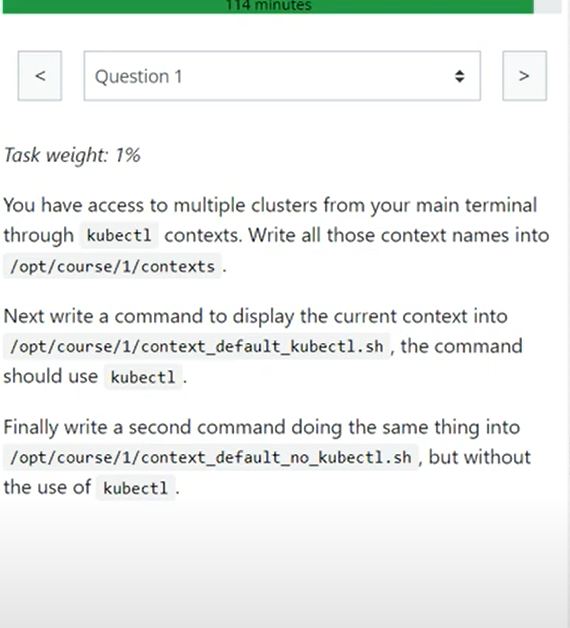
https://github.com/tariqsheikhsw/CKA-Simulator-Practice-Test-Killer-Sh



kubectl config get-contexts

kubectl config get-contexts --no-headers | awk {'print $2'} > /opt/course/1/contexts

OR

k config get-contexts -o name > /opt/course/1/contexts

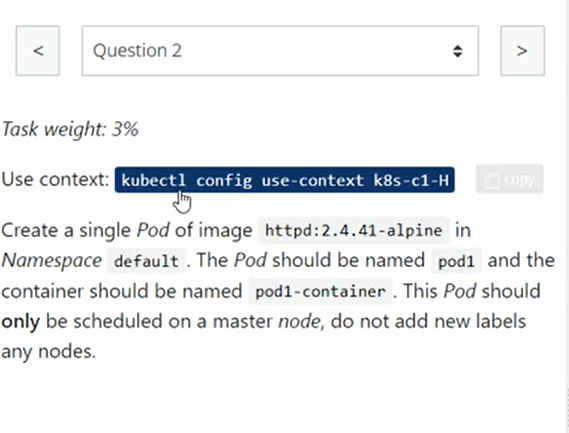
kubectl config get-contexts --output=name > /opt/course/1/contexts

echo "kubectl config current-context" > /opt/course/1/context\_default\_kubectl.sh

bash /opt/course/1/context\_default\_kubectl.sh

echo "cat .kube/config | grep -i current-context" > /opt/course/1/context\_default\_no\_kubectl.sh

bash /opt/course/1/context\_default\_no\_kubectl.sh



alias k=kubectl

kubectl config use-context k8s-c1-H

k get nodes --show-labels

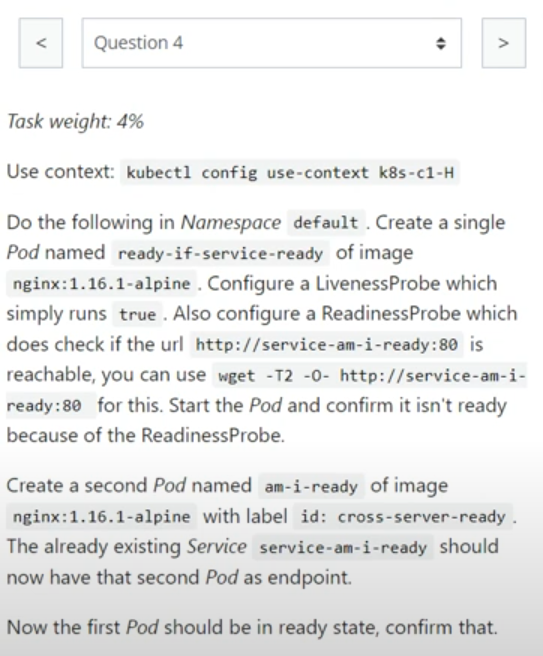
k run pod1 --image=httpd:2.4.41-alpine --dry-run=client -o yaml >> q2.yaml

