-container.yaml

then

vim multi-container.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

run: multi-container

name: multi-container

spec:

containers:

- image: nginx

name: nginx-container

- image: redis

name: redis-container

- image: consul

name: consul-container

restartPolicy: Always

Question #:38

Get list of all the pods showing name and namespace with a jsonpath expression.

See the solution below.

Explanation

kubectl get pods -o=jsonpath="{.items[*]['metadata.name'

, 'metadata.namespace']}"

Question #:39

Create a pod that echo “hello world” and then exists. Have the pod deleted automatically when it’s completed

See the solution below.

Explanation

kubectl run busybox --image=busybox -it --rm --restart=Never --

/bin/sh -c 'echo hello world'

kubectl get po # You shouldn't see pod with the name "busybox"

Question #:40

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

See the solution below.

Explanation

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

Question #:41

Create a file:

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

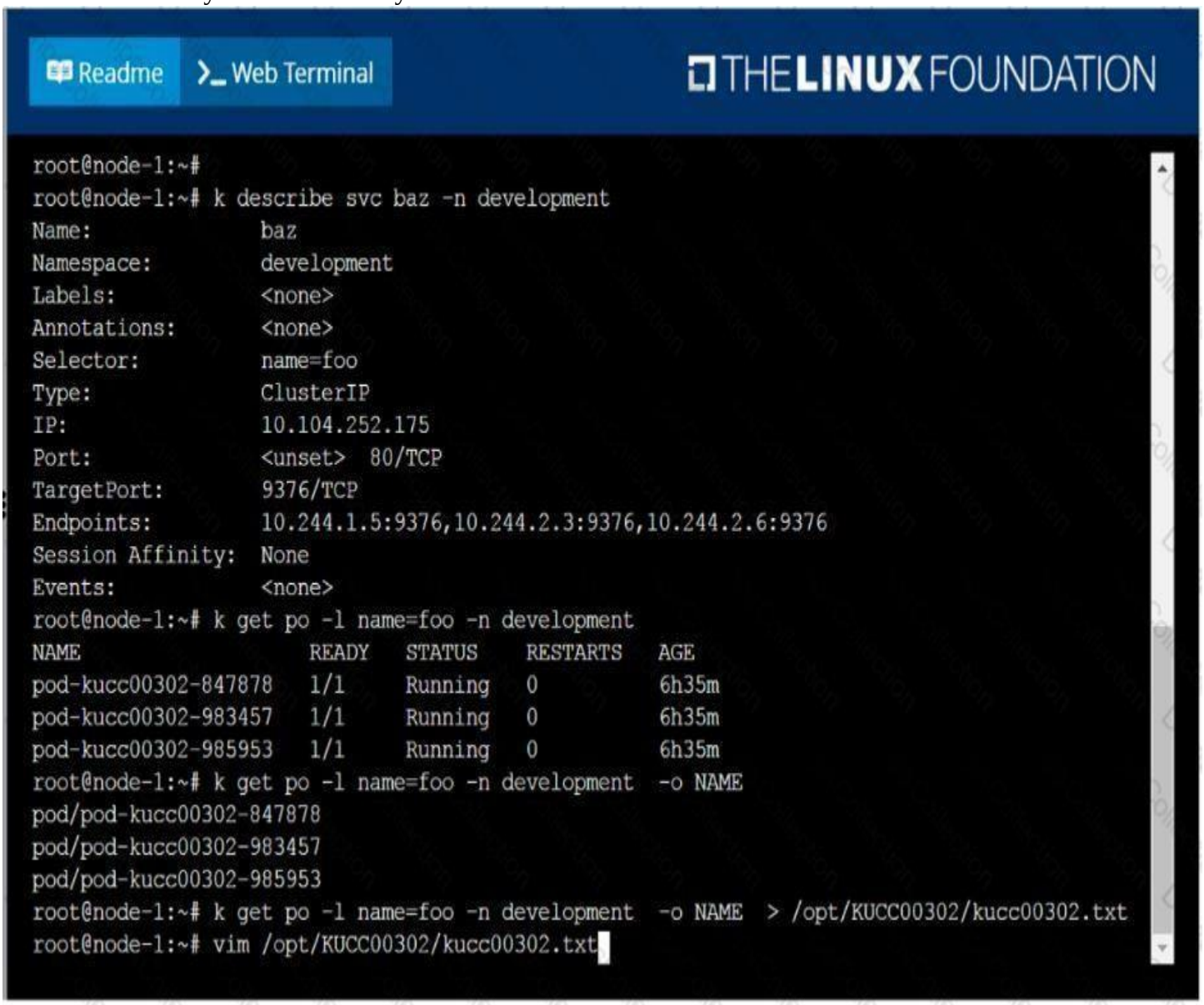
The format of the file should be onepod name per line.

