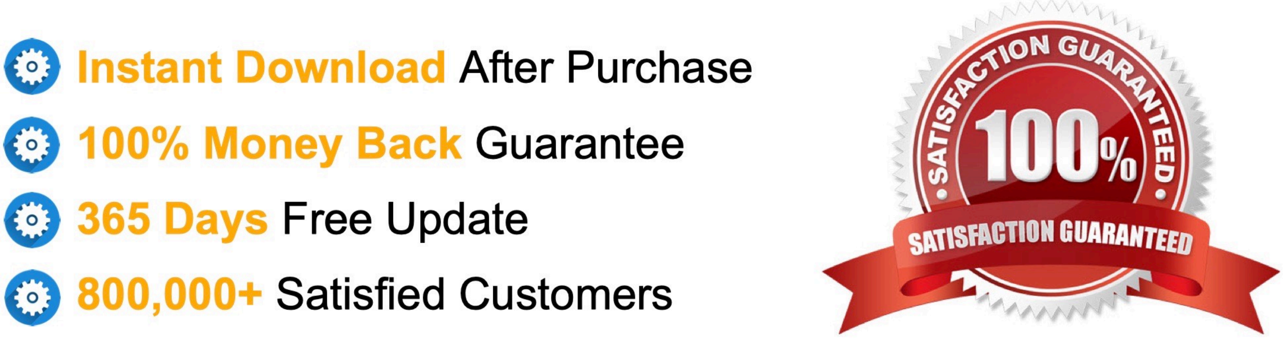
CKS Q&As

Certified Kubernetes Security Specialist (CKS) Exam

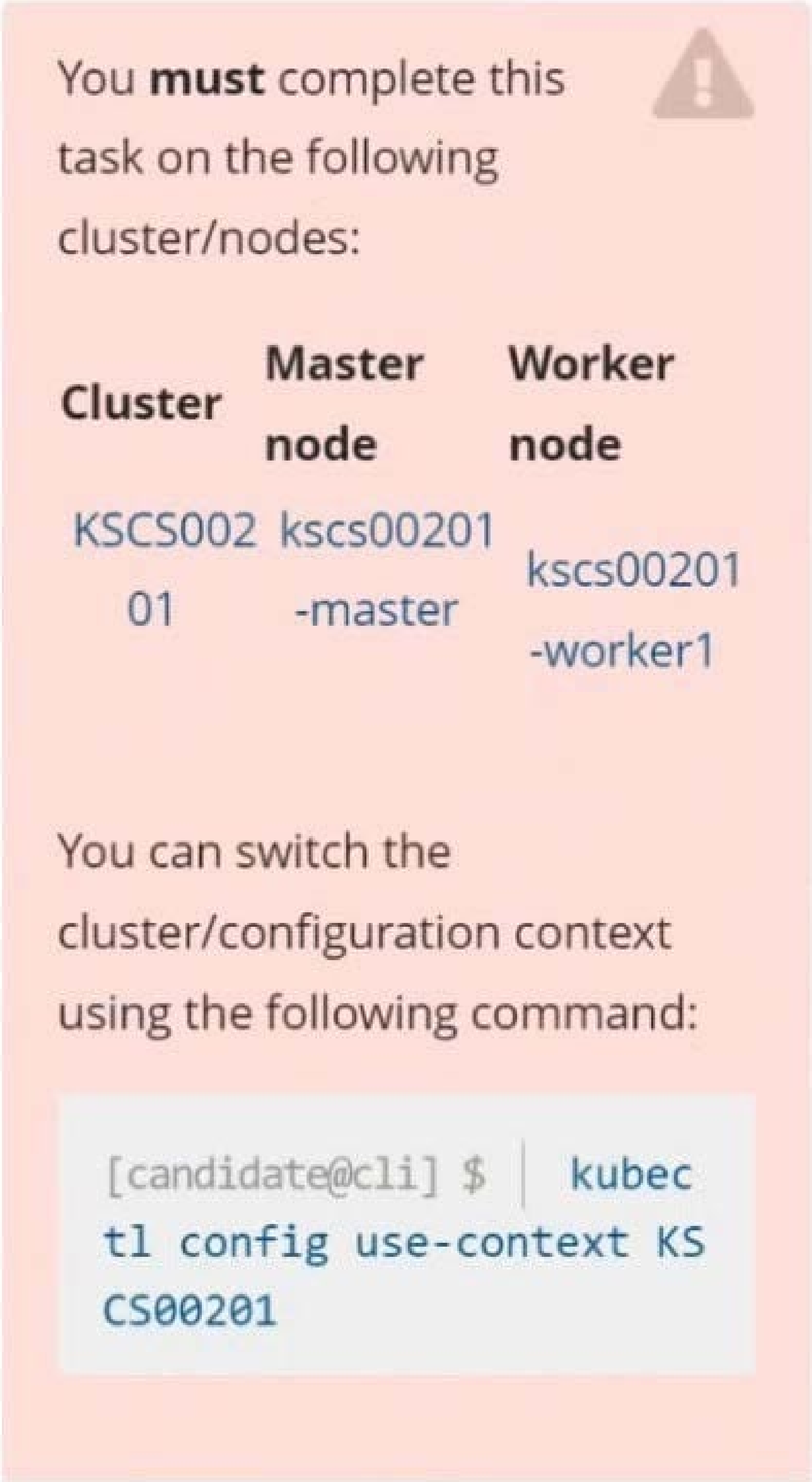
Pass Linux Foundation CKS Exam with 100% Guarantee

