Linux Foundation CKA Practice Questions

Shared by Castro on 18-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

SIMULATION

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-O as a master node. .

Join the node ik8s-node-o to the cluster.

Options:

A- See the solution below

Answer:

A

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 2

Question Type: MultipleChoice

SIMULATION

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

Options:

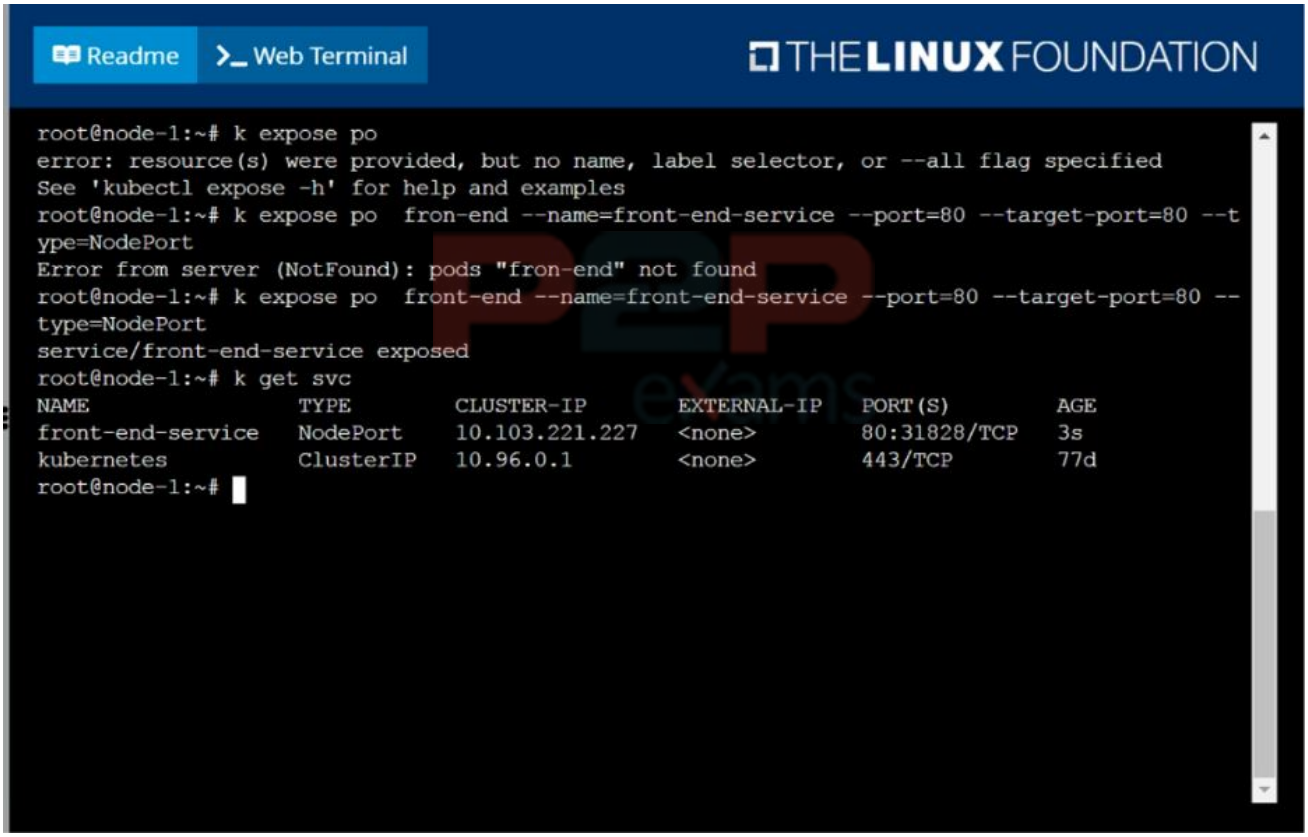
A- See the solution below

Answer:

A

Explanation:

solution



Question 3

Question Type: MultipleChoice

SIMULATION

You must connect to the correct host.

Failure to do so may result in a zero score.

[candidate@base] $ ssh Cka000046

Task

First, create a new StorageClass named local-path for an existing provisioner named rancher.io/local-path .

Set the volume binding mode to WaitForFirstConsumer .

