DumpsCafe

Linux Foundation

CKS

Certified Kubernetes Security Specialist (CKS)

Version: Demo

[Total Questions: 10]

Web: www.dumpscafe.com

Email: support@dumpscafe.com

IMPORTANT NOTICE

Feedback

We have developed quality product and state-of-art service to ensure our customers interest. If you have any suggestions, please feel free to contact us at feedback@dumpscafe.com

Support

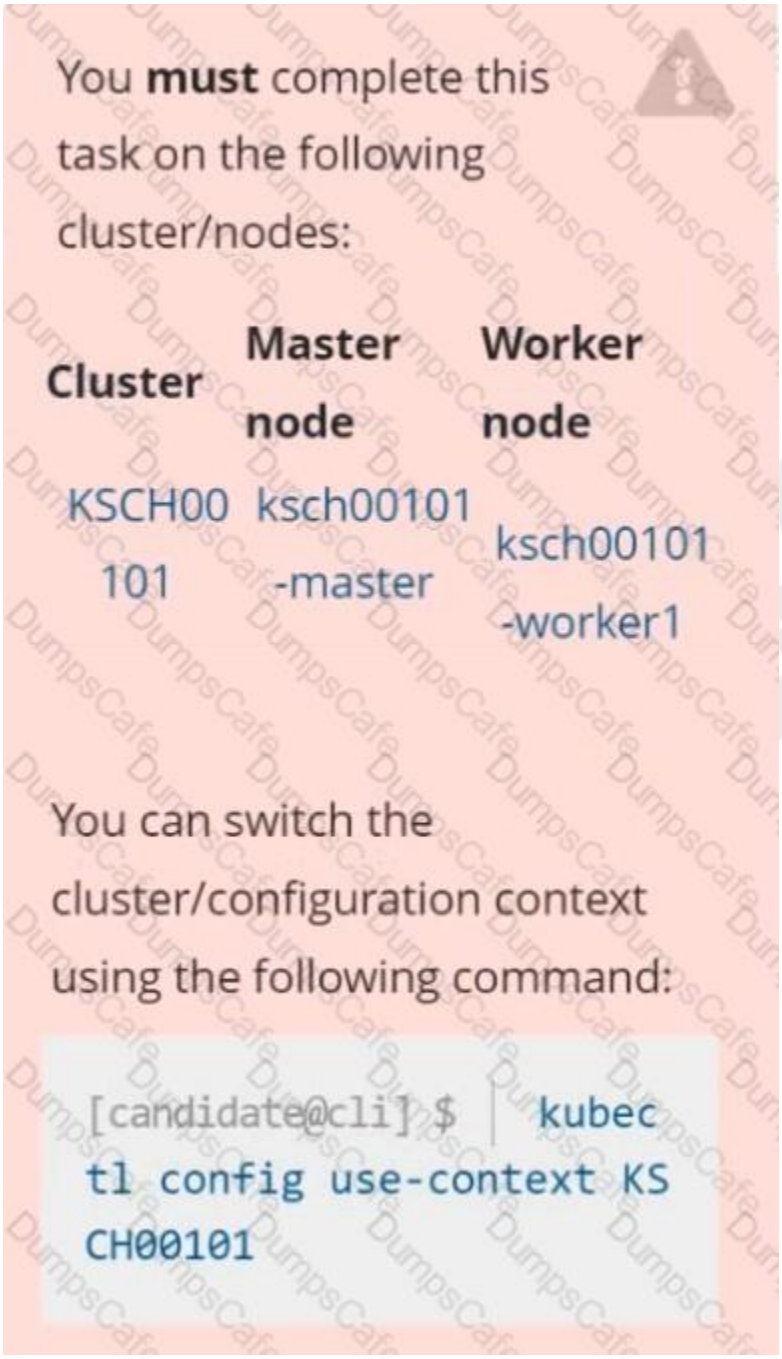
If you have any questions about our product, please provide the following items:

- exam code
- screenshot of the question
- login id/email

please contact us at support@dumpscafe.com and our technical experts will provide support within 24 hours.

Copyright

Question #:1



Context

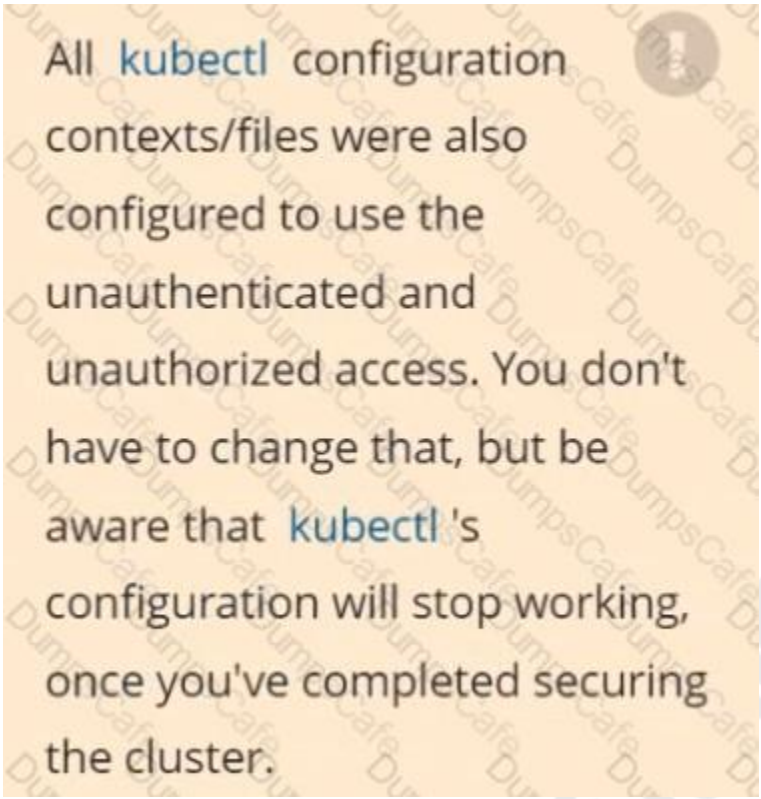
The kubeadm-created cluster's Kubernetes API server was, for testing purposes, temporarily configured to allow unauthenticated and unauthorized access granting the anonymous user duster-admin access.

Task

Reconfigure the cluster's Kubernetes API server to ensure that only authenticated and authorized REST requests are allowed.

Use authorization mode Node,RBAC and admission controller NodeRestriction.

Cleaning up, remove the ClusterRoleBinding for user system:anonymous.

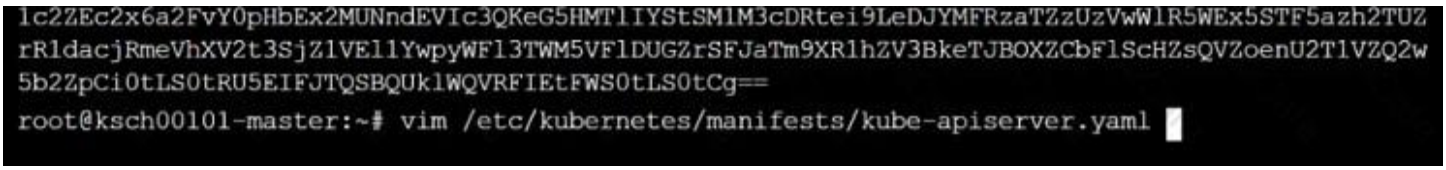


See explanation below.