Following Questions and Answers are all new published by Linux Foundation Official Exam Center


Instant Download After Purchase
100% Money Back Guarantee
365 Days Free Update
800,000+ Satisfied Customers
SATISFACTION GUARANTEED
100%
SATISFACTION GUARANTEED

QUESTION 1

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



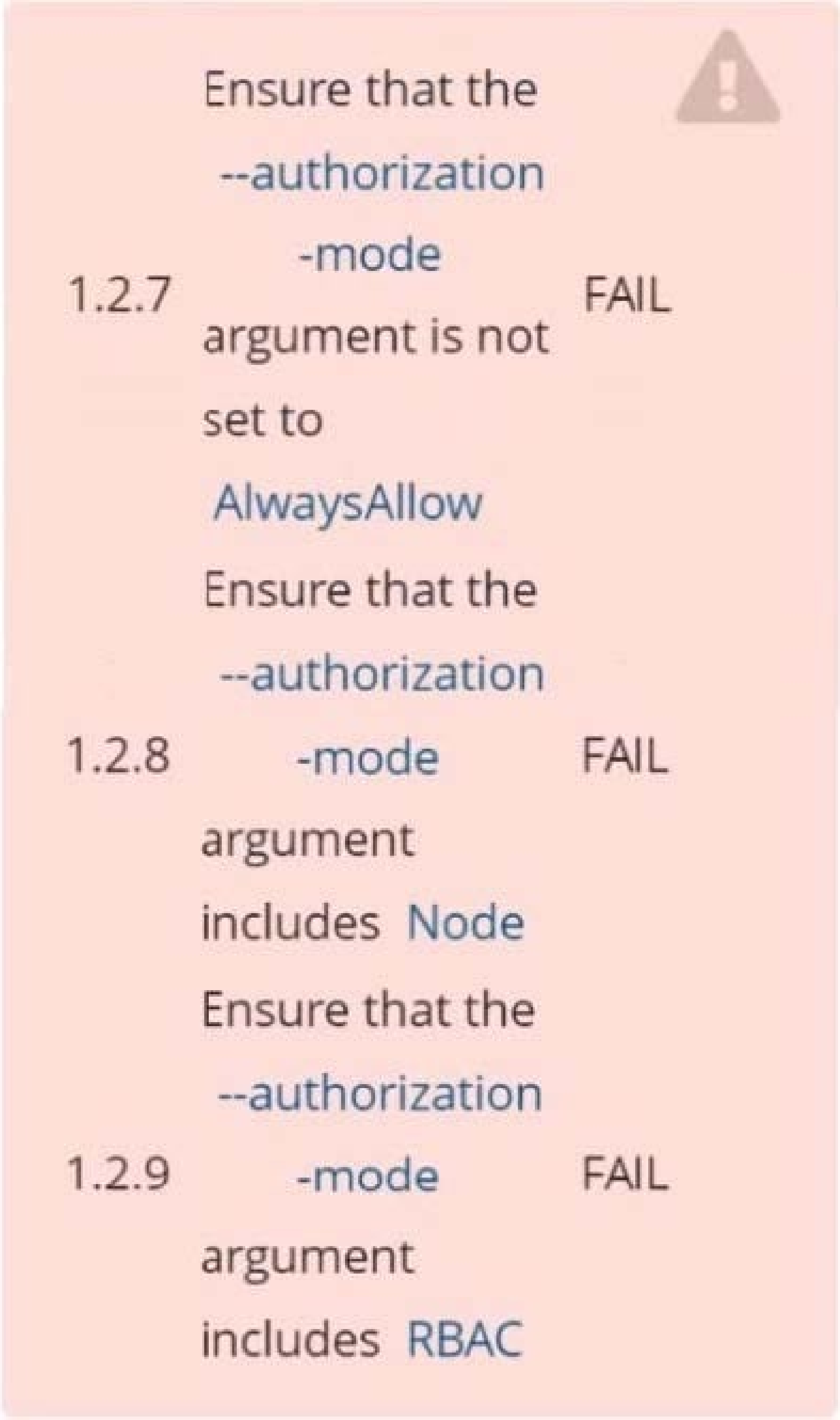
You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCS00201	kscs00201-master	kscs00201-worker1