Not setting the volume binding mode or setting it to anything other than WaitForFirstConsumer may result in a reduced score.

Next, configure the StorageClass local-path as the default StorageClass .

Options:

A- See the solution below

Answer:

A

Explanation:

Task Summary

You need to:

SSH into cka000046

Create a StorageClass named local-path using the provisioner rancher.io/local-path

Set the volume binding mode to WaitForFirstConsumer

Make this StorageClass the default

Step-by-Step Solution

1 SSH into the correct host

ssh cka000046

Required. Skipping this = zero score

2 Create a StorageClass YAML file

Create a file named local-path-sc.yaml:

cat <<EOF > local-path-sc.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: local-path

annotations:

storageclass.kubernetes.io/is-default-class: 'true'

provisioner: rancher.io/local-path

volumeBindingMode: WaitForFirstConsumer

EOF

This:

Sets WaitForFirstConsumer (as required)

Marks the class as default using the correct annotation

3 Apply the StorageClass

kubectl apply -f local-path-sc.yaml

4 Verify it's the default StorageClass

kubectl get storageclass

You should see local-path with a (default) marker:

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE

local-path rancher.io/local-path Delete WaitForFirstConsumer false 10s

Final Command Summary

ssh cka000046

cat <<EOF > local-path-sc.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: local-path

annotations:

storageclass.kubernetes.io/is-default-class: 'true'

provisioner: rancher.io/local-path

volumeBindingMode: WaitForFirstConsumer

EOF

kubectl apply -f local-path-sc.yaml

kubectl get storageclass

Question 4

Question Type: MultipleChoice

SIMULATION

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

Options:

A- See the solution below

Answer:

A

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 5

Question Type: MultipleChoice

SIMULATION

You must connect to the correct host.

Failure to do so may result in a zero score.

[candidate@base] $ ssh Cka000060

Task

Install Argo CD in the cluster by performing the following tasks:

Add the official Argo CD Helm repository with the name argo

The Argo CD CRDs have already been pre-installed in the cluster

Generate a template of the Argo CD Helm chart version 7.7.3 for the argocd namespace and save it to ~/argo-helm.yaml . Configure the chart to not install CRDs.

Options:

A- See the solution below

Answer:

A

Explanation:

Task Summary

SSH into cka000060

Add the Argo CD Helm repo named argo

Generate a manifest (~/argo-helm.yaml) for Argo CD version 7.7.3

Target namespace: argocd

Do not install CRDs

Just generate, don't install

Step-by-Step Solution

1 SSH into the correct host

ssh cka000060

Required --- skipping this = zero score

2 Add the Argo CD Helm repository

helm repo add argo https://argoproj.github.io/argo-helm

helm repo update

This adds the official Argo Helm chart source.

3 Generate Argo CD Helm chart template (version 7.7.3)

Use the helm template command to generate a manifest and write it to ~/argo-helm.yaml.

helm template argocd argo/argo-cd \

--version 7.7.3 \

--namespace argocd \

--set crds.install=false \

> ~/argo-helm.yaml

argocd Release name (can be anything; here it's same as the namespace)

--set crds.install=false Disables CRD installation

> ~/argo-helm.yaml Save to required file

4 Verify the generated file (optional but smart)

head ~/argo-helm.yaml

Check that it contains valid Kubernetes YAML and does not include CRDs.

Final Command Summary

ssh cka000060

helm repo add argo https://argoproj.github.io/argo-helm

helm repo update

helm template argocd argo/argo-cd \

--version 7.7.3 \

--namespace argocd \

--set crds.install=false \

> ~/argo-helm.yaml

head ~/argo-helm.yaml # Optional verification

Question 6

Question Type: MultipleChoice

SIMULATION

You must connect to the correct host.

Failure to do so may result in a zero score.

[candidate@base] $ ssh Cka000047

Task

A MariaDB Deployment in the mariadb namespace has been deleted by mistake. Your task is to restore the Deployment ensuring data persistence. Follow these steps:

Create a PersistentVolumeClaim (PVC) named mariadb in the mariadb namespace with the

following specifications:

Access mode ReadWriteOnce

Storage 250Mi

You must use the existing retained PersistentVolume (PV).

Failure to do so will result in a reduced score.