vim q2.yaml

//add nodeName



k -n project-c13 scale statefulset 03db --replicas=1



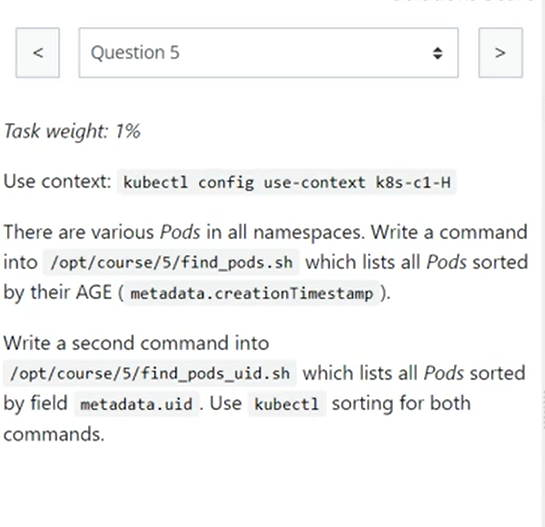
kubectl config use-context k8s-c1-H

k run ready-if-service-ready --image=nginx:1.16.1-alpine --dry-run=client -o yaml > q4pod1.yaml

k apply -f q4pod1.yaml

k run am-i-ready --image=nginx:1.16.1-alpine --labels=id=cross-server-ready

k describe svc service-am-i-ready

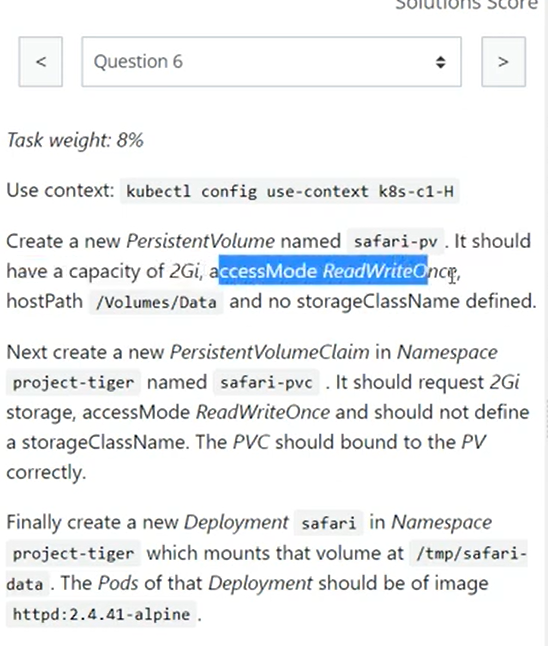


echo "kubectl get pod -A --sort-by=.metadata.creationTimestamp" > /opt/course/5/find\_pods.sh

bash find\_pods.sh

echo "kubectl get pod -A --sort-by=.metadata.uid" > find\_pods\_uid.sh

bash find\_pods\_uid.sh



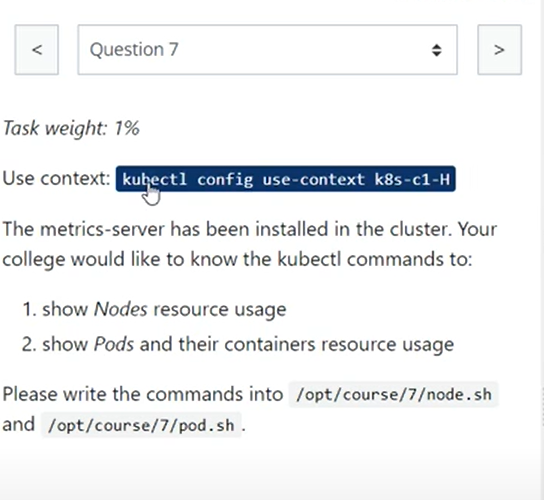
k create -f q6pv.yaml

k create -f q6pvc.yaml

k get pv,pvc -n project-tiger

k create deployment safari --image=httpd:2.4.41-alpine -n project-tiger --dry-run=client -o yaml > q6dep.yaml