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

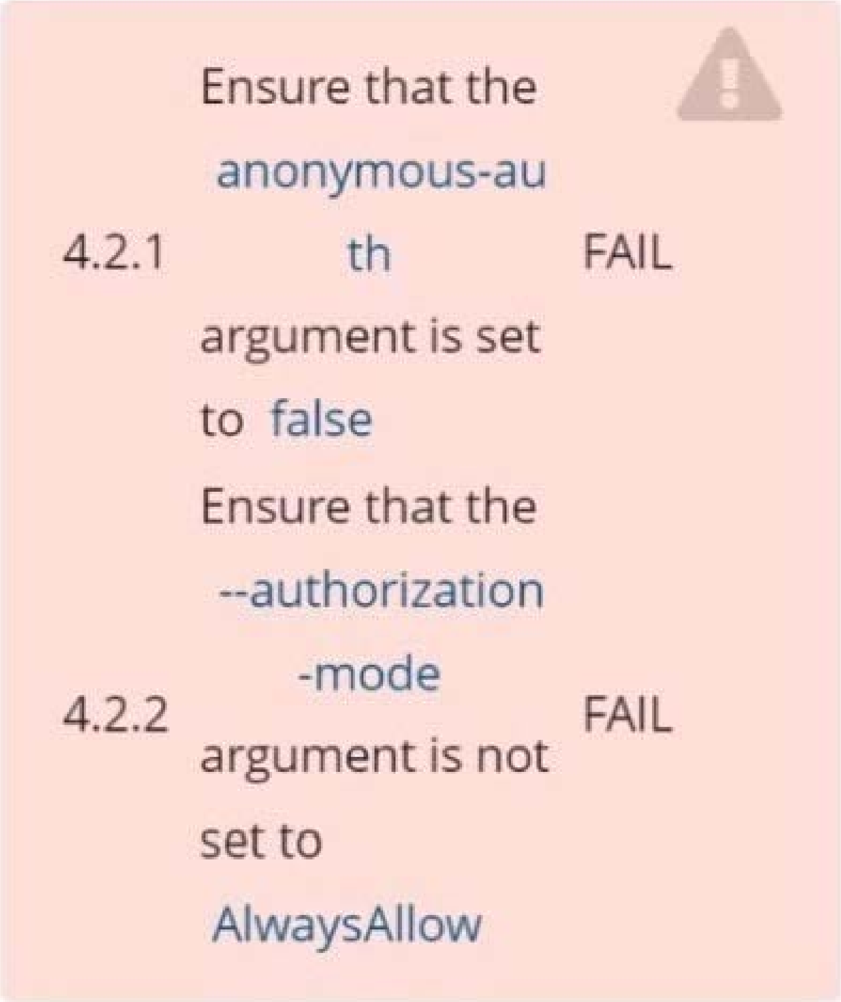
```
[candidate@cli] $ | kubectl config use-context KSCS00201
```

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:

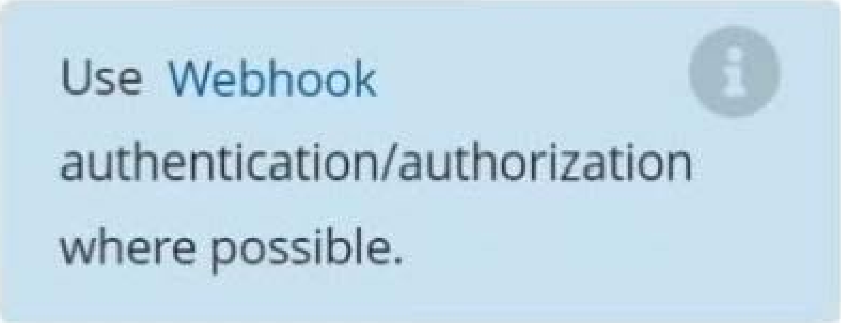


1.2.7	Ensure that the --authorization-mode argument is not set to AlwaysAllow	FAIL
1.2.8	Ensure that the --authorization-mode argument includes Node	FAIL
1.2.9	Ensure that the --authorization-mode argument includes RBAC	FAIL

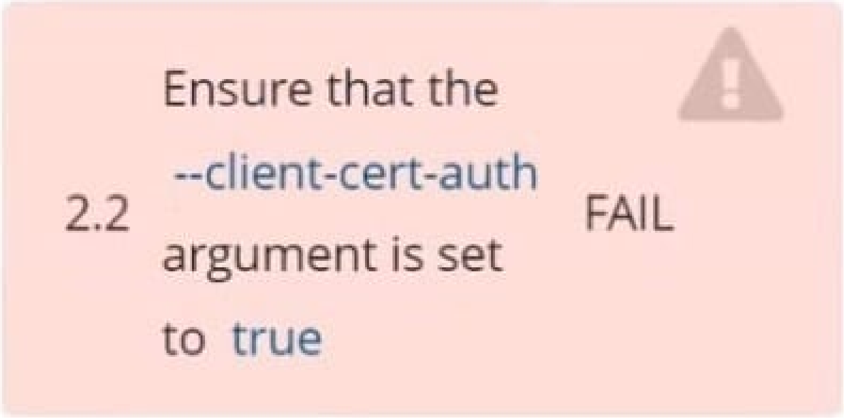
Fix all of the following violations that were found against the Kubelet: Fix all of the following violations that were found against etcd:



4.2.1	Ensure that the anonymous-auth argument is set to false	FAIL
4.2.2	Ensure that the --authorization-mode argument is not set to AlwaysAllow	FAIL



Use Webhook authentication/authorization where possible.