There is only one existing PersistentVolume .

Edit the MariaDB Deployment file located at ~/mariadb-deployment.yaml to use PVC you

created in the previous step.

Apply the updated Deployment file to the cluster.

Ensure the MariaDB Deployment is running and stable.

Options:

A- See the solution below

Answer:

A

Explanation:

Task Overview

You're restoring a MariaDB deployment in the mariadb namespace with persistent data.

Tasks:

SSH into cka000047

Create a PVC named mariadb:

Namespace: mariadb

Access mode: ReadWriteOnce

Storage: 250Mi

Use the existing retained PV (there's only one)

Edit ~/mariadb-deployment.yaml to use the PVC

Apply the deployment

Verify MariaDB is running and stable

Step-by-Step Solution

1 SSH into the correct host

ssh cka000047

Required --- skipping = zero score

2 Inspect the existing PersistentVolume

kubectl get pv

Identify the only existing PV, e.g.:

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS

mariadb-pv 250Mi RWO Retain Available <none> manual

Ensure the status is Available, and it is not already bound to a claim.

3 Create the PVC to bind the retained PV

Create a file mariadb-pvc.yaml:

cat <<EOF > mariadb-pvc.yaml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: mariadb

namespace: mariadb

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 250Mi

volumeName: mariadb-pv # Match the PV name exactly

EOF

Apply the PVC:

kubectl apply -f mariadb-pvc.yaml

This binds the PVC to the retained PV.

4 Edit the MariaDB Deployment YAML

Open the file:

nano ~/mariadb-deployment.yaml

Look under the spec.template.spec.containers.volumeMounts and spec.template.spec.volumes

sections and update them like so:

Add this under the container:

yaml

CopyEdit

volumeMounts:

- name: mariadb-storage

mountPath: /var/lib/mysql

And under the pod spec:

volumes:

- name: mariadb-storage

persistentVolumeClaim:

claimName: mariadb

These lines mount the PVC at the MariaDB data directory.

5 Apply the updated Deployment

kubectl apply -f ~/mariadb-deployment.yaml

6 Verify the Deployment is running and stable

kubectl get pods -n mariadb

kubectl describe pod -n mariadb <mariadb-pod-name>

Ensure the pod is in Running state and volume is mounted.

Final Command Summary

ssh cka000047

kubectl get pv # Find the retained PV

Create PVC

cat <<EOF > mariadb-pvc.yaml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: mariadb

namespace: mariadb

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 250Mi

volumeName: mariadb-pv

EOF

kubectl apply -f mariadb-pvc.yaml

Edit Deployment

nano ~/mariadb-deployment.yaml

Add volumeMount and volume to use the PVC as described

kubectl apply -f ~/mariadb-deployment.yaml

kubectl get pods -n mariadb

Question 7

Question Type: MultipleChoice

SIMULATION

You must connect to the correct host.

Failure to do so may result in a zero score.

[candidate@base] $ ssh Cka000037

Context

A legacy app needs to be integrated into the Kubernetes built-in logging architecture (i.e.

kubectl logs). Adding a streaming co-located container is a good and common way to

accomplish this requirement.

Task

Update the existing Deployment synergy-leverager, adding a co-located container named sidecar using the busybox:stable image to the existing Pod . The new co-located container has to run the following command:

/bin/sh -c "tail -n+1 -f /var/log/syne

rgy-leverager.log"

Use a Volume mounted at /var/log to make the log file synergy-leverager.log available to the co-

located container .

Do not modify the specification of the existing container other than adding the required volume mount .

Failure to do so may result in a reduced score.

Options:

A- Define a shared volume
Add under volumes: (at the same level as containers):
volumes:
- name: log-volume
emptyDir: {}
B. Add volume mount to the existing container
Locate the existing container under containers: and add this:
volumeMounts:
- name: log-volume
mountPath: /var/log
Do not change any other configuration for this container.
C. Add the sidecar container
Still inside containers:, add the new container definition after the first one:
- name: sidecar
image: busybox:stable
command:
- /bin/sh
- -c
- 'tail -n+1 -f /var/log/synergy-leverager.log'
volumeMounts:
- name: log-volume
mountPath: /var/log
spec:
containers:
- name: main-container
image: your-existing-image
volumeMounts:
- name: log-volume
mountPath: /var/log