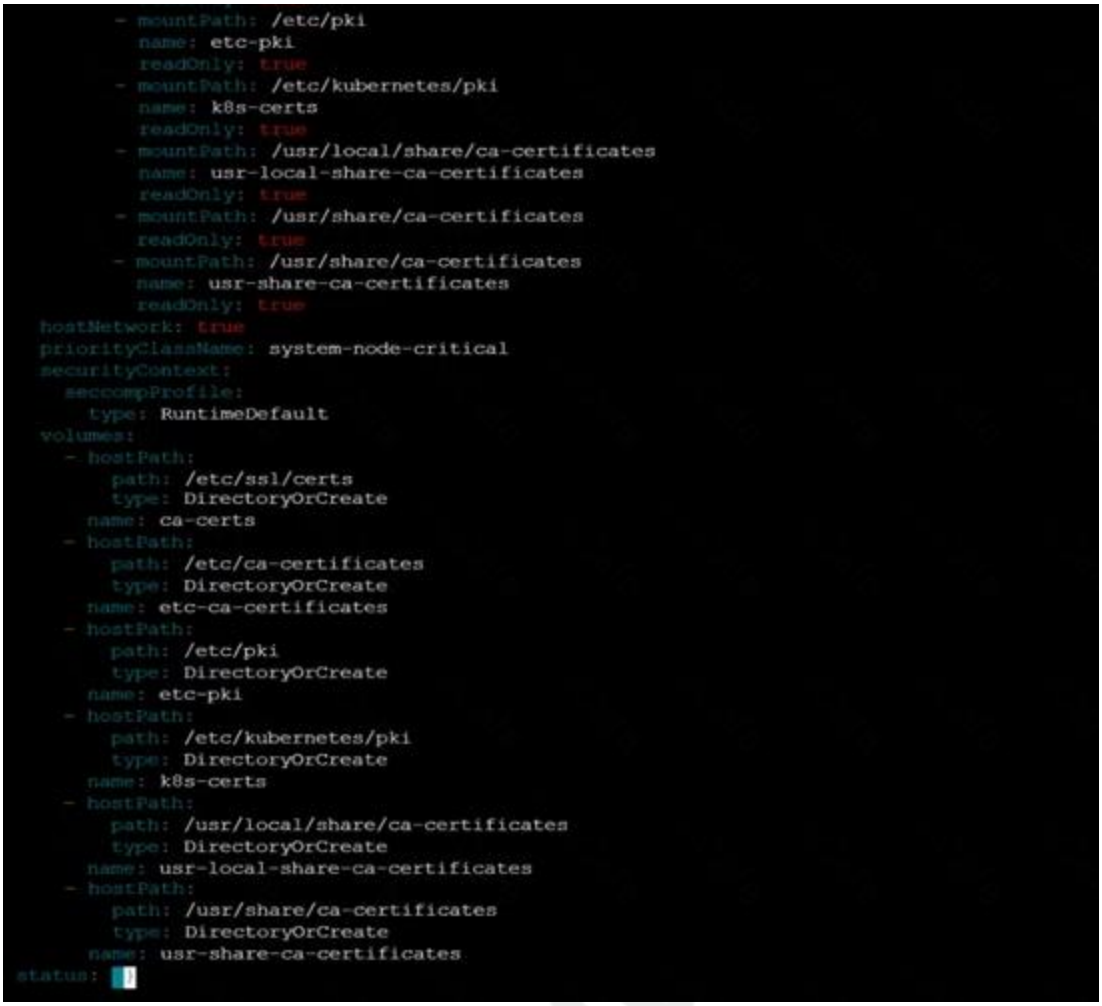
```
candidate@cli:~$ kubectl config use-context KSCH00101
Switched to context "KSCH00101".
candidate@cli:~$ ssh ksch00101-master
Warning: Permanently added '10.240.86.190' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
    - command:
        - kube-apiserver
        - --advertise-address=10.240.86.190
        - --allow-privileged=true
        - --authorization-mode=Node,RBAC
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --enable-admission-plugins=AlwaysAdmit
        - --enable-bootstrap-token-auth=true
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
        - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
        - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
"/etc/kubernetes/manifests/kube-apiserver.yaml" 128L, 4343C                1,1           Top
```

```
root@ksch00101-master:~# cat /etc/kubernetes/admin.conf
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJB
Z01CQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1ESXhO
akF3TlRVeE9Wb1hEVE15TURJeE5EQXdOVFV4T1Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdE
UVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTIvNzNZTktkSFdZU3JUaUx0QStr
N01qTXpRZ1lzM2ttNGl1alpoM0tZc3Y1bUdpN0UyQ2tYc0MKUnhlLlNiZnBDMzlla2k5V3hOSHc5eTM0OEtXUVE3VXBL
UmZRdXVxd1A1WXdDZkord1JmWGNGTXQxLzRNQVhWLwpkdjZ5YWRKSitPeFFSVjZlaHFBZHR0M3FtOFdVcW84UE5JT1E0
OEc3WWhnRUg5RHU3SFdkMS8raXVkSjNOMXl6CnNISEdtYklsWENSbEcydFV0M2RScDczSnRIS1JjS2tnMGxYM3FWS1Uy
QmJRblBmK01wb0VlTXFGcmZvcWVaVWcKYlBKK3ROVmZIM1JLTkhVUnYydVJIa3ZZc2JrclhUMW8rMXFNNHZrYnFNMHlq
KzNxTUtiSyt5V3dzUTlBYUVPMApUdXR4UUdlTFp3OUE3TjZZeTFVQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURB
Z0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZEcUlwLzdYbzZaNkJNVjVEK2w3bFZPcGpBOWlN
QlVHQTFVZEVRUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBSlNWNm9wNGgxYkNv
eGZLRUZ4bwoxaVlHUFlnM1hhOTN0WEZ1TTY3RnA2NkdqUEc5SXBONnNHUnRnWVlyd0Mya1BDeFVOb2IySWtUQ1FNbDV3
cWRHCkdPS2JwVVp6Smc3Y0dyS2E3RlpZWVNyVUVGRWhyd2xZWXNGME56aFBoZVcwcHJjcWtSdXN1bm55SG5YNGVOMUoK
NlNzbGZYTjJIdVFJdlVIRGl5L0JsL1ZWRmZNZnRxOGF0Z0pYSFZGTmlVcDRpNXlJTXFRNTB4ZjVqcnFlWFRmVwpVdmJq
ZjEyOThXVTk3QkxHcDdRZE9QYWVKU051UStlVkMrdnpVZ2tVQVNjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3UlhLbGcvangvcitqd0dTc0xwWUxDZTlxa1FraTdCSVRJT1N3ejd3c2hzbERuNzBFY0IKa0VBPQotLS0tLUVO
RCBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://10.240.86.190:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ01
ocEdQcDB4Zk9JbkYxaGJwcTh5YlBUMGx1Tm5VNjBiSUpxRXVKckxJbEtXClNVa1h1VkYzNkl0ZHc1ZU1OT2JxK1haaHd
hY2JURVZCM1VDVURsbDgzdG5teFQyVXJmY0pUQmhLTCtZTFAvcWYKdjdXR3BwQ1ZXNnhVZGFibGNuUklMnpleUVJTEt
Tck5XbUQ0TzZsMU13blZ0OVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExKL3hiSDdUTzkwYlRFdlIwazl3cFVYdllkdkljSXN
MRkYwL3F2bDA3U3lxbGplOEI1SnNpQlhCUlZxbS9wCmNUUSs3SnZlbmdaZzlkOWdZaVJVdFFTcHBONkx4UnhkSzNKMGR
BK240SWxFZEtHRWh3TE00d0tMaldERG9scHgKYzB3WHkwVXBORGZ6UUxuRUFzVUJsbDRCQ3VkdW5QNVVDN2FuS3dJREF
RQUJBb01CQURWRkZNSVRqYnNySTZTTwpQOGM0MTByN3RWZ251cXJVS202dHRnZWtXOWdlSlpvMnZyb3RsbG9qOGFRamF
0MTZnaEUwOXdZd2xMSDhId0tLCklMb2NrZnFCUytlOWo1Zm1FWGxYTG00cElCVDFRbGFJQ1JRMDRyQ0JZbHdCN1VFbVB
lWjhuQ3lmR2JYTC9HM2wKcXBYTDVKdzJqcVh2MXdzcWsrdWNCRk0zZ0FYZk5YZkh1RExnV0VyNXRZRlF4VXo5UFFHODl
pcDY1OTBkYnB1SApOMnU2NGk4UTgldk83OFVITlc2eUFZU1loZVdha093RDFwZzNPdkhxV3FhbnVlMnlrOWxaUUR0WW5
2MytBeU5DCnloNlRaRHluZ01ZdEptbDFTQ01TNEpSR2d4NXNwaCtKOC9XOGx0Ri9wMWZxbTA0bXZSRndxU3M2Y1JCQ2Z
PVVcKbFV1MGxLRUNnWUVBNWJzT0lVVzFBVndjTmJsc0pSVDNURkI2OVlxbDRYcnZRR0FZY3BhdktENnd5VmtEOTV1QQp
SaXVRSlNNKzY4REtBVmlpYllpaThJemExTkdqdC9JZDUwTGVoNklaRVg2enVpK0g3dlBSbVd6SE9ueWNmU2FmClVQMEF
RL0RiM2lCNWJQTmJHYXNkaDNIb2JvR0NSSHZmTFFXY2tYbUVXM2ZudVlIRlJLZ2x1TEVDZllFQTVUdysKTEVTV1BESFF
mamNBN0htNmdsMndGRjdCUG1sSGdaYVVRN25Eb3ZvRmMxa1BMRWVCMWJ6OHJNWld1eGdmaHN0OQpMZ0xSUDBXdkJWdlJ
sVTdMTmFLTlVzRmkxU2dvaWZsS01ZWkMyZmpLWTY1RFE3YUUzcTdnVis4U2pIZHpoclhCCkVQclAvWXQ3S0QrbFBMZmh
aNXNKZWFtelY3b3gveno5Y0s0U0ZKc0NnWUJ4OVk2VzFydHBoMFcvS05JS3V4SEoKMjRxRFQxbm1ObE9FdmFhakFUaTJ
ZQkxXYnIvWERsNWRjTEs4bFcxYkNYR3JwY2s3U0xKN1hZUVlXajQ2dTNJMgpEQ2ZUWlFiRWRQTzNBbWtPR2ZqWmdPcDd
pdUVkL0JDLzNpRkprcXVlenNFdFdMTHlVcjM5T0hZeWl6QVJ4Tmo2CnZuUGlma000Rkl6d3F2MHVoN0xlblFLQmdBTlZ
XZTRZM1RwbzJ3aEswbmVkMllsMXhVNjJoZ2JiVHcvaVdhdVcKY3ZMV3dlZUlmd0Q4MVRWL2R3a29KVEM1VEJRUXQzUkk
xRFFnVmtnMHFwcUtOOGhDNGQwM05MRzIwTWdZMk94WgpjSFZzK2J4elYwVVB6V1RUbEMyVEsyamhmOHVRcndzSktxY2N
OU0ZEczZwclhocThsV213Znd3aWlBR1hLSFJRCkE3RkxBb0dCQUx3NW8rbHFVZ3hHQlpKdy9EelRGek5TekQreVd6Um8
```