k get pv,pvc,deployments.apps -n project-tiger



kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

k edit deployments.apps -n kube-system

add –kubelet-insecure-tls=true

k get deployments.apps -n kube-system

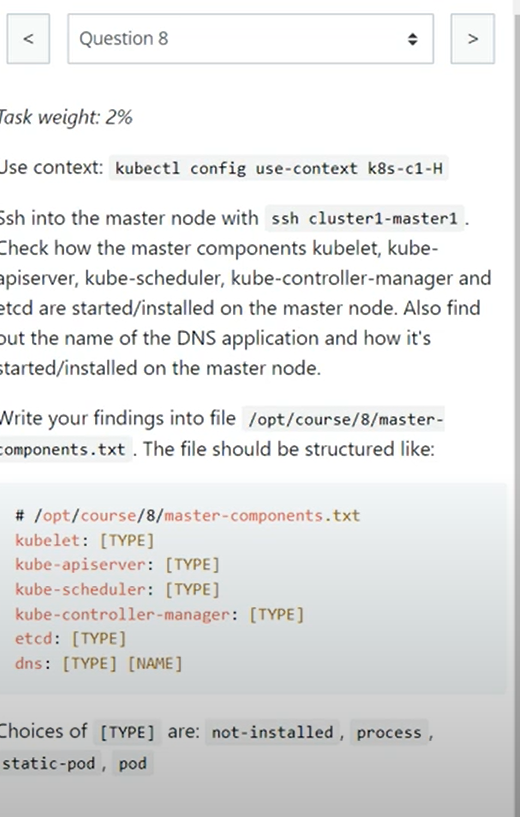
kubectl top nodes

echo "kubectl top nodes" > /opt/course/7/node.sh

bash /opt/course/7/node.sh

echo "kubectl top pods" > /opt/course/7/pod.sh

bash /opt/course/7/pod.sh



ssh cluster1-master1

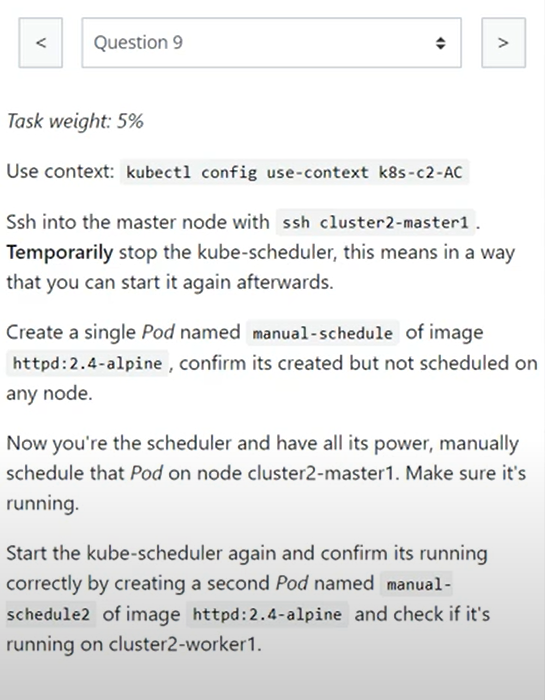
k get pod -n kube-system

k get deploy -n kube-system

k get all -n kube-system

ps -ef | grep -i kubelet

cd /etc/kubernetes/manifests/



kubectl config use-context k8s-c2-AC

ssh cluster2-master1

k get pod -n kube-system

cd /etc/kubernetes/manifests/

mv kube-scheduler.yaml /etc/kubernetes/

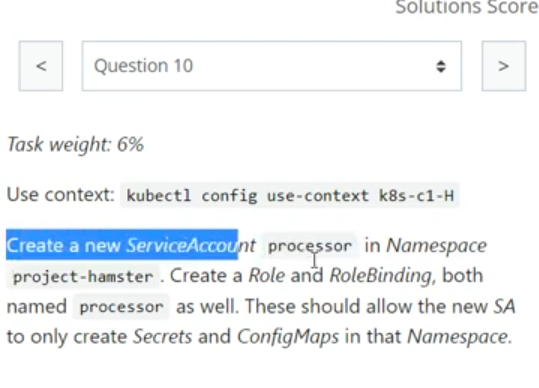
k run manual-schedule --image=httpd:2.4-alpine

k edit pod manual-schedule

k replace --force -f /tmp/kubectl-edit-2160969323.yaml

mv /etc/kubernetes/kube-scheduler.yaml /etc/kubernetes/manifests/

k run manual-schedule2 --image=httpd:2.4-alpine

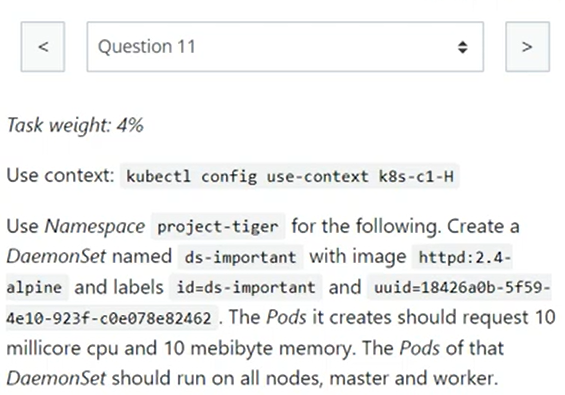


k create sa processor -n project-hamster

k create role processor --resource=secrets,configmaps --verb=create -n project-hamster

k create rolebinding processor --role processor --serviceaccount=project-hamster:process -n project-hamster