- name: sidecar
image: busybox:stable
command:
- /bin/sh
- -c
- 'tail -n+1 -f /var/log/synergy-leverager.log'
volumeMounts:
- name: log-volume
mountPath: /var/log
volumes:
- name: log-volume
emptyDir: {}
Save and exit
If using vi or vim, type:
bash
CopyEdit
wq
5 Verify
Check the updated pods:
kubectl get pods -l app=synergy-leverager
Pick a pod name and describe it:
kubectl describe pod
Confirm:
2 containers running (main-container + sidecar)
Volume mounted at /var/log
ssh cka000037
kubectl edit deployment synergy-leverager
Modify as explained above
kubectl get pods -l app=synergy-leverager
kubectl describe pod
A- See the solution below

Answer:

A, A

Explanation:

Task Summary

SSH into the correct node: cka000037

Modify existing deployment synergy-leverager

Add a sidecar container:

Name: sidecar

Image: busybox:stable

Command:

/bin/sh -c 'tail -n+1 -f /var/log/synergy-leverager.log'

Use a shared volume mounted at /var/log

Don't touch existing container config except adding volume mount

Step-by-Step Solution

1 SSH into the correct node

ssh cka000037

Skipping this will result in a zero score.

2 Edit the deployment

kubectl edit deployment synergy-leverager

This opens the deployment YAML in your default editor (vi or similar).

3 Modify the spec as follows

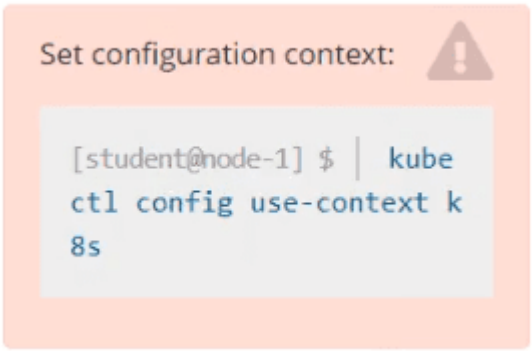
Inside the spec.template.spec, do these 3 things:

Question 8

Question Type: MultipleChoice

SIMULATION

Score: 5%



Task

Monitor the logs of pod bar and:

* Extract log lines corresponding to error file-not-found

* Write them to /opt/KUTR00101/bar

Options:

A- See the solution below

Answer:

A

Explanation:

Solution:

kubectl logs bar | grep 'unable-to-access-website' > /opt/KUTR00101/bar

cat /opt/KUTR00101/bar

Question 9

Question Type: MultipleChoice

SIMULATION

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

Options:

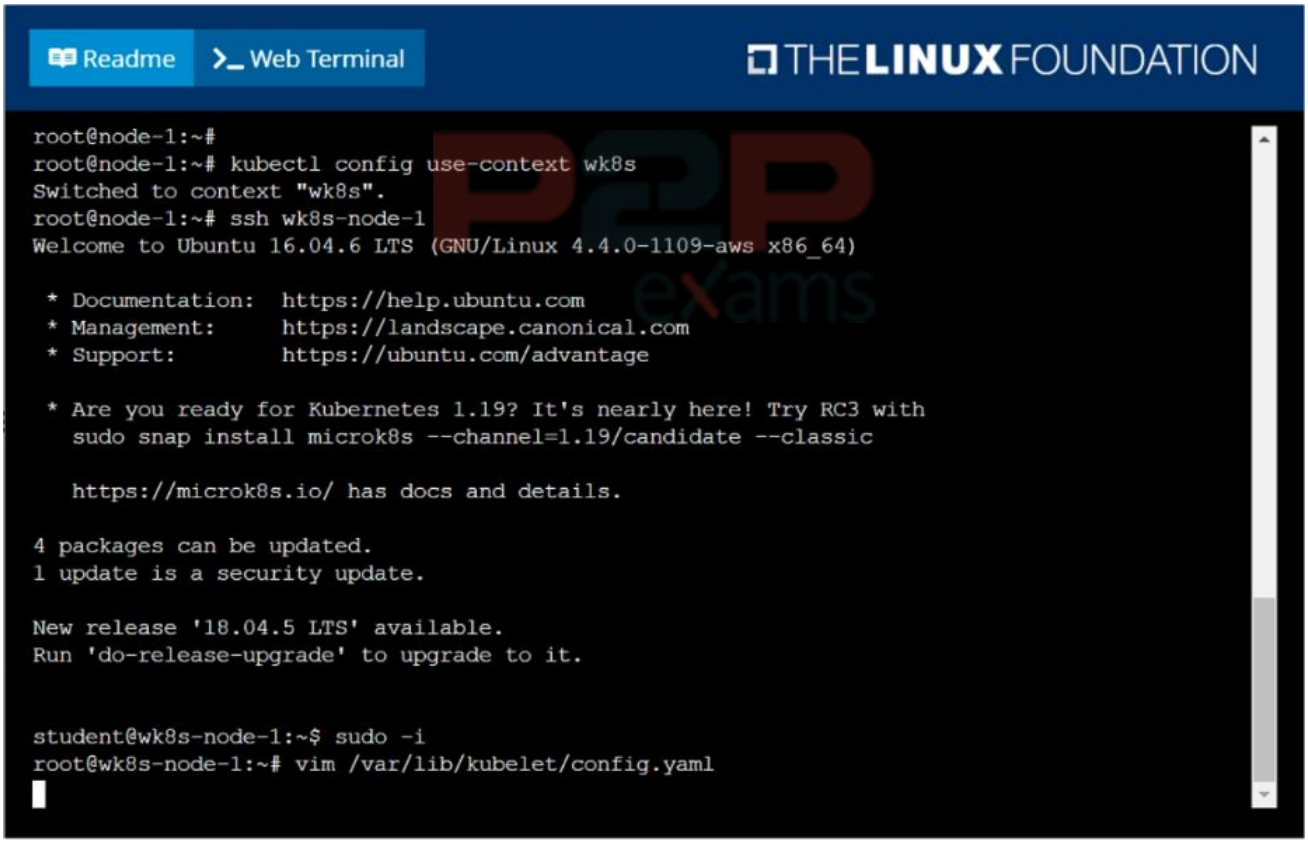
A- See the solution