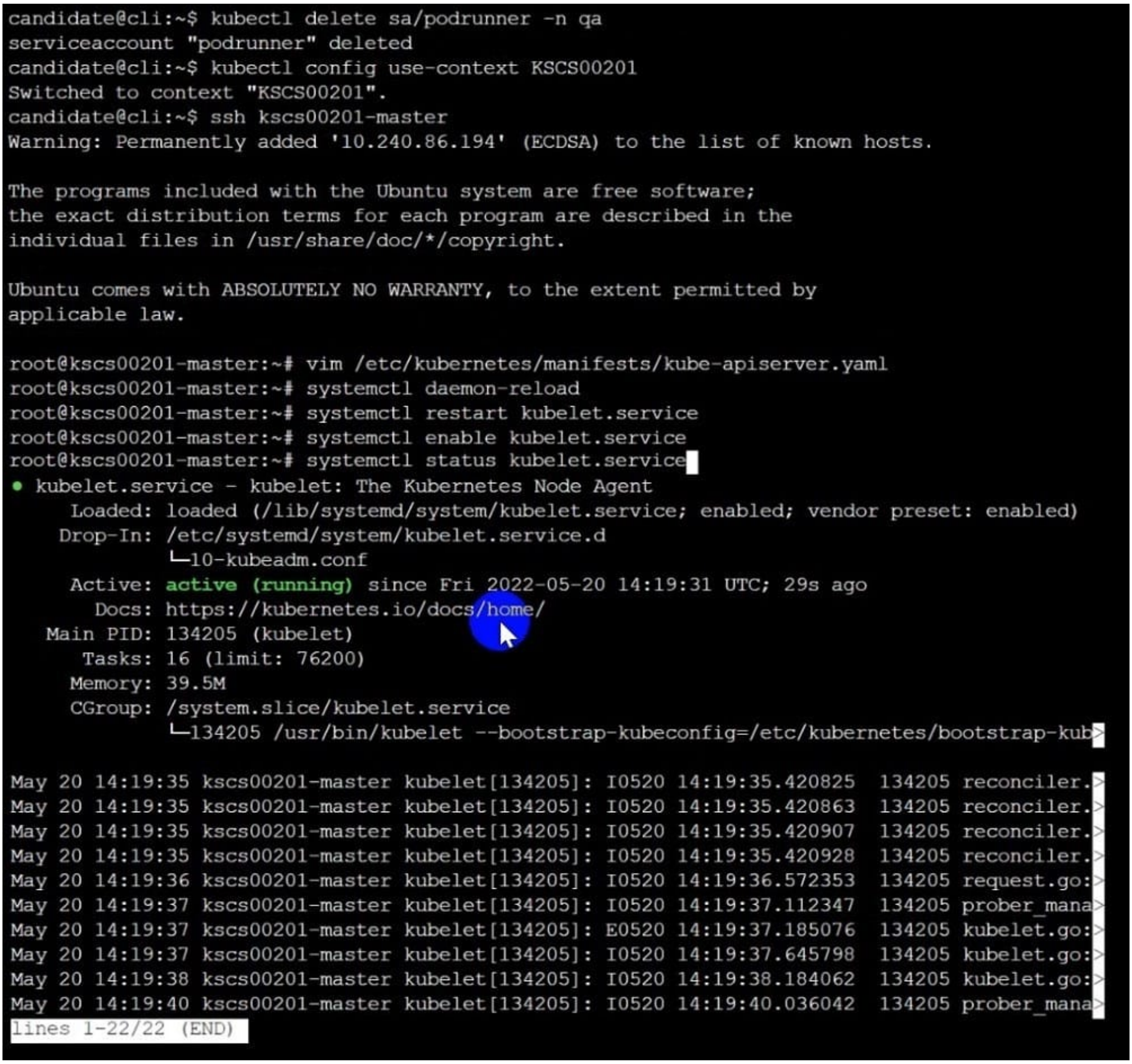


2.2	Ensure that the --client-cert-auth argument is set to true	FAIL	⚠

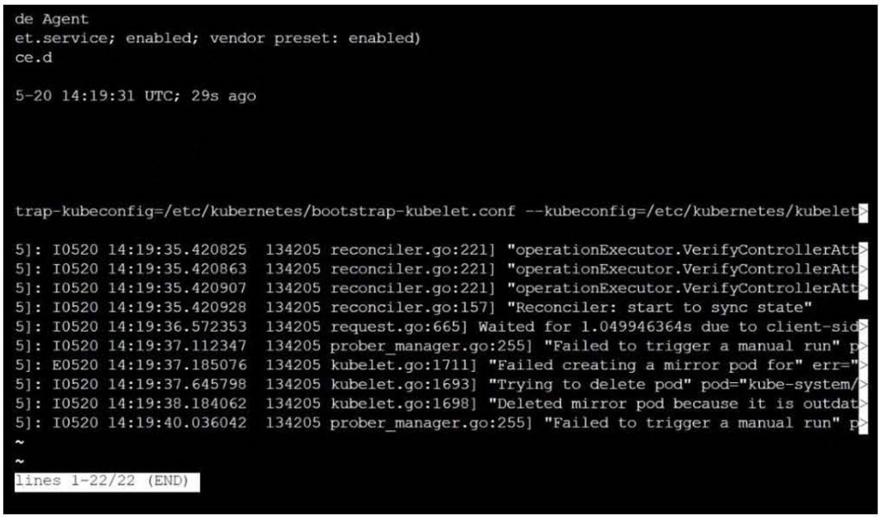
A. See explanation below.