k get sa,role,rolebinding -n project-hamster

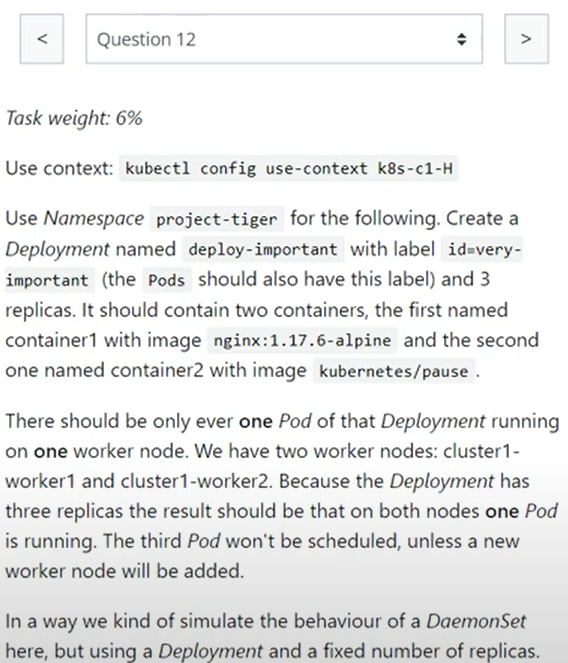


Create DaemonSet using template: <https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/>

kubectl config use-context k8s-c1-H

k create -f q11.yaml

k get ds -n project-tiger



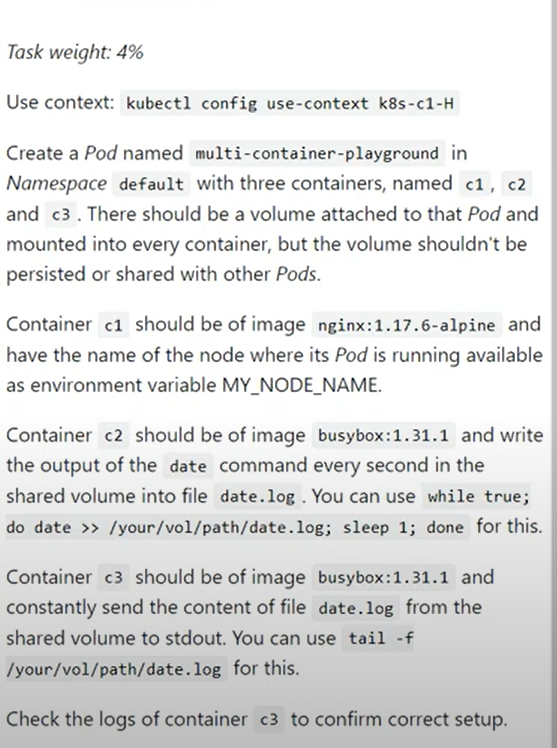
Deployment acting as DaemonSet (One Pod per Node)

k create deployment deploy-important --image=nginx:1.17.6-alpine -n project-tiger --replicas=3 --dry-run=client -o yaml > q12.yaml

k get pods -n project-tiger

k replace --force -f q12.yaml

Q: 13



k run multi-container-playground --image=nginx:1.17.6-alpine --dry-run=client -o yaml > q13.yaml

k replace --force -f q13.yaml

k get pod

k describe pod multi-container-playground

//path can be adjusted //can use args also in addition to command

- image: busybox

name: c2

command: ["bin/sh", "-c"]

args:

- while true; do

date >> /vol/date.log;

sleep 1;

done

volumeMounts:

- mountPath: /vol

name: volume

- image: busybox

name: c3

command: ["bin/sh","-c"]

args:

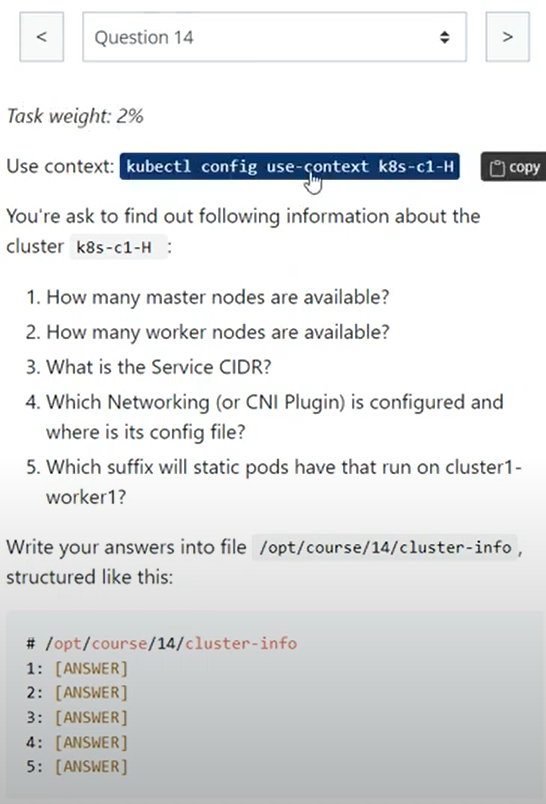
- tail -f /vol/date.log

Logs

k logs multi-container-playground c1

k logs multi-container-playground c2

k logs multi-container-playground c3



/opt/course/14/cluster-info

q1:How many master nodes are available?