below

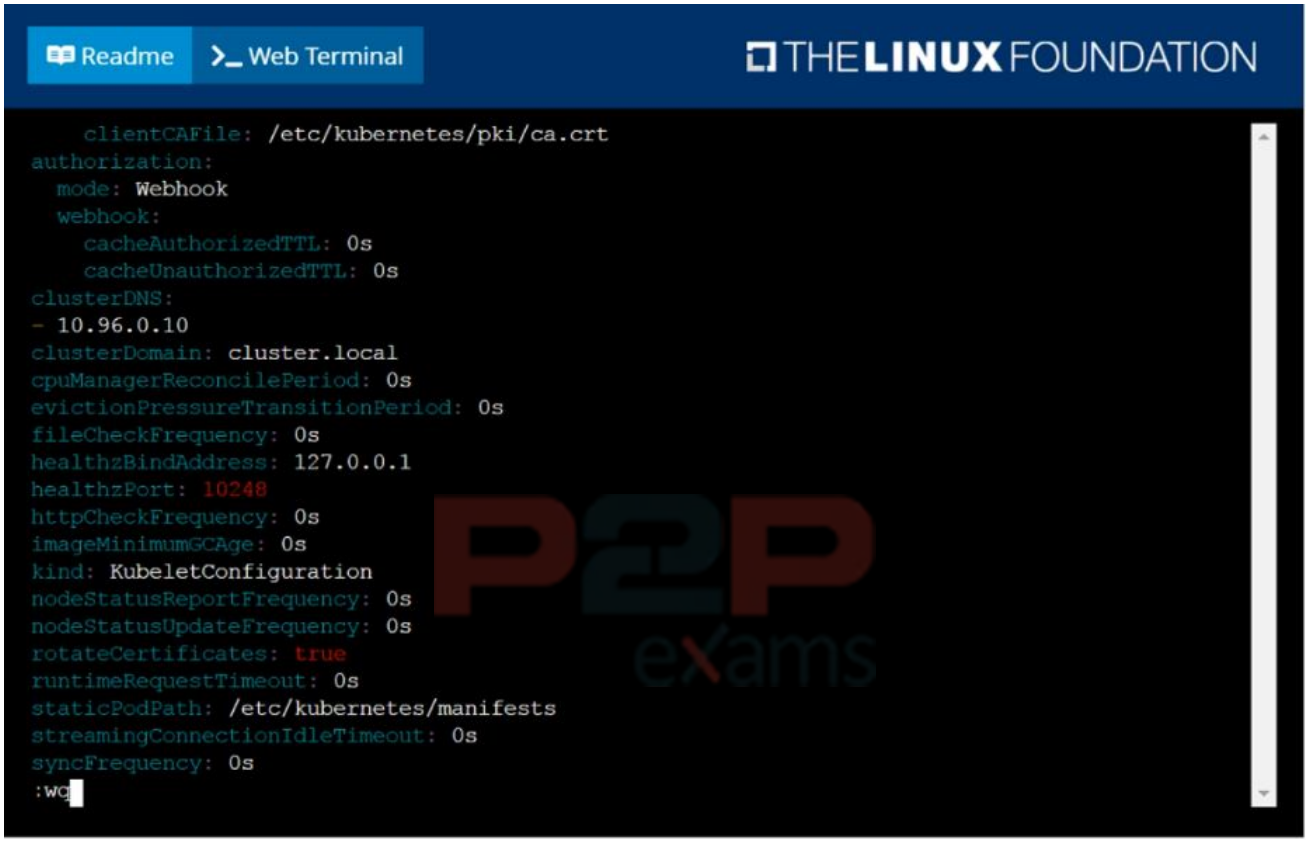
Answer:

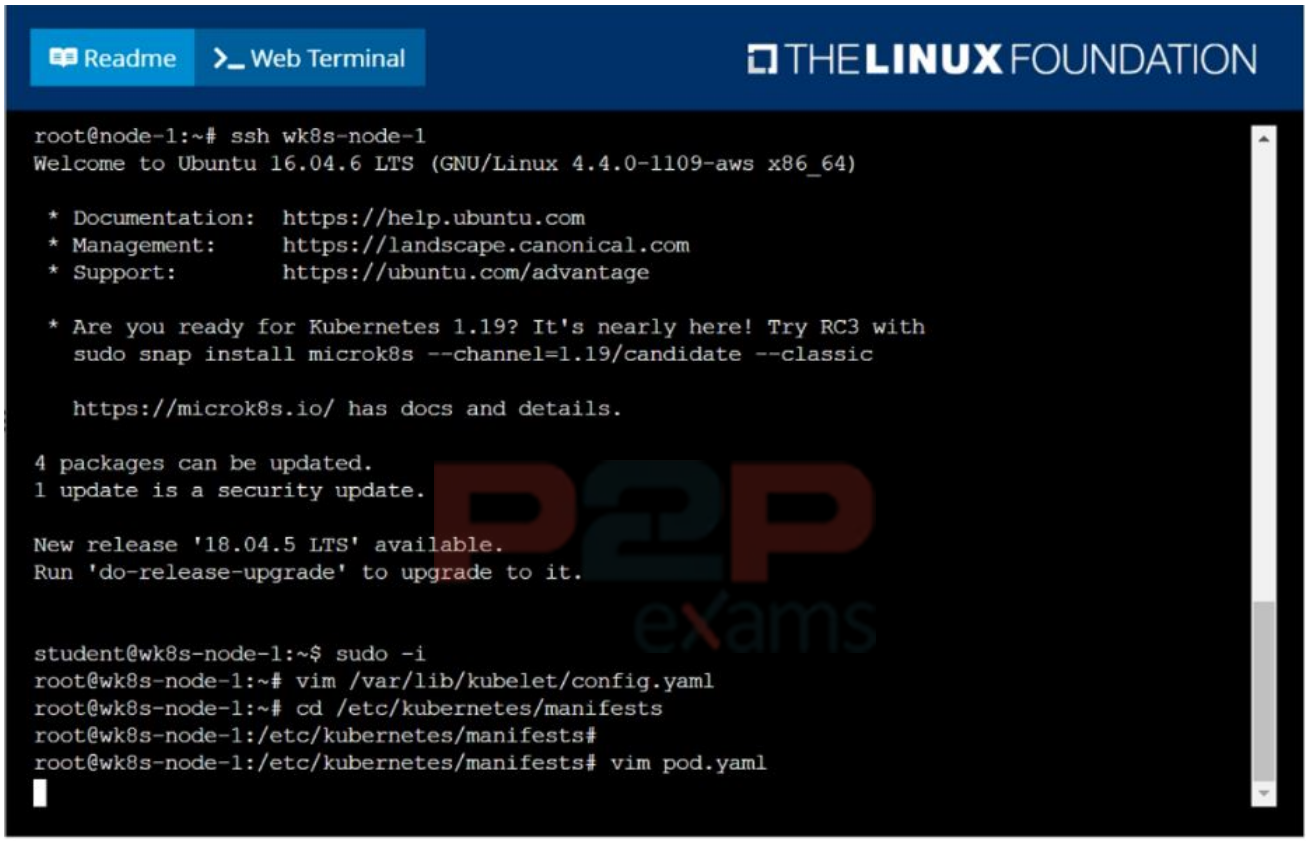
A

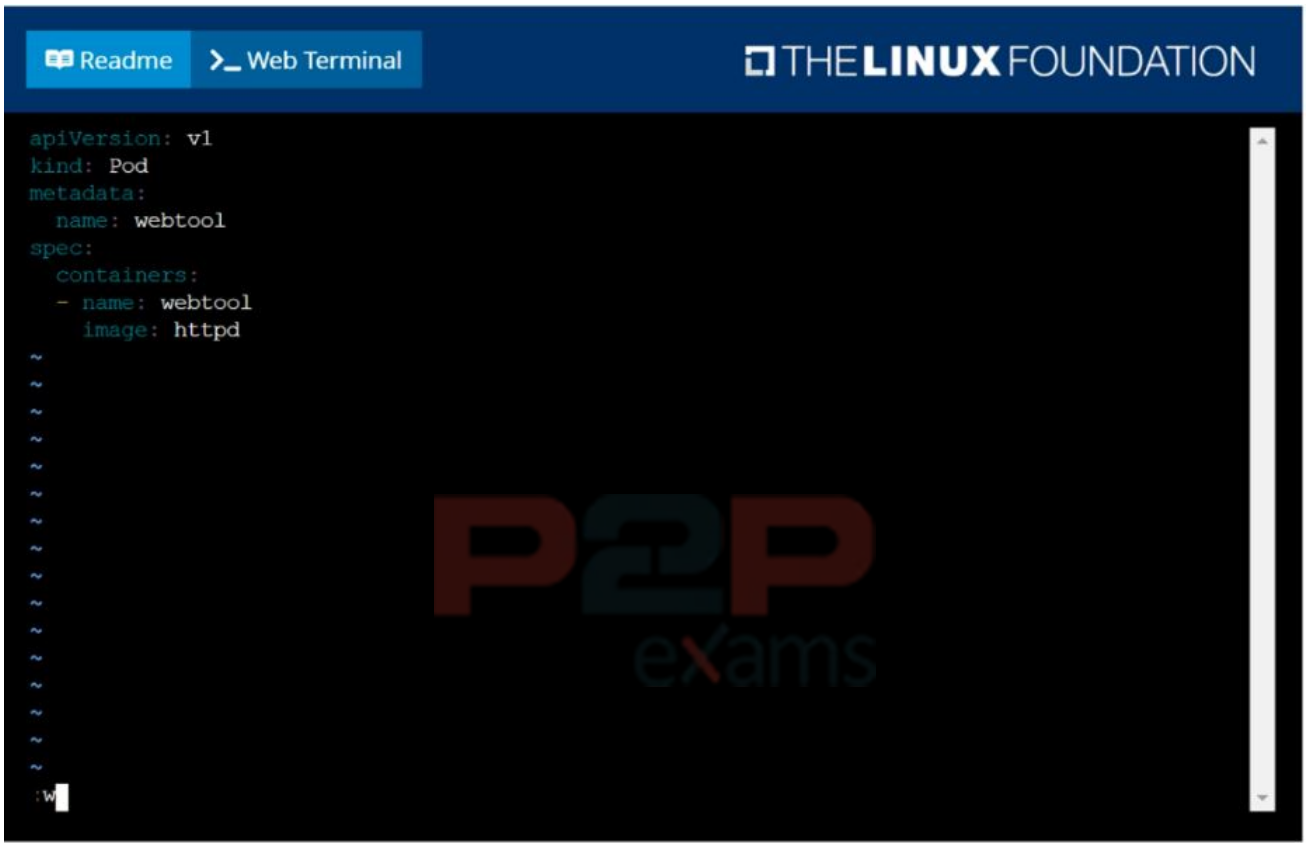
Explanation:

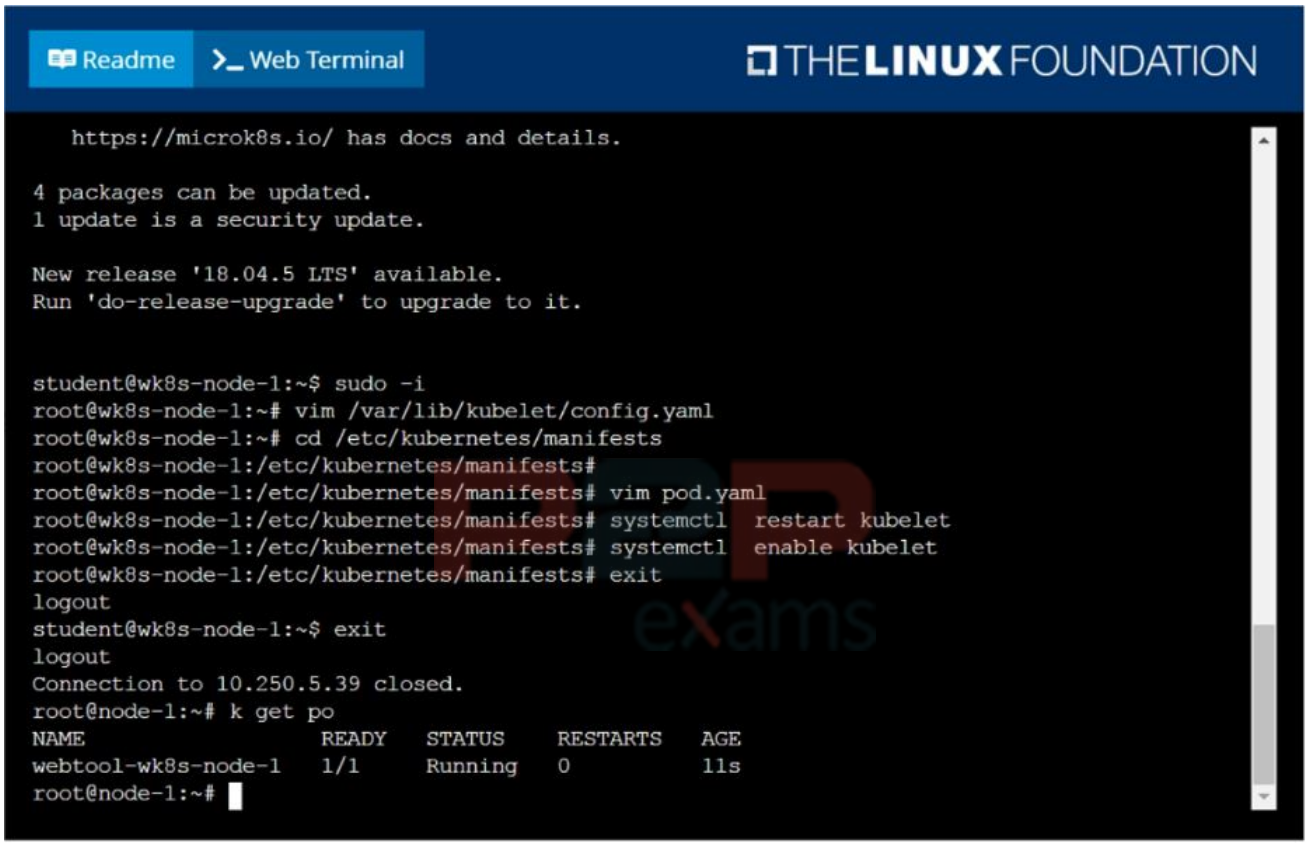
solution




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


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


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
name: webtool
spec:
containers:
- name: webtool
image: httpd
:w


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:~#