B. PlaceHolder

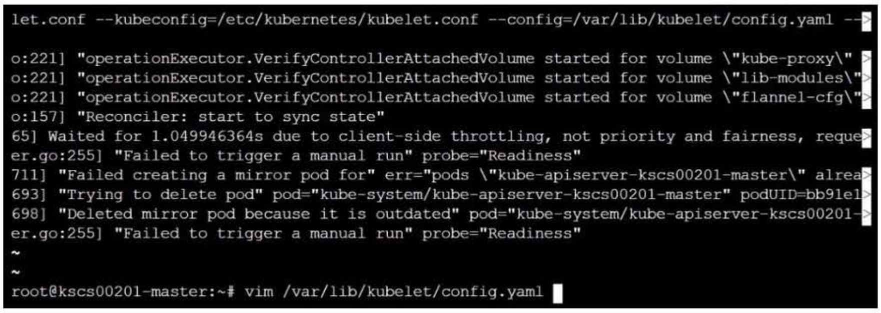
Correct Answer: A



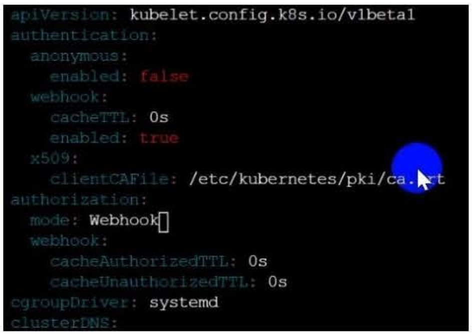
```
candidate@cli:~$ kubectl delete sa/podrunner -n qa
serviceaccount "podrunner" deleted
candidate@cli:~$ kubectl config use-context KSCS00201
Switched to context "KSCS00201".
candidate@cli:~$ ssh kscs00201-master
Warning: Permanently added '10.240.86.194' (ECDSA) to the list of known hosts.

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)
```



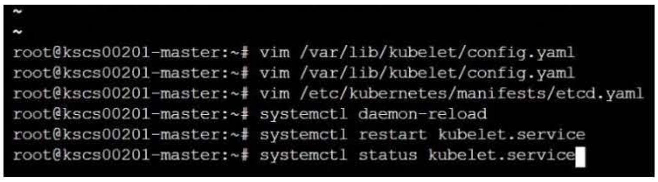
```
de Agent
et.service; enabled; vendor preset: enabled)
ce.d

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)
```



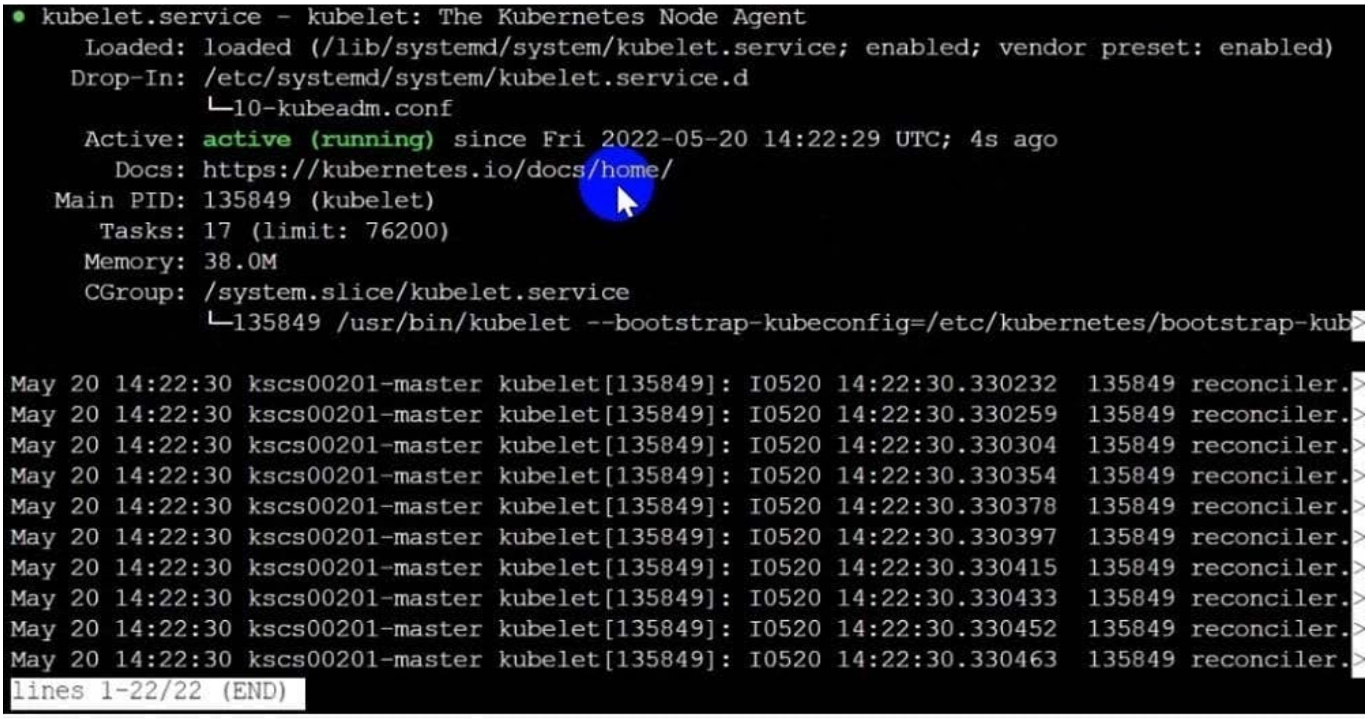
```
let.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml -->

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
```