1

[k get node]

q2:How many worker nodes are available?

2

[k get node]

q3:What is the Pod CIDR of cluster1-worker1?

10.244.1.0/24

[k describe node | less -p PodCIDR]

q4:What is the Service CIDR?

10.96.0.0/12

[cat /etc/kubernetes/manifests/kube-apiserver.yaml | grep range]

OR

[k get pod kube-apiserver-controlplane -n kube-system -o yaml | grep -i service]

q5:Which Networking (or CNI Plugin) is configured and where is its config file?

Weave, /etc/cni/net.d/10-weave.conflist

OR (in some cases)

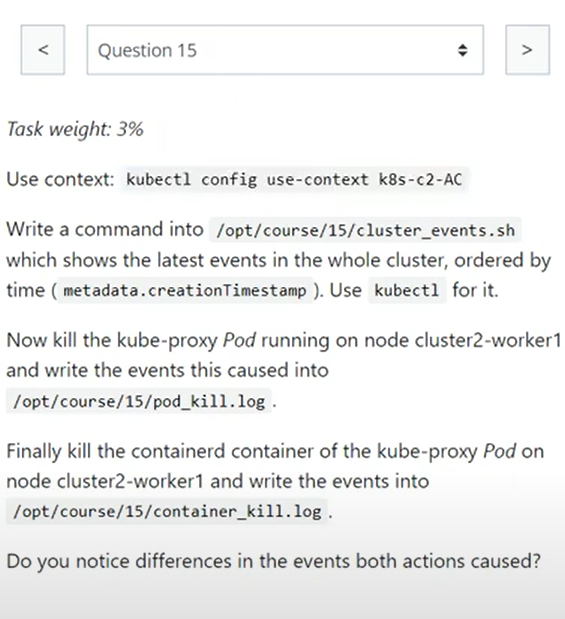
Calico, /etc/cni/net.d/calico-kubeconfig, /etc/cni/net.d/10-canal.conflist

[find /etc/cni/net.d/]

q6:Which suffix will static pods have that run on cluster1-worker1?

-cluster1-worker1

[k get pod]



kubectl get events -A --sort-by=.metadata.creationTimestamp

echo "kubectl get events -A --sort-by=.metadata.creationTimestamp" > /opt/course/15/cluster\_events.sh

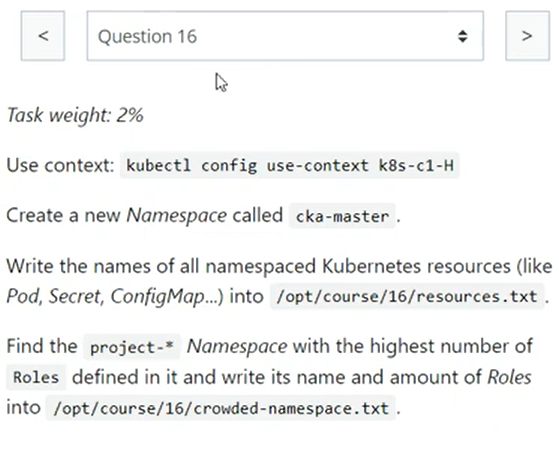
bash /opt/course/15/cluster\_events.sh

k get events -n kube-system > /opt/course/15/pod\_kill.log

crictl ps | grep kube-proxy

crictl stop ab4ae2d9784d7

k get events -n kube-system > /opt/course/15/container\_kill.log



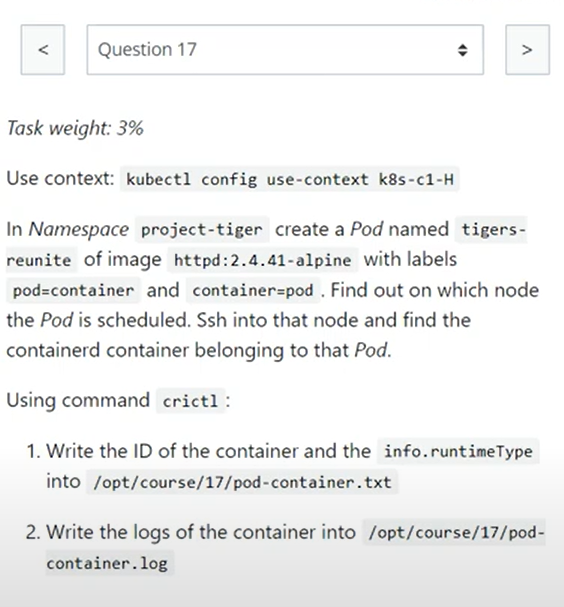
k create ns cka-master

k api-resources --namespaced -o name > /opt/course/16/resources.txt

OR

k api-resources --namespace=true | awk {'print $1'} > /opt/course/16/resources.txt