lc2ZEc2x6a2FvY0pHbEx2MUNndEVIc3QKeG5HMTlIYStSMlM3cDRtei9LeDJYMFRzaTZzUzVwWlR5WEx5STF5azh2TUZ
rR1dacjRmeVhXV2t3SjZ1VEl1YwpyWFl3TWM5VFlDUGZrSFJaTm9XR1hZV3BkeTJBOXZCbFlScHZsQVZoenU2T1VZQ2w
5b2ZpCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml


apiVersion: v1
kind: Pod
metadata:
annotations:
kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
creationTimestamp: null
labels:
component: kube-apiserver
tier: control-plane
name: kube-apiserver
namespace: kube-system
spec:
containers:
- command:
- kube-apiserver
- --advertise-address=10.240.86.190
- --allow-privileged=true
- --authorization-mode=Node,RBAC
- --client-ca-file=/etc/kubernetes/pki/ca.crt
- --enable-admission-plugins=NodeRestriction
- --enable-bootstrap-token-auth=true
- --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
- --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
- --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
- --etcd-servers=https://127.0.0.1:2379
- --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
- --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
- --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
- --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
- --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
- --requestheader-allowed-names=front-proxy-client
- --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
- --requestheader-extra-headers-prefix=X-Remote-Extra-
- --requestheader-group-headers=X-Remote-Group
- --requestheader-username-headers=X-Remote-User
- --secure-port=6443
- --service-account-issuer=https://kubernetes.default.svc.cluster.local
- --service-account-key-file=/etc/kubernetes/pki/sa.pub
- --service-account-signing-key-file=/etc/kubernetes/pki/sa.key
- --service-cluster-ip-range=10.96.0.0/12
- --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
- --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
- --anonymous-auth=false
image: k8s.gcr.io/kube-apiserver:v1.23.3
imagePullPolicy: IfNotPresent
livenessProbe:
failureThreshold: 8
httpGet:
host: 10.240.86.190
path: /livez
port: 6443
scheme: HTTPS
initialDelaySeconds: 10
periodSeconds: 10
timeoutSeconds: 15
name: kube-apiserver
readinessProbe:
failureThreshold: 3
httpGet:
host: 10.240.86.190
path: /readyz
port: 6443
scheme: HTTPS
periodSeconds: 1
timeoutSeconds: 15
resources:
requests:
cpu: 250m
startupProbe:
failureThreshold: 24
httpGet:
host: 10.240.86.190
path: /livez
port: 6443
scheme: HTTPS
initialDelaySeconds: 10
periodSeconds: 10
timeoutSeconds: 15
volumeMounts:
- mountPath: /etc/ssl/certs
name: ca-certs
readOnly: true
- mountPath: /etc/ca-certificates
name: etc-ca-certificates
readOnly: true


- mountPath: /etc/pki
name: etc-pki
readOnly: true
- mountPath: /etc/kubernetes/pki
name: k8s-certs
readOnly: true
- mountPath: /usr/local/share/ca-certificates
name: usr-local-share-ca-certificates
readOnly: true
- mountPath: /usr/share/ca-certificates
readOnly: true
- mountPath: /usr/share/ca-certificates
name: usr-share-ca-certificates
readOnly: true
hostNetwork: true
priorityClassName: system-node-critical
securityContext:
seccompProfile:
type: RuntimeDefault
volumes:
- hostPath:
path: /etc/ssl/certs
type: DirectoryOrCreate
name: ca-certs
- hostPath:
path: /etc/ca-certificates
type: DirectoryOrCreate
name: etc-ca-certificates
- hostPath:
path: /etc/pki
type: DirectoryOrCreate
name: etc-pki
- hostPath:
path: /etc/kubernetes/pki
type: DirectoryOrCreate
name: k8s-certs
- hostPath:
path: /usr/local/share/ca-certificates
type: DirectoryOrCreate
name: usr-local-share-ca-certificates
- hostPath:
path: /usr/share/ca-certificates
type: DirectoryOrCreate
name: usr-share-ca-certificates
status:

```
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@ksch00101-master:~# systemctl daemon-reload
sroot@ksch00101-master:~# systemctl restart kubelet.service
root@ksch00101-master:~# kubectl get nodes
error: You must be logged in to the server (Unauthorized)
root@ksch00101-master:~# exit
logout
Connection to 10.240.86.190 closed.
candidate@cli:~$ kubectl get nodes
NAME               STATUS   ROLES                  AGE   VERSION
ksch00101-master   Ready    control-plane,master   93d   v1.23.3
ksch00101-worker1  Ready    <none>                 93d   v1.23.3
candidate@cli:~$ kubectl get pod -n kube-system
NAME                                       READY   STATUS    RESTARTS       AGE
coredns-64897985d-7pnhm                    1/1     Running   1 (7h2m ago)   93d
coredns-64897985d-rr7sd                    1/1     Running   1 (7h2m ago)   93d
etcd-ksch00101-master                      1/1     Running   1 (7h2m ago)   93d
kube-apiserver-ksch00101-master            0/1     Running   0              24s
kube-controller-manager-ksch00101-master   1/1     Running   3 (42s ago)    93d
kube-flannel-ds-llktn                      1/1     Running   1 (93d ago)    93d
kube-flannel-ds-q9vnl                      1/1     Running   1 (93d ago)    93d
kube-proxy-2c4ht                           1/1     Running   1 (93d ago)    93d
kube-proxy-pmmbc                           1/1     Running   1 (93d ago)    93d
kube-scheduler-ksch00101-master            1/1     Running   3 (42s ago)    93d
candidate@cli:~$ kubectl get pod -n kube-system
NAME                                       READY   STATUS    RESTARTS       AGE
coredns-64897985d-7pnhm                    1/1     Running   1 (7h2m ago)   93d
coredns-64897985d-rr7sd                    1/1     Running   1 (7h2m ago)   93d
etcd-ksch00101-master                      1/1     Running   1 (7h2m ago)   93d
kube-apiserver-ksch00101-master            0/1     Running   0              30s
kube-controller-manager-ksch00101-master   1/1     Running   3 (48s ago)    93d
kube-flannel-ds-llktn                      1/1     Running   1 (93d ago)    93d
kube-flannel-ds-q9vnl                      1/1     Running   1 (93d ago)    93d
kube-proxy-2c4ht                           1/1     Running   1 (93d ago)    93d
kube-proxy-pmmbc                           1/1     Running   1 (93d ago)    93d
kube-scheduler-ksch00101-master            1/1     Running   3 (48s ago)    93d
candidate@cli:~$ kubectl get clusterrolebindings.rbac.authorization.k8s.io | grep anon
system:anonymous                                       ClusterRole/cluster-admin
                                                 7h1m
candidate@cli:~$ kubectl delete clusterrolebindings.rbac.authorization.k8s.io/system:anonymo
us
```

```
clusterrolebinding.rbac.authorization.k8s.io "system:anonymous" deleted
```

Question #:2

use the Trivy to scan the following images,

1. amazonlinux:1

2. k8s.gcr.io/kube-controller-manager:v1.18.6

Look for images with HIGH or CRITICAL severity vulnerabilities and store the output of the same in /opt/trivy-vulnerable.txt

Send us your suggestion on it.