See the solution below.

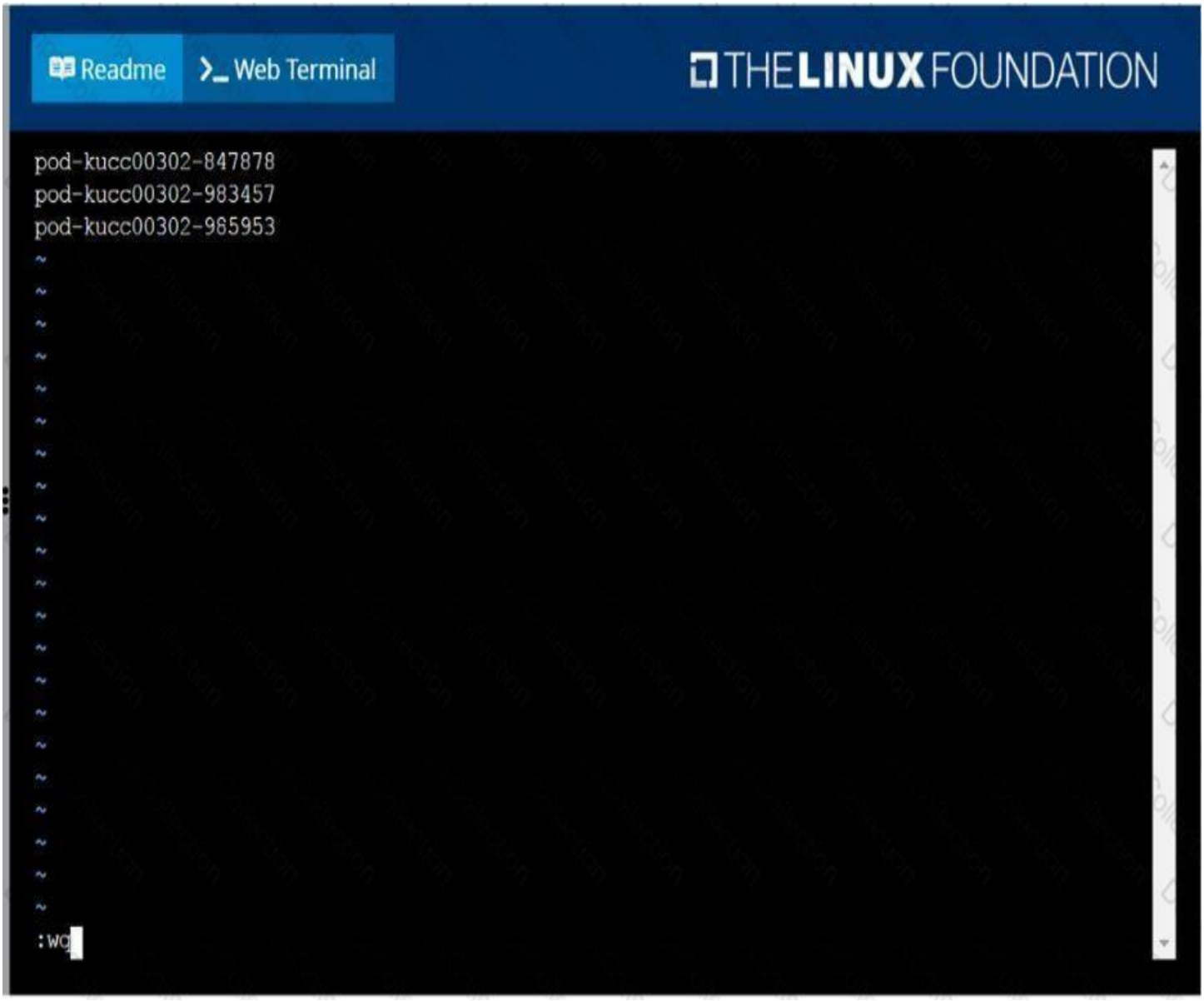
Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\11 B.JPG



F:\Work\Data Entry Work\Data Entry\abc\CKA\11 C.JPG



F:\Work\Data Entry Work\Data Entry\abc\CKA\11 D.JPG

Readme Web Terminal THE LINUX FOUNDATION

```
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
root@node-1:~# vim /opt/KUCC00302/kucc00302.txt
root@node-1:~#
```

Question #:42

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

See the solution below.