k get role -n <namespace> --no-headers | wc -l



k run tigers-reunite --image=httpd:2.4.41-alpine --labels "pod=container,container=pod" -n project-tiger

k get pod -n project-tiger -o wide

crictl ps | grep -i reunite

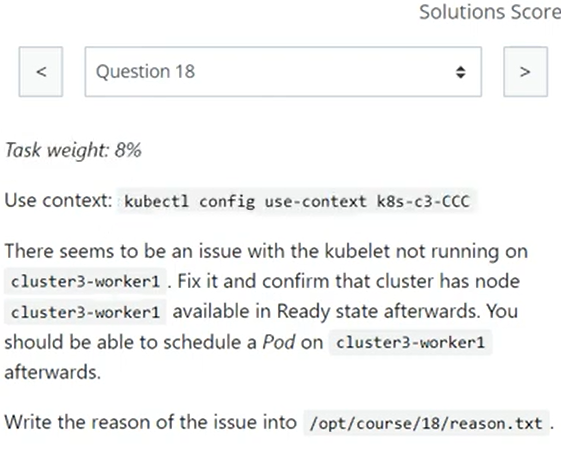
//output into given file path

crictl logs 70f8623c3ad4d

//output into given file path

k logs tigers-reunite -n project-tiger >> pod-container.log

//output into given file path



ps aux | grep kubelet

service kubelet status

service kubelet start

systemctl status kubelet

systemctl start kubelet

journalctl -u kubelet.service -f

ps -ef | grep -i kubelet

whereis kubelet

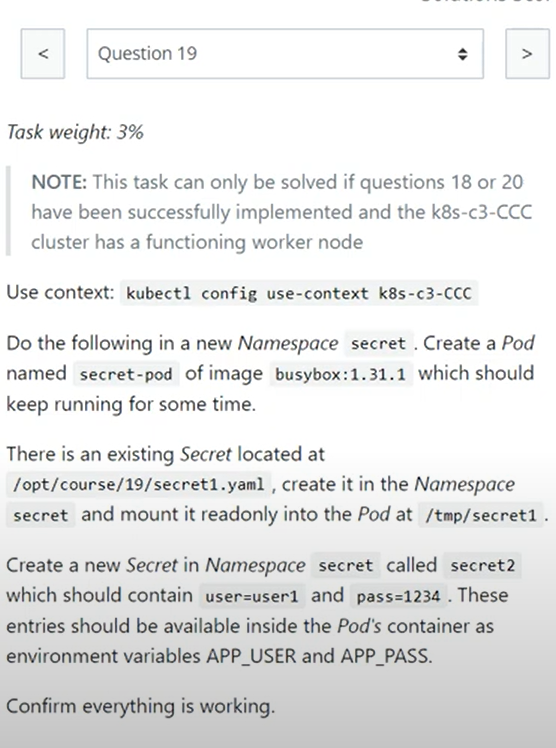
[/usr/bin/kubelet]

//correct path in config file

/etc/kubernetes/kubelet.conf

OR

/etc/systemd/system/kubelet.service.d/10-kubeadm.conf



k -n secret run secret-pod --image=busybox:1.31.1 --dry-run=client -o yaml -- sh -c "sleep 5d" > q19.yaml

OR

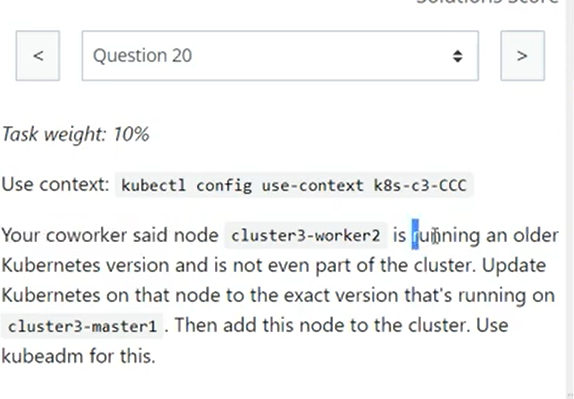
k -n secret run secret-pod --image=busybox:1.31.1 --dry-run=client -o yaml --command -- sleep 4800 > q19.yaml