Question #:3

Cluster: qa-cluster

Master node: master Worker node: worker1

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context qa-cluster

Task:

Create a NetworkPolicy named restricted-policy to restrict access to Pod product running in namespace dev.

Only allow the following Pods to connect to Pod products-service:

1. Pods in the namespace qa

2. Pods with label environment: stage, in any namespace

See the Explanation below.

Explanation

Text Description automatically generated

```
candidate@cli:~$ kubectl config use-context KSSH00301
Switched to context "KSSH00301".
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl get ns dev-team --show-labels
NAME       STATUS   AGE     LABELS
dev-team   Active   6h39m   environment=dev,kubernetes.io/metadata.name=dev-team
candidate@cli:~$ kubectl get pods -n dev-team --show-labels
NAME            READY   STATUS    RESTARTS   AGE     LABELS
users-service   1/1     Running   0          6h40m   environment=dev
candidate@cli:~$ ls
KSCH00301  KSMV00102  KSSC00301  KSSH00401     test-secret-pod.yaml
KSCS00101  KSMV00301  KSSH00301  password.txt  username.txt
candidate@cli:~$ vim np.yaml
```

Text Description automatically generated

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
```

Text Description automatically generated

```
candidate@cli:~$ vim np.yaml
candidate@cli:~$ cat np.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl create -f np.yaml -n dev-team
networkpolicy.networking.k8s.io/pod-access created
candidate@cli:~$ kubectl describe netpol -n dev-team
Name:         pod-access
Namespace:    dev-team
Created on:   2022-05-20 15:35:33 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  PodSelector:     environment=dev
  Allowing ingress traffic:
    To Port: <any> (traffic allowed to all ports)
    From:
      NamespaceSelector: environment=dev
    From:
      PodSelector: environment=testing
  Not affecting egress traffic
  Policy Types: Ingress
candidate@cli:~$ cat KSSH00301/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ""
  namespace: ""
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
```

```
    - from: []
    - from: []
candidate@cli:~$ cp np.yaml KSSH00301/network-policy.yaml
candidate@cli:~$ cat KSSH00301/network-policy.yaml
```

Text Description automatically generated

```
candidate@cli:~$ cat KSSH00301/network-policy.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
candidate@cli:~$
```

Question #:4

Cluster: dev

Master node: **master1**

Worker node: **worker1**

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context dev

Task:

Retrieve the content of the existing secret named **adam** in the **safe** namespace.

Store the username field in a file names **/home/cert-masters/username.txt**, and the password field in a file named **/home/cert-masters/password.txt**.

1. You must create both files; they don't exist yet.

2. Do not use/modify the created files in the following steps, create new temporary files if needed.

Create a new secret names **newsecret** in the **safe** namespace, with the following content:

Username: **dbadmin**

Password: **moresecurepas**

Finally, create a new Pod that has access to the secret **newsecret** via a volume:

- Namespace:safe
- Pod name:mysecret-pod
- Container name:db-container
- Image:redis
- Volume name:secret-vol
- Mount path:/etc/mysecret

See the explanation below

Explanation

Text Description automatically generated

```
candidate@cli:~$ kubectl config use-context KSMV00201
Switched to context "KSMV00201".
candidate@cli:~$ kubectl get secret -n monitoring
NAME                  TYPE                                  DATA   AGE
db1-test              Opaque                                2      6h23m
default-token-cqqf6   kubernetes.io/service-account-token   3      6h23m
candidate@cli:~$ kubectl get secret/db1-test -n monitoring
NAME       TYPE     DATA   AGE
db1-test   Opaque   2      6h23m
candidate@cli:~$ kubectl get secret/db1-test -n monitoring -o yaml
apiVersion: v1
data:
  password: QVU3dHh1bXFOTHZt
  username: cHJvZHVjdGlvbi0x
kind: Secret
metadata:
  creationTimestamp: "2022-05-20T08:37:33Z"
  name: db1-test
  namespace: monitoring
  resourceVersion: "2588"
  uid: 659bd4ac-e0ba-4d9f-b411-816f2aedf7e6
type: Opaque
candidate@cli:~$ echo "cHJvZHVjdGlvbi0x" | base64 -d
production-1candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ echo "cHJvZHVjdGlvbi0x" | base64 -d  > /home/candidate/username.txt
candidate@cli:~$ cat /home/candidate/username.txt
production-1candidate@cli:~$
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ echo "QVU3dHh1bXFOTHZt" | base64 -d
AU7txumqNLvmcandidate@cli:~$ echo "QVU3dHh1bXFOTHZt" | base64 -d > /home/candidate/password.
txt
candidate@cli:~$ cat /home/candidate/password.txt
AU7txumqNLvmcandidate@cli:~$
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl create secret generic test-workflow --from-literal=username=dev-dat
abase --from-literal=password=aV7HR7nU3JLx -n monitoring
secret/test-workflow created
candidate@cli:~$
candidate@cli:~$
```

```
candidate@cli:~$ kubectl  -n monitoring run test-secret-pod --image=httpd --dry-run=client -
o yaml > test-secret-pod.yaml
candidate@cli:~$ vim test-secret-pod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: test-secret-pod
  name: test-secret-pod
  namespace: monitoring
spec:
  volumes:
    - name: dev-volume
      secret:
        secretName: test-workflow
  containers:
  - image: httpd
    name: dev-container
    resources: {}
    volumeMounts:
      - name: dev-volume
        mountPath: /etc/credentials
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

Text Description automatically generated

```
candidate@cli:~$ kubectl  -n monitoring run test-secret-pod --image=httpd --dry-run=client -
o yaml > test-secret-pod.yaml
candidate@cli:~$ vim test-secret-pod.yaml
candidate@cli:~$ cat test-secret-pod.yaml
```

Text Description automatically generated

```
  labels:
    run: test-secret-pod
  name: test-secret-pod
  namespace: monitoring
spec:
  volumes:
    - name: dev-volume
      secret:
        secretName: test-workflow
  containers:
  - image: httpd
    name: dev-container
    resources: {}
    volumeMounts:
      - name: dev-volume
        mountPath: /etc/credentials
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
candidate@cli:~$ kubectl create -f test-secret-pod.yaml
pod/test-secret-pod created
candidate@cli:~$ kubectl get pods -n monitoring
NAME              READY   STATUS    RESTARTS   AGE
test-secret-pod   1/1     Running   0          9s
candidate@cli:~$
```

Question #:5

Enable audit logs in the cluster, To Do so, enable the log backend, and ensure that

- 1. logs are stored at /var/log/kubernetes/kubernetes-logs.txt.
- 2. Log files are retained for 5 days.
- 3. at maximum, a number of 10 old audit logs files are retained.

Edit and extend the basic policy to log:

- 1. Cronjobs changes at RequestResponse
- 2. Log the request body of deployments changes in the namespace kube-system.

- 3. Log all other resources in core and extensions at the Request level.
- 4. Don't log watch requests by the "system:kube-proxy" on endpoints or

See explanation below.

Explanation

Text Description automatically generated

```
candidate@cli:~$ kubectl config use-context KSRS00602
Switched to context "KSRS00602".
candidate@cli:~$ ssh ksrs00602-master
Warning: Permanently added '10.240.86.243' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@ksrs00602-master:~# cat /etc/kubernetes/logpolicy/sample-policy.yaml
---
apiVersion: audit.k8s.io/v1
kind: Policy
# Don't generate audit events for all requests in RequestReceived stage.
omitStages:
  - "RequestReceived"
rules:
  # Don't log watch requests by the "system:kube-proxy" on endpoints or services
  - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
    - group: "" # core API group
      resources: ["endpoints", "services"]

  # Don't log authenticated requests to certain non-resource URL paths.
  - level: None
    userGroups: ["system:authenticated"]
    nonResourceURLs:
    - "/api*" # Wildcard matching.
    - "/version"
  # Edit form here below
root@ksrs00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
```

Text Description automatically generated

```
    - "/api*" # Wildcard matching.
    - "/version"
  # Edit form here below
  - level: RequestResponse
    resources:
    - group: ""
      resources: ["cronjobs"]
  - level: Request
    resources:
    - group: "" # core API group
      resources: ["pods"]
      namespaces: ["webapps"]
# Log configmap and secret changes in all other namespaces at the Metadata level.
  - level: Metadata
    resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
```