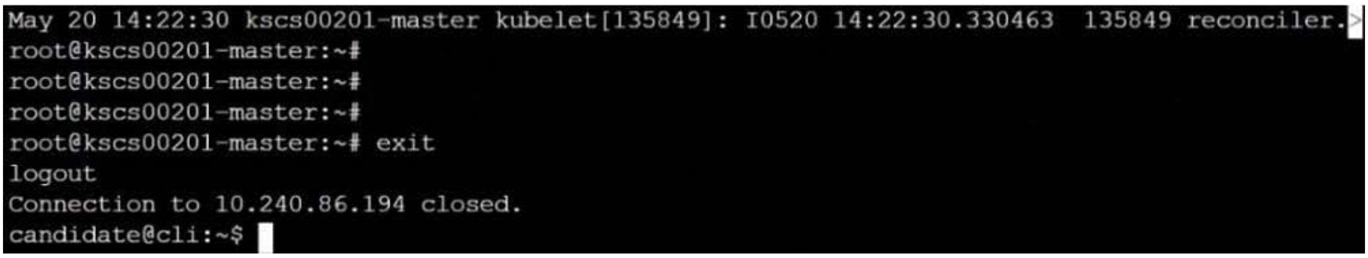
```
apiVersion: kubelet.config.k8s.io/v1beta1
authentication:
  anonymous:
    enabled: false
  webhook:
    cacheTTL: 0s
    enabled: true
  x509:
    clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
cgroupDriver: systemd
clusterDNS:
```



```
~
~
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)
```



```
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 2

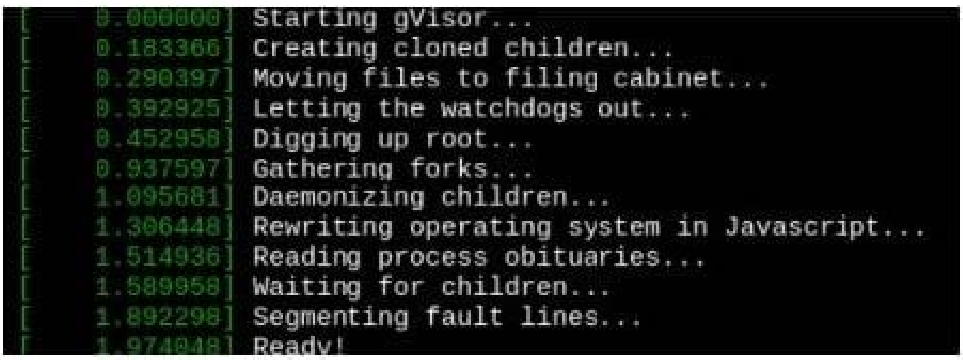
Create a RuntimeClass named untrusted using the prepared runtime handler named runsc.

Create a Pods of image alpine:3.13.2 in the Namespace default to run on the gVisor runtime class.

A. See the explanation below:

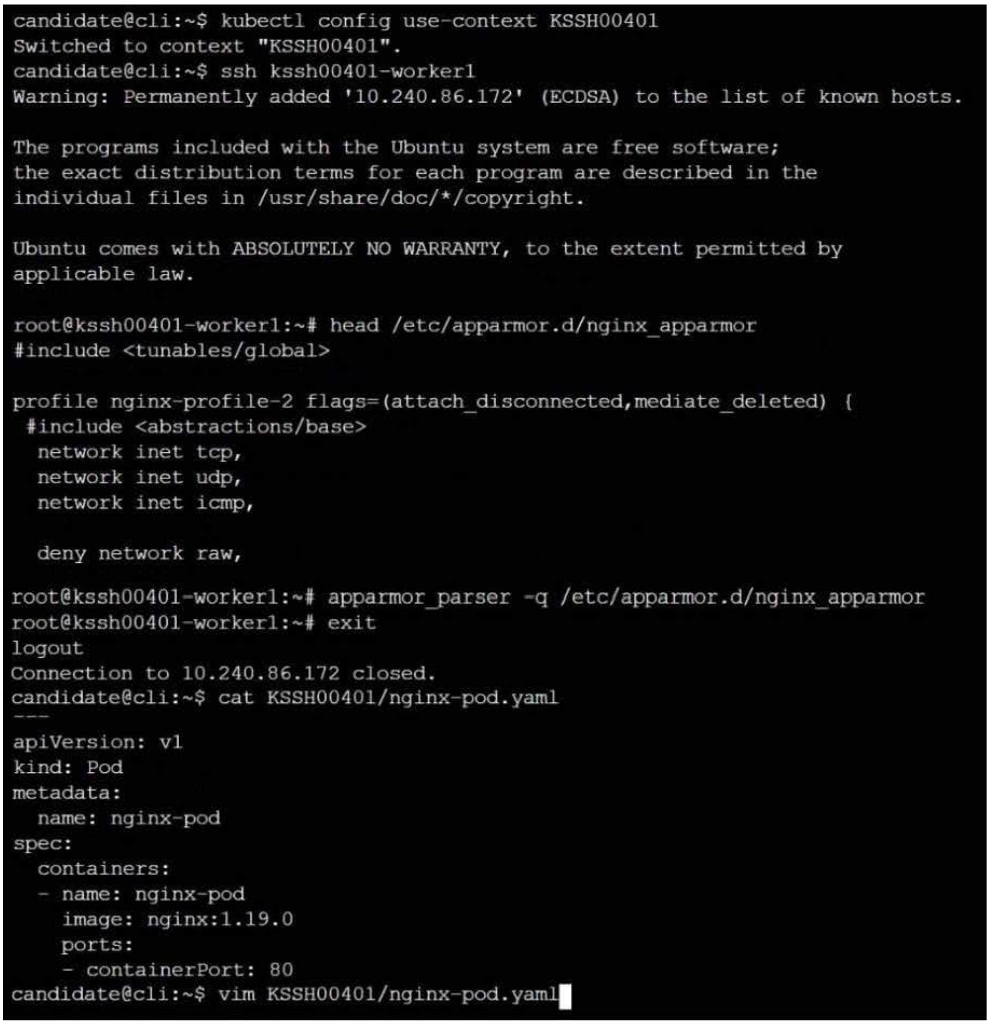
B. PlaceHolder

Correct Answer: A

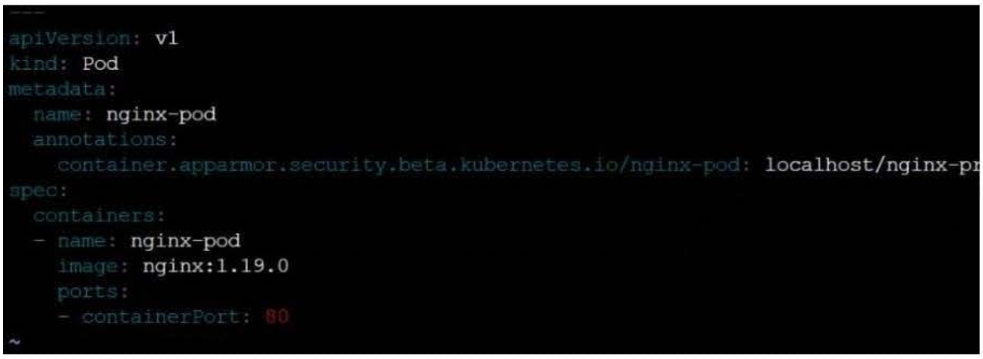