k create -f /opt/course/19/secret1.yaml -n secret

k apply -f q19.yaml

k create secret generic secret2 --from-literal=user=user1 --from-literal=pass=1234 -n secret

k describe pod -n secret



kubeadm version

kubectl version

kubelet --version

kubeadm upgrade node

apt-get update

apt-cache madison kublet | grep -i 1.24.1

apt-cache show kubectl | grep 1.24.1

apt-mark unhold kublet kubectl

apt-get update && apt-get install kubectl=1.24.1-00 kubelet=1.24.1-00

apt-mark hold kublet kubectl

kubectl version --client

kubelet --version

sudo systemctl daemon-reload

sudo systemctl restart kubelet

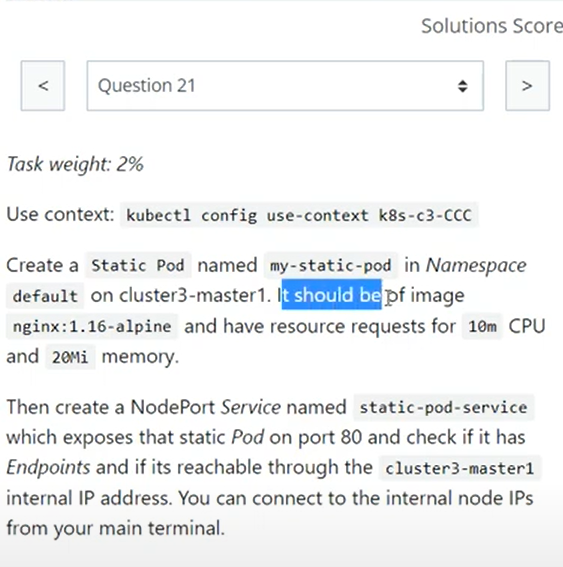
service kubelet status

kubeadm token create --print-join-command

//use resulting command 'kubadmin join xyz:6443 --token txy etc.]

kubeadm token list

service kubelet status



cd /etc/kubernetes/manifests/

kubectl run my-static-pod --image=nginx:1.16-alpine --dry-run=client -o yaml > my-static-pod.yaml

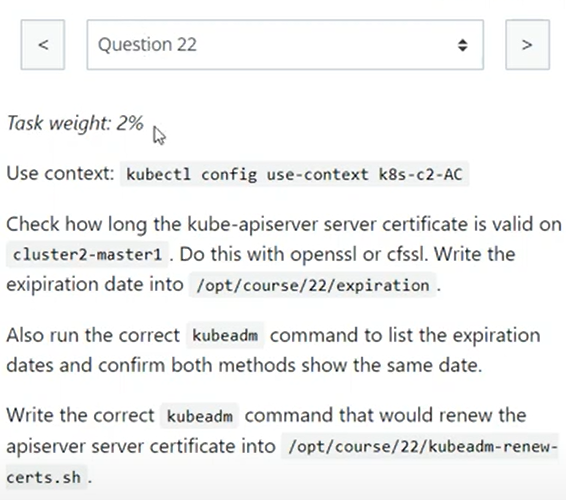
//--requests "cpu=10m,memory=20Mi"

k apply -f my-static-pod.yaml

k get pod -A | grep my-static-pod

kubectl expose pod my-static-pod-cluster3-master1 --name static-pod-service --type=NodePort --port 80

k get svc,ep -l run=my-static-pod



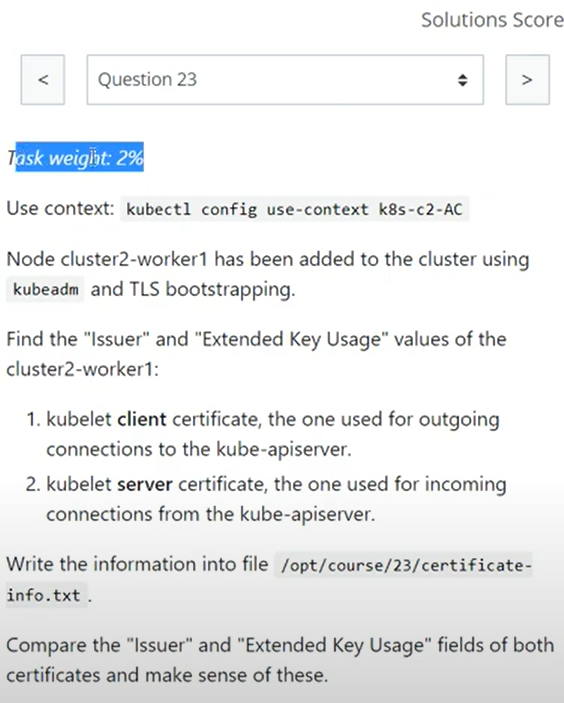
find /etc/kubernetes/pki | grep apiserver