Text Description automatically generated

```
    - "/version"
  # Edit form here below
  - level: RequestResponse
    resources:
    - group: ""
      resources: ["cronjobs"]
  - level: Request
    resources:
    - group: "" # core API group
      resources: ["pods"]
      namespaces: ["webapps"]
# Log configmap and secret changes in all other namespaces at the Metadata level.
  - level: Metadata
    resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
root@ksrs00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
root@ksrs00602-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

Text Description automatically generated

```
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
    - command:
        - kube-apiserver
        - --advertise-address=10.240.86.243
        - --allow-privileged=true
        - --audit-policy-file=/etc/kubernetes/logpolicy/sample-policy.yaml
        - --audit-log-path=/var/log/kubernetes/kubernetes-logs.txt
        - --audit-log-maxbackup=1
        - --audit-log-maxage=30
        - --authorization-mode=Node,RBAC
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --enable-admission-plugins=NodeRestriction
        - --enable-bootstrap-token-auth=true
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
```

Text Description automatically generated

```
  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
root@ksrs00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
root@ksrs00602-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@ksrs00602-master:~# systemctl daemon-reload
root@ksrs00602-master:~# systemctl restart kubelet.service
root@ksrs00602-master:~# systemctl enable kubelet
root@ksrs00602-master:~# exit
logout
Connection to 10.240.86.243 closed.
candidate@cli:~$
```

Question #:6

Task

Create a NetworkPolicy named pod-access to restrict access to Pod users-service running in namespace dev-team.

Only allow the following Pods to connect to Pod users-service:

Pods in the namespace qa ▪

Pods with label environment: testing, in any namespace ▪

Make sure to apply the NetworkPolicy.

You can find a skeleton manifest file at /home/candidate/KSSH00301/network-policy.yaml

See explanation below.

Explanation

Text Description automatically generated

```
candidate@cli:~$ kubectl config use-context KSSH00301
Switched to context "KSSH00301".
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl get ns dev-team --show-labels
NAME       STATUS   AGE     LABELS
dev-team   Active   6h39m   environment=dev,kubernetes.io/metadata.name=dev-team
candidate@cli:~$ kubectl get pods -n dev-team --show-labels
NAME            READY   STATUS    RESTARTS   AGE     LABELS
users-service   1/1     Running   0          6h40m   environment=dev
candidate@cli:~$ ls
KSCH00301  KSMV00102  KSSC00301  KSSH00401     test-secret-pod.yaml
KSCS00101  KSMV00301  KSSH00301  password.txt  username.txt
candidate@cli:~$ vim np.yaml
```

Text Description automatically generated

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
```

Text Description automatically generated

```
candidate@cli:~$ vim np.yaml
candidate@cli:~$ cat np.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl create -f np.yaml -n dev-team
networkpolicy.networking.k8s.io/pod-access created
candidate@cli:~$ kubectl describe netpol -n dev-team
Name:         pod-access
Namespace:    dev-team
Created on:   2022-05-20 15:35:33 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  PodSelector:     environment=dev
  Allowing ingress traffic:
    To Port: <any> (traffic allowed to all ports)
    From:
      NamespaceSelector: environment=dev
    From:
      PodSelector: environment=testing
  Not affecting egress traffic
  Policy Types: Ingress
candidate@cli:~$ cat KSSH00301/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ""
  namespace: ""
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
```

```
    - from: []
    - from: []
candidate@cli:~$ cp np.yaml KSSH00301/network-policy.yaml
candidate@cli:~$ cat KSSH00301/network-policy.yaml
```

Text Description automatically generated

```
candidate@cli:~$ cat KSSH00301/network-policy.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
candidate@cli:~$
```

Question #:7

Create a RuntimeClass named gvisor-rc using the prepared runtime handler named runsc.

Create a Pods of image Nginx in the Namespace server to run on the gVisor runtime class

See the explanation below:

Explanation

- Install the Runtime Class for gVisor

{ # Step 1: Install a RuntimeClass

cat <<EOF | kubectl apply -f -

```
apiVersion: node.k8s.io/v1beta1
kind: RuntimeClass
metadata:
name: gvisor
handler: runsc
EOF
}
```

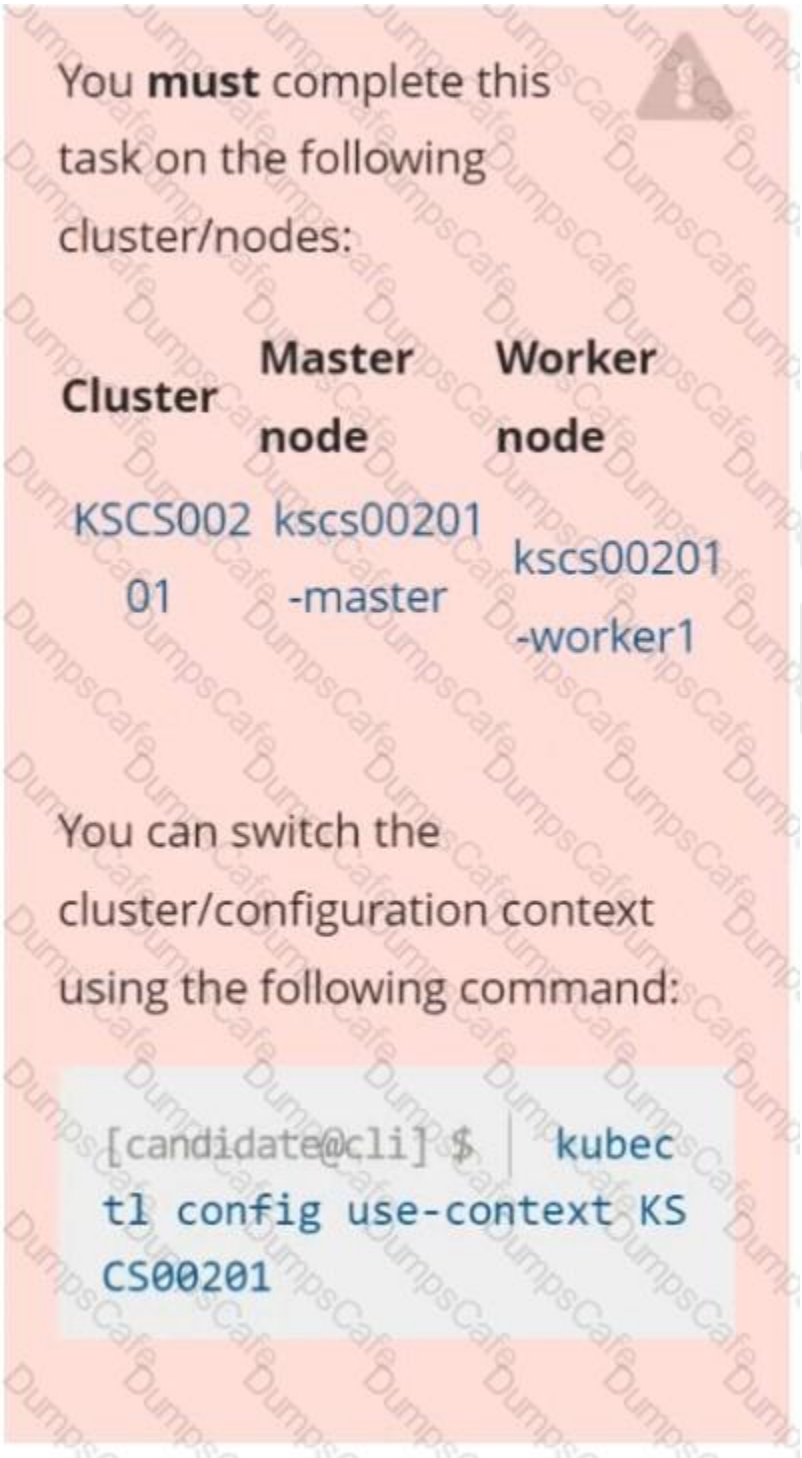
- Create a Pod with the gVisor Runtime Class

```
{ # Step 2: Create a pod
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
name: nginx-gvisor
spec:
runtimeClassName: gvisor
containers:
- name: nginx
image: nginx
EOF
}
```

- Verify that the Pod is running

```
{ # Step 3: Get the pod
kubectl get pod nginx-gvisor -o wide
}
```