Explanation

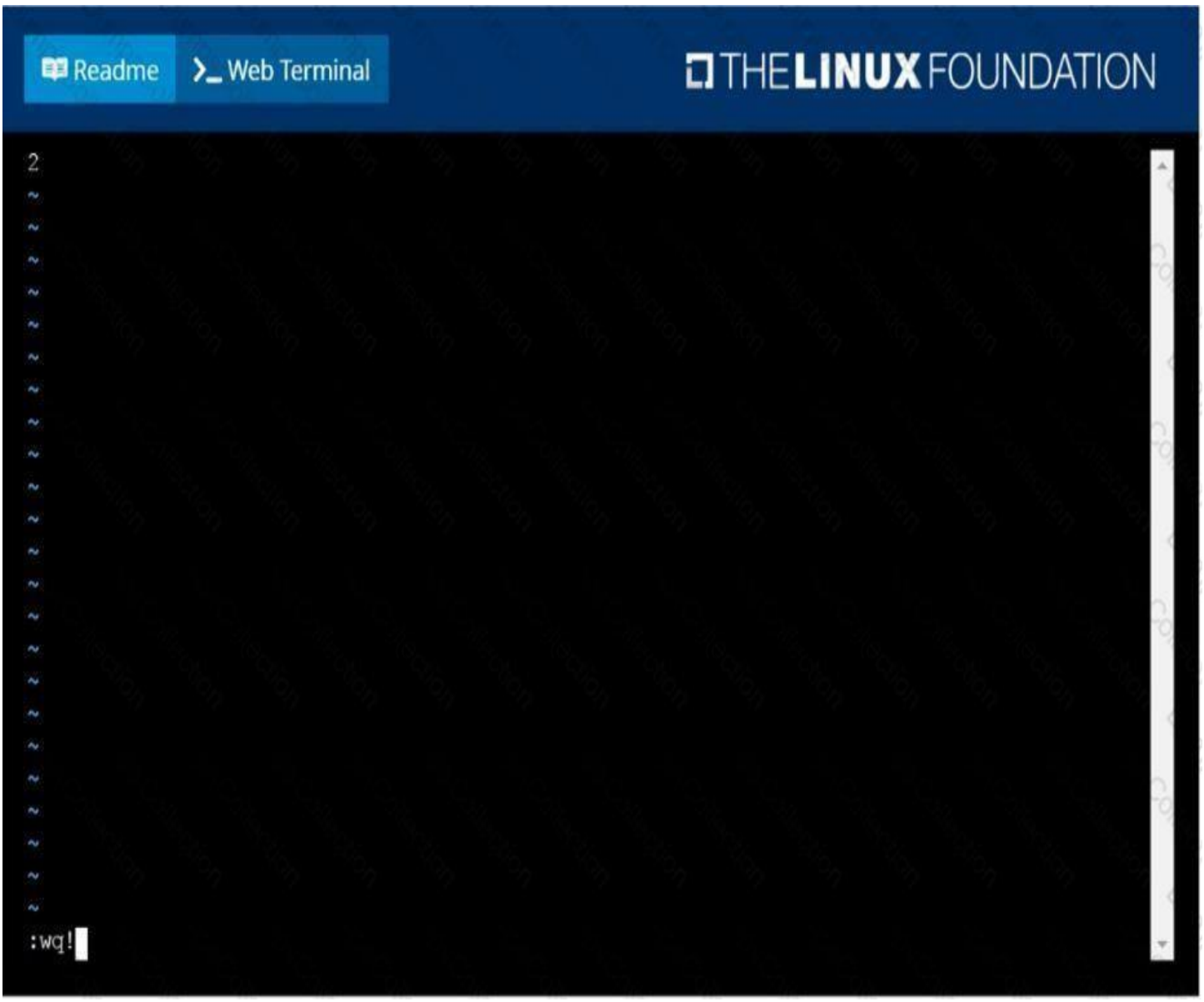
solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\15 B.JPG

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\15 C.JPG



Question #:43

Create a busybox pod and add “sleep 3600” command

See the solution below.

Explanation

kubectl run busybox --image=busybox --restart=Never -- /bin/sh -c

"sleep 3600"

Question #:44

Configure the kubelet systemd-managed service, on the nodelabelled with*name=wk8s-node-1*, tolaunch a pod containing a singlecontainer of Image*httpd*named*webtool*automatically. Any spec filesrequired should be placed in the*/etc/kubernetes/manifests*directoryon the node.

You canssh to theappropriate node using:

[student@node-1] $ sshwk8s-node-1

You can assume elevatedprivileges on the node with thefollowing command:

[student@wk8s-node-1] $ |sudo –i

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\21 C.JPG

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\21 D.JPG

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
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
syncFrequency: 0s
:wq
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\21 E.JPG

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\21 F.JPG

Readme Web Terminal THE LINUX FOUNDATION

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

F:\Work\Data Entry Work\Data Entry\abc\CKA\21 G.JPG

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-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 #:45

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

See the solution below.

Explanation

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

kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml | kubectl create -nengineering-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 #:46

Given a partially-functioningKubernetes cluster, identifysymptoms of failure on the cluster.