```
[    0.000000] Starting gVisor...
[    0.183366] Creating cloned children...
[    0.290397] Moving files to filing cabinet...
[    0.392925] Letting the watchdogs out...
[    0.452958] Digging up root...
[    0.937597] Gathering forks...
[    1.095681] Daemonizing children...
[    1.306448] Rewriting operating system in Javascript...
[    1.514936] Reading process obituaries...
[    1.589958] Waiting for children...
[    1.892298] Segmenting fault lines...
[    1.974048] Ready!
```

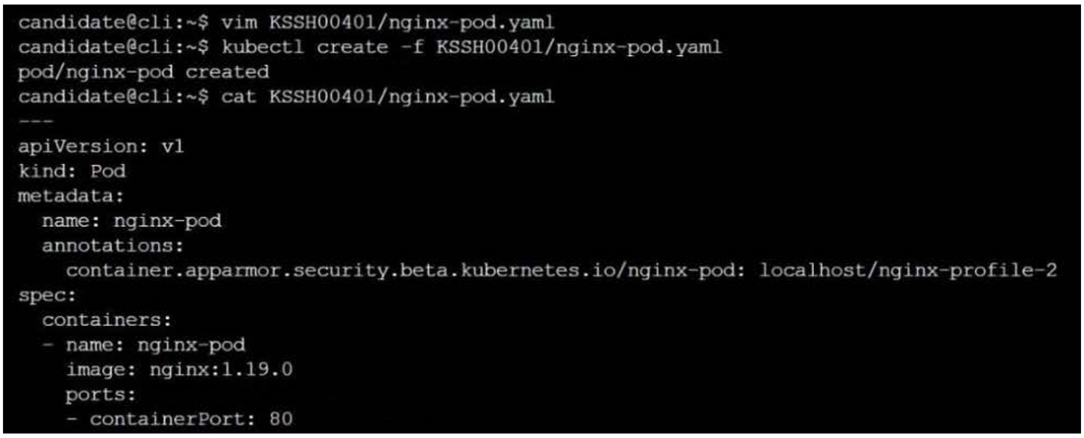
QUESTION 3



```
candidate@cli:~$ kubectl config use-context KSSH00401
Switched to context "KSSH00401".
candidate@cli:~$ ssh kssh00401-worker1
Warning: Permanently added '10.240.86.172' (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@kssh00401-worker1:~# head /etc/apparmor.d/nginx_apparmor
#include <tunables/global>

profile nginx-profile-2 flags=(attach_disconnected,mediate_deleted) {
 #include <abstractions/base>
  network inet tcp,
  network inet udp,
  network inet icmp,

  deny network raw,

root@kssh00401-worker1:~# apparmor_parser -q /etc/apparmor.d/nginx_apparmor
root@kssh00401-worker1:~# exit
logout
Connection to 10.240.86.172 closed.
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
candidate@cli:~$ vim KSSH00401/nginx-pod.yaml
```



```
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-pr
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
~
```