Question #:8



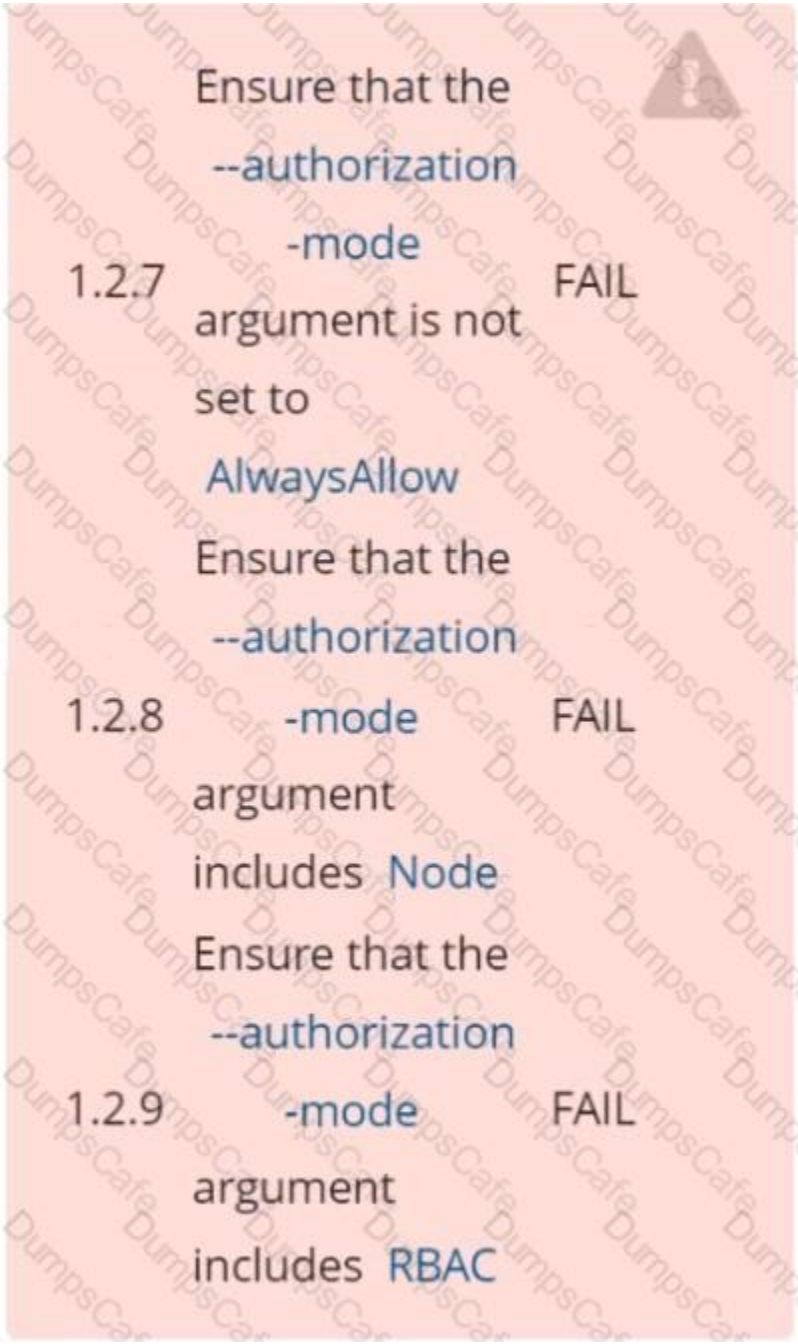
Context

A CIS Benchmark tool was run against the kubeadm-created cluster and found multiple issues that must be addressed immediately.

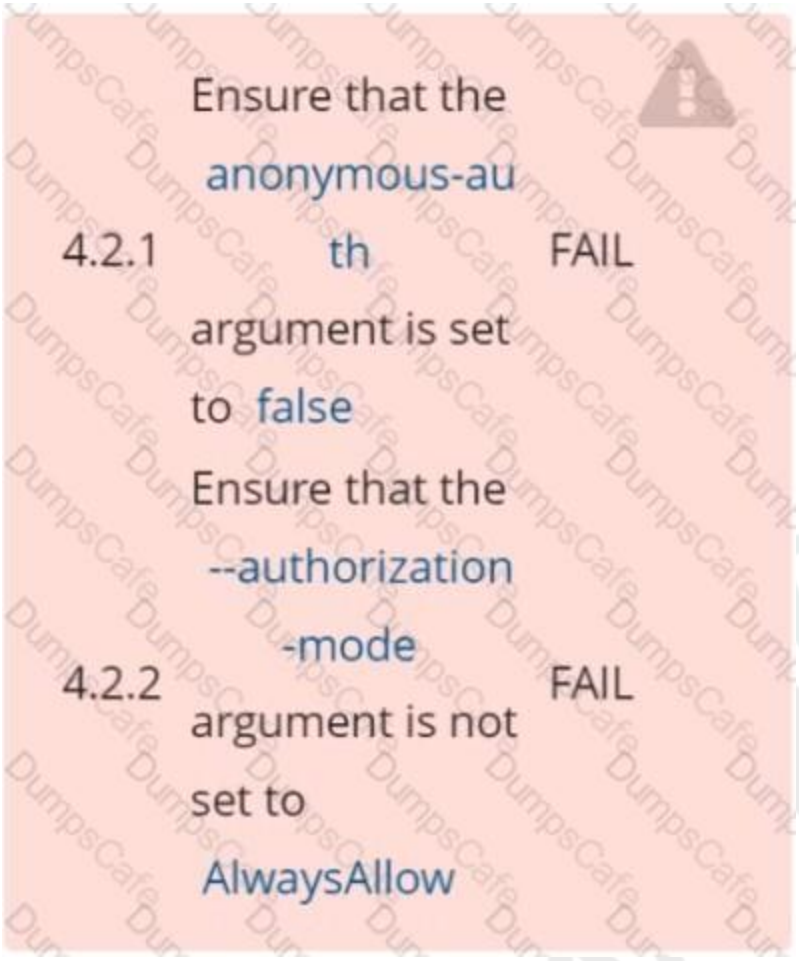
Task

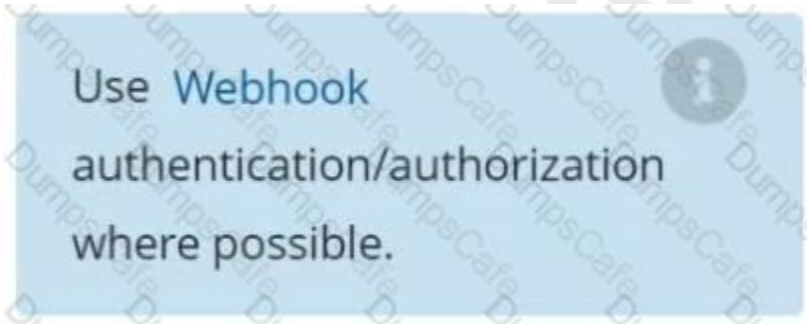
Fix all issues via configuration and restart the affected components to ensure the new settings take effect.

Fix all of the following violations that were found against the API server:

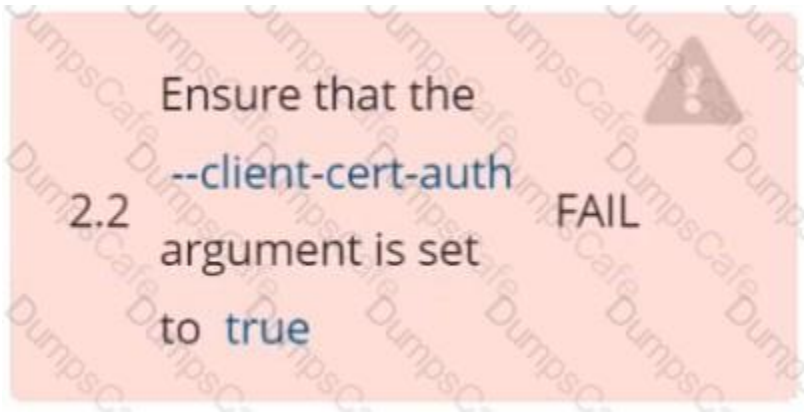


Fix all of the following violations that were found against the Kubelet:



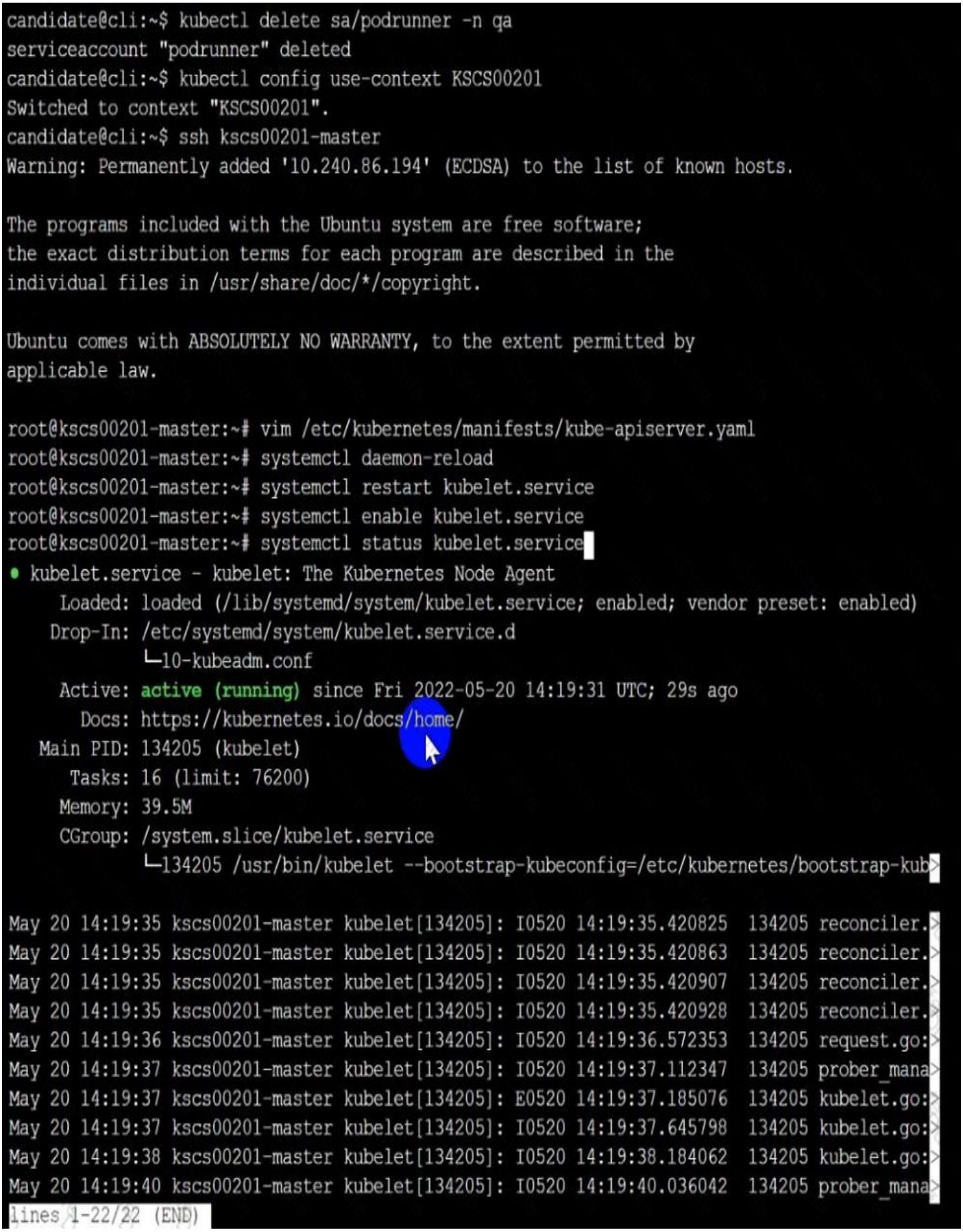


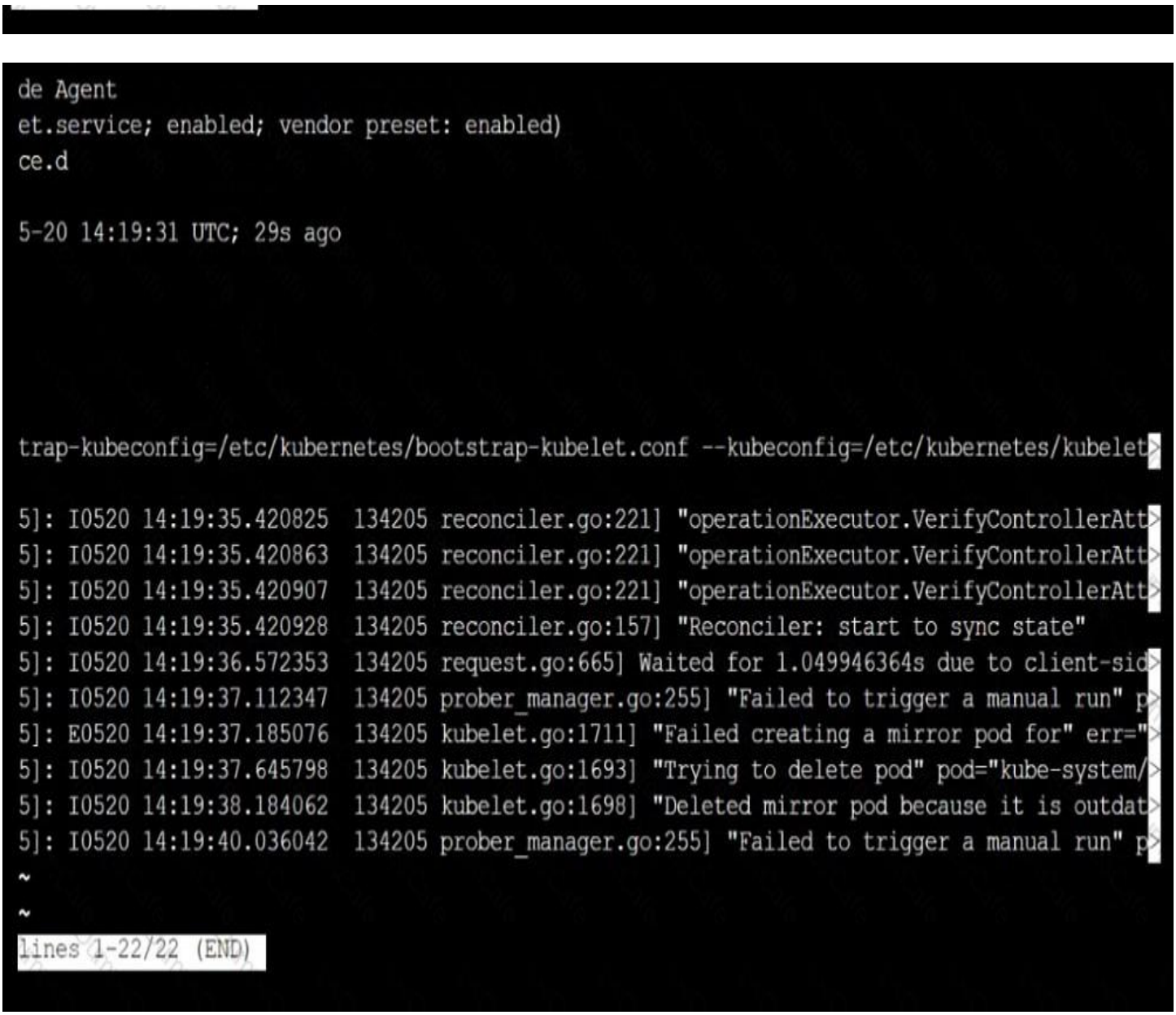
Fix all of the following violations that were found against etcd:



See explanation below.