Determine the node, the failingservice, and take actions to bring upthe failed service and restore thehealth of the cluster. Ensure that anychanges are made permanently.

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

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

You can assume elevatedprivileges on any node in thecluster with the followingcommand:

[student@nodename] $ | sudo –i

See the solutionbelow.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\23 C.JPG

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
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\23 D.JPG

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
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
syncFrequency: 0s
volumeStatsAggPeriod: 0s
:wq
```

F:\Work\Data Entry Work\Data Entry\abc\CKA\23 E.JPG

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 #:47

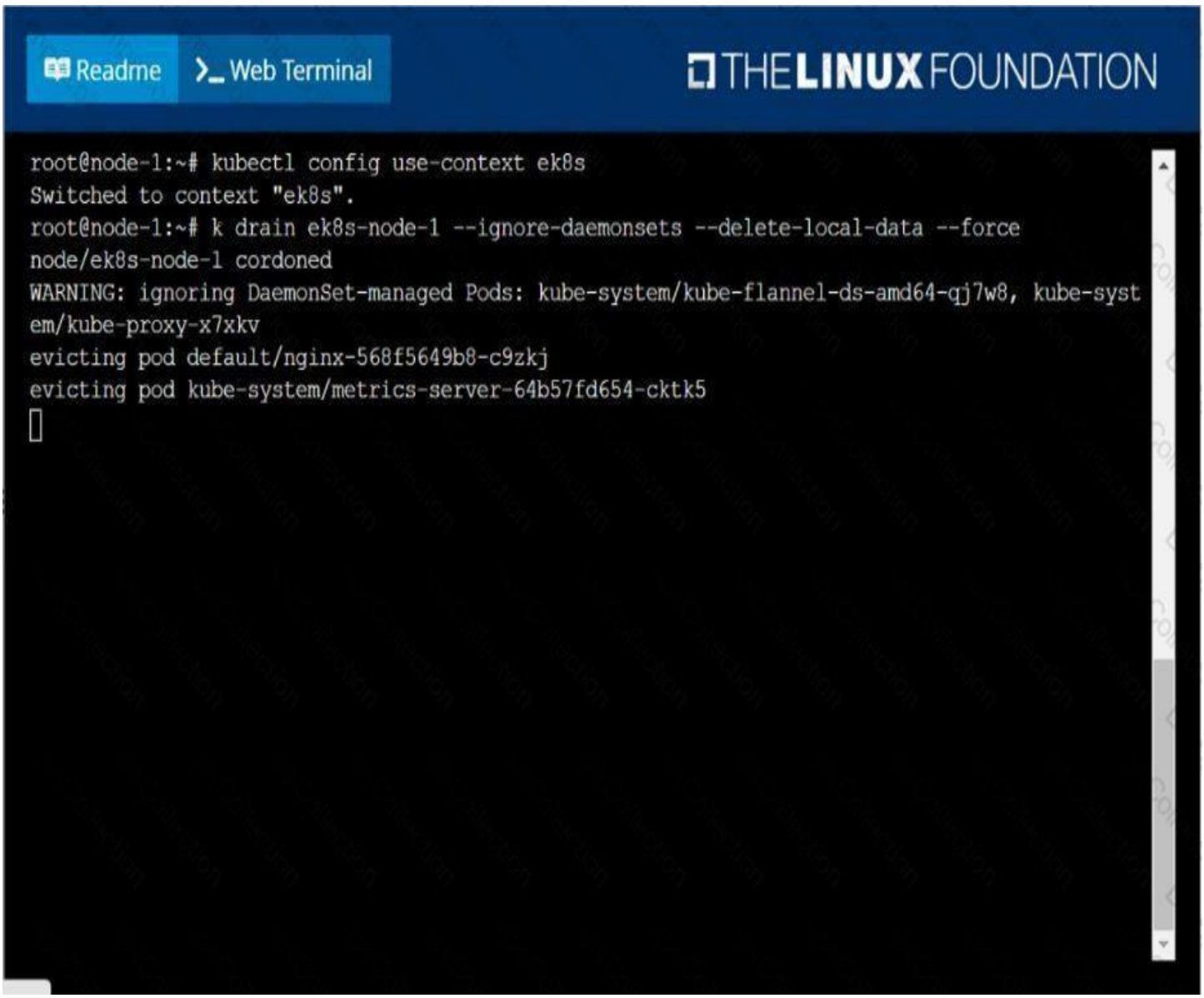
Set the node named*ek8s-node-1*asunavailable and reschedule all thepods running on it.

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\abc\CKA\19 B.JPG



Question #:48

Check the Image version of nginx-dev pod using jsonpath

See the solution below.

Explanation

kubect1 get po nginx-dev -o

jsonpath='{.spec.containers[].image}{"\n"}'