```
candidate@cli:~$ vim KSSH00401/nginx-pod.yaml
candidate@cli:~$ kubectl create -f KSSH00401/nginx-pod.yaml
pod/nginx-pod created
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-profile-2
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
```

Fix all issues via configuration and restart the affected components to ensure the new setting takes effect. Fix all of the following violations that were found against the API server:

1.

Ensure that the RotateKubeletServerCertificate argument is set to true.

2.

Ensure that the admission control plugin PodSecurityPolicy is set.

3.

Ensure that the --kubelet-certificate-authority argument is set as appropriate. Fix all of the following violations that were found against the Kubelet:

1.

Ensure the --anonymous-auth argument is set to false.

2.

Ensure that the --authorization-mode argument is set to Webhook. Fix all of the following violations that were found against the ETCD:

1.

Ensure that the --auto-tls argument is not set to true

2.

Ensure that the --peer-auto-tls argument is not set to true

Hint: Take the use of Tool Kube-Bench

A. See the below.

B. PlaceHolder

Correct Answer: A

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

a. Ensure that the RotateKubeletServerCertificate argument is set to true.

apiVersion: v1 kind: Pod metadata: creationTimestamp: null labels: component: kubelet tier: control-plane name: kubelet namespace: kube-system spec: containers:

-command:

-kube-controller-manager + - --feature-gates=RotateKubeletServerCertificate=true image: gcr.io/google_containers/kubelet-amd64:v1.6.0 livenessProbe: failureThreshold: 8 httpGet: host: 127.0.0.1 path: /healthz port: 6443 scheme: HTTPS initialDelaySeconds: 15 timeoutSeconds: 15 name: kubelet resources: requests: cpu: 250m volumeMounts:

-

mountPath: /etc/kubernetes/ name: k8s readOnly: true

-

mountPath: /etc/ssl/certs name: certs

-

mountPath: /etc/pki name: pki hostNetwork: true volumes:

-

hostPath: path: /etc/kubernetes name: k8s

-

hostPath: path: /etc/ssl/certs name: certs

-

hostPath: path: /etc/pki name: pki

b.

Ensure that the admission control plugin PodSecurityPolicy is set.

audit: "/bin/ps -ef | grep $apiserverbin | grep -v grep"

tests:

test_items:

-flag: "--enable-admission-plugins"

compare:

op: has

value: "PodSecurityPolicy"

set: true

remediation: |

Follow the documentation and create Pod Security Policy objects as per your environment.

Then, edit the API server pod specification file $apiserverconf

on the master node and set the --enable-admission-plugins parameter to a

value that includes PodSecurityPolicy :

--enable-admission-plugins=...,PodSecurityPolicy,...

Then restart the API Server.

scored: true

c. Ensure that the --kubelet-certificate-authority argument is set as appropriate. audit: "/bin/ps -ef | grep $apiserverbin | grep -v grep"

tests: test_items:

-flag: "--kubelet-certificate-authority"

set: true

remediation: |

Follow the Kubernetes documentation and setup the TLS connection between the

apiserver and kubelets. Then, edit the API server pod specification file

$apiserverconf on the master node and set the --kubelet-certificate-authority

parameter to the path to the cert file for the certificate authority.

--kubelet-certificate-authority=

scored: true

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

a.

Ensure that the --auto-tls argument is not set to true Edit the etcd pod specification file $etcdconf on the masternode and either remove the -- auto-tls parameter or set it to false.--auto-tls=false

b.

Ensure that the --peer-auto-tls argument is not set to true

Edit the etcd pod specification file $etcdconf on the masternode and either remove the -- peer-auto-tls parameter or set it to false.--peer-auto-tls=false

QUESTION 4

Create a User named john, create the CSR Request, fetch the certificate of the user after approving it.

Create a Role name john-role to list secrets, pods in namespace john

Finally, Create a RoleBinding named john-role-binding to attach the newly created role john-role to the user john in the namespace john.

To Verify: Use the kubectl auth CLI command to verify the permissions.

A. See the below.

B. PlaceHolder

Correct Answer: A

se kubectl to create a CSR and approve it.

Get the list of CSRs:

```
kubectl get csr
```

Approve the CSR:

```
kubectl certificate approve myuser
```

Get the certificateRetrieve the certificate from the CSR:

```
kubectl get csr/myuser -o yaml
```

here are the role and role-binding to give john permission to create NEW_CRD resource:

```
kubectl apply -f roleBindingJohn.yaml --as=john

rolebinding.rbac.authorization.k8s.io/john_external-rosource-rb created

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: john_crd

namespace: development-john

subjects:

-kind: User name: john apiGroup: rbac.authorization.k8s.io roleRef: kind: ClusterRole name: crd-creation

kind: ClusterRole apiVersion: rbac.authorization.k8s.io/v1 metadata: name: crd-creation rules:

-apiGroups: ["kubernetes-client.io/v1"] resources: ["NEW_CRD"] verbs: ["create, list, get"]
```

QUESTION 5

Create a new ServiceAccount named backend-sa in the existing namespace default, which has the capability to list the pods inside the namespace default.

Create a new Pod named backend-pod in the namespace default, mount the newly created sa backend-sa to the pod, and Verify that the pod is able to list pods.

Ensure that the Pod is running.

A. See the below:

B. PlaceHolder

Correct Answer: A

A service account provides an identity for processes that run in a Pod.

When you (a human) access the cluster (for example, using kubectl), you are authenticated by the apiserver as a particular User Account (currently this is usually admin, unless your cluster administrator has customized your cluster). Processes in containers inside pods can also contact the apiserver. When they do, they are authenticated as a particular Service Account (for example, default).

When you create a pod, if you do not specify a service account, it is automatically assigned the default service account in the same namespace. If you get the raw json or yaml for a pod you have created (for example, kubectl get pods/ -o yaml), you can see the spec.serviceAccountName field has been automatically set. You can access the API from inside a pod using automatically mounted service account credentials, as described in Accessing the Cluster. The API permissions of the service account depend on the authorization plugin and policy in use. In version 1.6+, you can opt out of automounting API credentials for a service account by setting automountServiceAccountToken: false on the service account:

apiVersion: v1 kind: ServiceAccount metadata: name: build-robot automountServiceAccountToken: false

In version 1.6+, you can also opt out of automounting API credentials for a particular pod: apiVersion: v1 kind: Pod metadata: name: my-pod spec: serviceAccountName: build-robot automountServiceAccountToken: false

The pod spec takes precedence over the service account if both specify a automountServiceAccountToken value.