openssl x509 -noout -text -in /etc/kubernetes/pki/apiserver.crt | grep Validity -A2

kubeadm certs check-expiration | grep apiserver

kubeadm certs renew apiserver

kubectl -n kube-system get cm kubeadm-config -o yaml

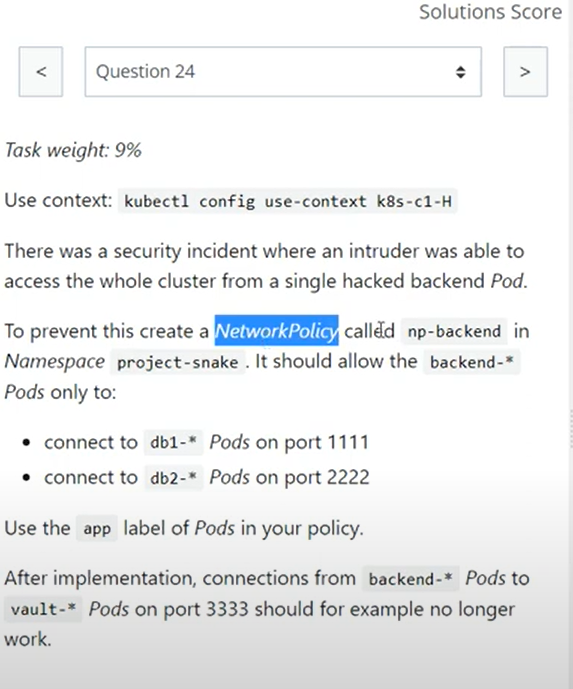


openssl x509 -noout -text -in /var/lib/kubelet/pki/kubelet-client-current.pem | grep Issuer

openssl x509 -noout -text -in /var/lib/kubelet/pki/kubelet-client-current.pem | grep "Extended Key Usage" -A1

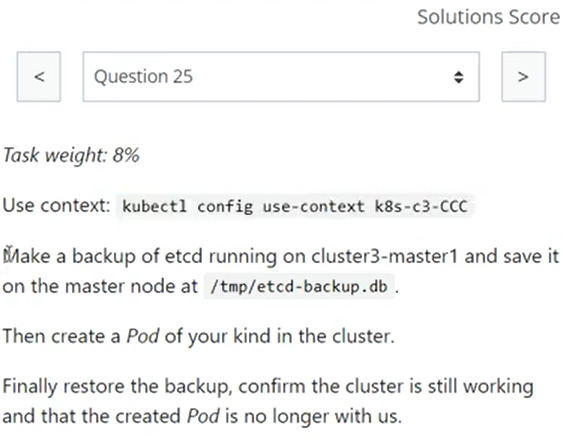
openssl x509 -noout -text -in /var/lib/kubelet/pki/kubelet.crt | grep Issuer

openssl x509 -noout -text -in /var/lib/kubelet/pki/kubelet.crt | grep "Extended Key Usage" -A1



k apply -f q24.yaml

k get networkpolicies -n project-snake



ETCD Snapshot SAVE and RESTORE Run 'k get pods -A' before and after RESTORE operation

ETCDCTL\_API=3 etcdctl snapshot save /tmp/etcd-backup.db

cat /etc/kubernetes/manifests/etcd.yaml

cat /etc/kubernetes/manifests/kube-apiserver.yaml | grep etcd

ETCDCTL\_API=3 etcdctl snapshot save /tmp/etcd-backup.db --cacert /etc/kubernetes/pki/etcd/ca.crt --cert /etc/kubernetes/pki/etcd/server.crt --key /etc/kubernetes/pki/etcd/server.key

OR

ETCDCTL\_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key snapshot save /tmp/etcd-backup.db

kubectl run test --image=nginx

ETCDCTL\_API=3 etcdctl snapshot restore /tmp/etcd-backup.db --data-dir /var/lib/etcd-backup

ETCDCTL\_API=3 etcdctl --data-dir /var/lib/etcd-backup snapshot restore /tmp/etcd-backup.db

//rm -r /var/lib/etcd-backup/\*

cd /etc/kubernetes/manifests/

cat etcd.yaml | grep data-dir

//- --data-dir=/var/lib/etcd

vim /etc/kubernetes/manifests/etcd.yaml

//change etcd-data path

- hostPath:

path: /var/lib/etcd-backup

type: DirectoryOrCreate

name: etcd-data

journalctl -u kubelet.service | grep -