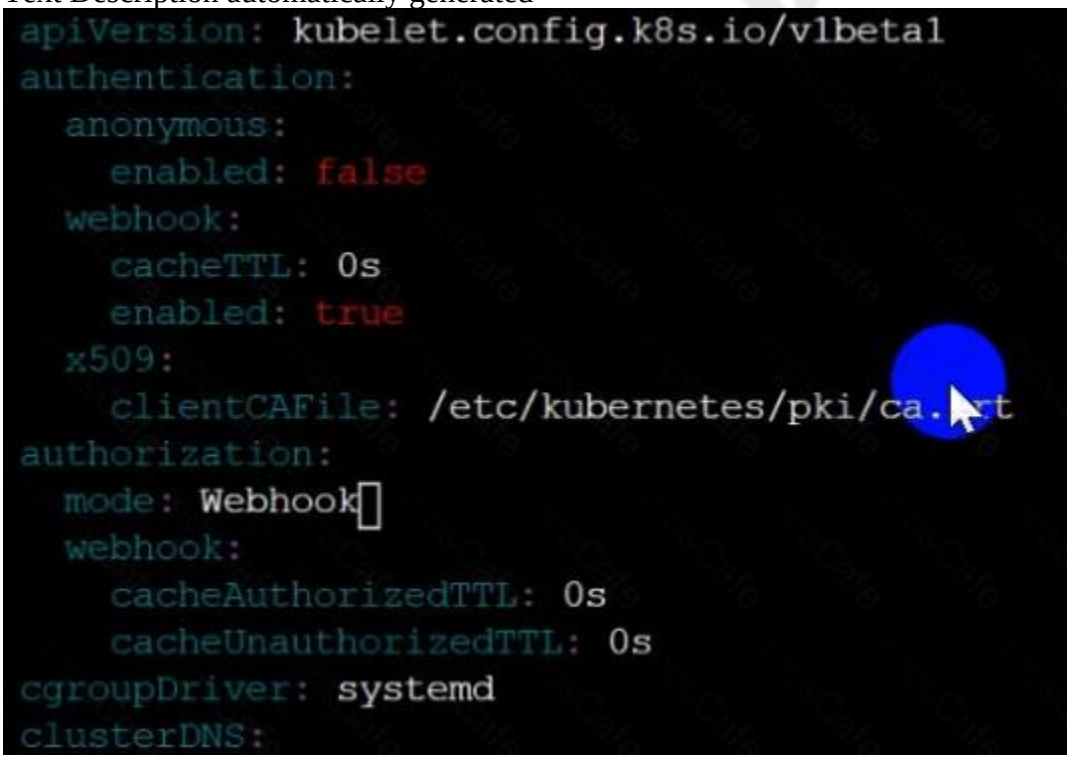
Explanation


candidate@cli:~$ kubectl delete sa/podrunner -n qa
serviceaccount "podrunner" deleted
candidate@cli:~$ kubectl config use-context KSCS00201
Switched to context "KSCS00201".
candidate@cli:~$ ssh kscs00201-master
Warning: Permanently added '10.240.86.194' (ECDSA) to the list of known hosts.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
root@kscs00201-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@kscs00201-master:~# systemctl daemon-reload
root@kscs00201-master:~# systemctl restart kubelet.service
root@kscs00201-master:~# systemctl enable kubelet.service
root@kscs00201-master:~# systemctl status kubelet.service
● kubelet.service - kubelet: The Kubernetes Node Agent
Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
Drop-In: /etc/systemd/system/kubelet.service.d
└10-kubeadm.conf
Active: active (running) since Fri 2022-05-20 14:19:31 UTC; 29s ago
Docs: https://kubernetes.io/docs/home/
Main PID: 134205 (kubelet)
Tasks: 16 (limit: 76200)
Memory: 39.5M
CGroup: /system.slice/kubelet.service
└134205 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kub
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420825 134205 reconciler.
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420863 134205 reconciler.
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420907 134205 reconciler.
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420928 134205 reconciler.
May 20 14:19:36 kscs00201-master kubelet[134205]: I0520 14:19:36.572353 134205 request.go:
May 20 14:19:37 kscs00201-master kubelet[134205]: I0520 14:19:37.112347 134205 prober_mana
May 20 14:19:37 kscs00201-master kubelet[134205]: E0520 14:19:37.185076 134205 kubelet.go:
May 20 14:19:37 kscs00201-master kubelet[134205]: I0520 14:19:37.645798 134205 kubelet.go:
May 20 14:19:38 kscs00201-master kubelet[134205]: I0520 14:19:38.184062 134205 kubelet.go:
May 20 14:19:40 kscs00201-master kubelet[134205]: I0520 14:19:40.036042 134205 prober_mana
lines 1-22/22 (END)


de Agent
et.service; enabled; vendor preset: enabled)
ce.d
5-20 14:19:31 UTC; 29s ago
trap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet
5]: I0520 14:19:35.420825 134205 reconciler.go:221] "operationExecutor.VerifyControllerAtt
5]: I0520 14:19:35.420863 134205 reconciler.go:221] "operationExecutor.VerifyControllerAtt
5]: I0520 14:19:35.420907 134205 reconciler.go:221] "operationExecutor.VerifyControllerAtt
5]: I0520 14:19:35.420928 134205 reconciler.go:157] "Reconciler: start to sync state"
5]: I0520 14:19:36.572353 134205 request.go:665] Waited for 1.049946364s due to client-sid
5]: I0520 14:19:37.112347 134205 prober_manager.go:255] "Failed to trigger a manual run" p
5]: E0520 14:19:37.185076 134205 kubelet.go:1711] "Failed creating a mirror pod for" err="
5]: I0520 14:19:37.645798 134205 kubelet.go:1693] "Trying to delete pod" pod="kube-system/
5]: I0520 14:19:38.184062 134205 kubelet.go:1698] "Deleted mirror pod because it is outdat
5]: I0520 14:19:40.036042 134205 prober_manager.go:255] "Failed to trigger a manual run" p
~
~
lines 1-22/22 (END)

```
let.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kube-proxy\"
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"lib-modules\"
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"flannel-cfg\"
o:157] "Reconciler: start to sync state"
65] Waited for 1.049946364s due to client-side throttling, not priority and fairness, reque
er.go:255] "Failed to trigger a manual run" probe="Readiness"
711] "Failed creating a mirror pod for" err="pods \"kube-apiserver-kscs00201-master\" alrea
693] "Trying to delete pod" pod="kube-system/kube-apiserver-kscs00201-master" podUID=bb91e1
698] "Deleted mirror pod because it is outdated" pod="kube-system/kube-apiserver-kscs00201-
er.go:255] "Failed to trigger a manual run" probe="Readiness"
~
~
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml
```

Text Description automatically generated



Text Description automatically generated

```
~
~
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml
root@kscs00201-master:~# vim /etc/kubernetes/manifests/etcd.yaml
root@kscs00201-master:~# systemctl daemon-reload
root@kscs00201-master:~# systemctl restart kubelet.service
root@kscs00201-master:~# systemctl status kubelet.service
```

```
● kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Fri 2022-05-20 14:22:29 UTC; 4s ago
       Docs: https://kubernetes.io/docs/home/
   Main PID: 135849 (kubelet)
      Tasks: 17 (limit: 76200)
     Memory: 38.0M
     CGroup: /system.slice/kubelet.service
             └─135849 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kub>

May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330232  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330259  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330304  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330354  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330378  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330397  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330415  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330433  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330452  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330463  135849 reconciler.>
lines 1-22/22 (END)
```

Text Description automatically generated

```
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330463  135849 reconciler.
root@kscs00201-master:~#
root@kscs00201-master:~#
root@kscs00201-master:~#
root@kscs00201-master:~# exit
logout
Connection to 10.240.86.194 closed.
candidate@cli:~$
```

Question #:9

Context:

Cluster: **gvisor**

Master node: **master1**

Worker node: **worker1**

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context gvisor

Context: This cluster has been prepared to support runtime handler, runsc as well as traditional one.

Task:

Create a RuntimeClass named **not-trusted** using the prepared runtime handler names **runsc**.

Update all Pods in the namespace server to run on **newruntime**.

See the explanation below

Explanation

1. Create runtime class by the name of not-trusted using runsc handler

```
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: not-trusted
handler: runsc
```

2. Find all the pods/deployment and edit runtimeClassName parameter to not-trusted under spec

[desk@cli] $ `k edit deploy nginx`

```
spec:
  runtimeClassName: not-trusted. # Add this
```

Explanation[desk@cli] $vim runtime.yaml

- apiVersion: node.k8s.io/v1
- kind: RuntimeClass
- metadata:
- name: not-trusted
- handler: runsc

[desk@cli] $ k apply -f runtime.yaml[desk@cli] $ k get pods

- NAME READY STATUS RESTARTS AGE
- nginx-6798fc88e8-chp6r 1/1 Running 0 11m
- nginx-6798fc88e8-fs53n 1/1 Running 0 11m
- nginx-6798fc88e8-ndved 1/1 Running 0 11m

[desk@cli] $ k get deploy

- NAME READY UP-TO-DATE AVAILABLE AGE
- nginx 3/3 11 3 5m

[desk@cli] $ k edit deploy nginx

Text Description automatically generated

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      labels:
        app: nginx
    spec:
      runtimeClassName: not-trusted     # Add this
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
```

Question #:10

Using the runtime detection tool Falco, Analyse the container behavior for at least 20 seconds, using filters that detect newly spawning and executing processes in a single container of Nginx.

store the incident file art /opt/falco-incident.txt, containing the detected incidents. one per line, in the format

[timestamp],[uid],[processName]

Send us your feedback on it.

About dumpscafe.com

dumpscafe.com was founded in 2007. We provide latest & high quality IT / Business Certification Training Exam Questions, Study Guides, Practice Tests.

We help you pass any IT / Business Certification Exams with 100% Pass Guaranteed or Full Refund. Especially Cisco, CompTIA, Citrix, EMC, HP, Oracle, VMware, Juniper, Check Point, LPI, Nortel, EXIN and so on.

View list of all certification exams: All vendors

We prepare state-of-the art practice tests for certification exams. You can reach us at any of the email addresses listed below.

- Sales: sales@dumpscafe.com
- Feedback: feedback@dumpscafe.com
- Support: support@dumpscafe.com

Any problems about IT certification or our products, You can write us back and we will get back to you within 24 hours.