



CKS^{Q&As}

Certified Kubernetes Security Specialist (CKS) Exam

Pass Linux Foundation CKS Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/cks.html>

100% Passing Guarantee
100% Money Back Assurance

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





QUESTION 1

```
candidate@cli:~$ kubectl config use-context KRSR00602
Switched to context "KRSR00602".
candidate@cli:~$ ssh krsr00602-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@krsr00602-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@krsr00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
```



```
- "/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"
```



```
- "/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
```

```
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
```

```
# 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:~$
```



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

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

Context:

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

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:

1.2.7 authorization-mode argument is not set to AlwaysAllow FAIL

1.2.8 authorization-mode argument includes Node FAIL

1.2.7 authorization-mode argument includes RBAC FAIL

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

4.2.1 Ensure that the anonymous-auth argument is set to false FAIL

4.2.2 authorization-mode argument is not set to AlwaysAllow FAIL (Use Webhook authn/authz where possible)

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

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

A. See the explanation below

B. Placeholder

Correct Answer: A

```
worker1 $ vim /var/lib/kubelet/config.yaml uk.co.certification.simulator.questionpool.PList@132b77a0 worker1 $
systemctl restart kubelet. # To reload kubelet configssh to master1master1 $ vim /etc/kubernetes/manifests/kube-
apiserver.yaml- -- authorizationmode=Node,RBACmaster1 $ vim /etc/kubernetes/manifests/etcd.yaml- --client-cert-
auth=true
```

Explanationssh to worker1worker1 \$ vim /var/lib/kubelet/config.yaml apiVersion: kubelet.config.k8s.io/v1beta1
authentication: anonymous: enabled: true #Delete this enabled: false #Replace by this webhook: cacheTTL: 0s enabled:
true x509: clientCAFile: /etc/kubernetes/pki/ca.crt authorization: mode: AlwaysAllow #Delete this mode: Webhook
#Replace by this webhook: cacheAuthorizedTTL: 0s cacheUnauthorizedTTL: 0s cgroupDriver: systemd 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 logging: {} nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s resolvConf: /run/systemd/resolve/resolv.conf rotateCertificates: true
runtimeRequestTimeout: 0s staticPodPath: /etc/kubernetes/manifests streamingConnectionIdleTimeout: 0s
syncFrequency: 0s volumeStatsAggPeriod: 0s worker1 $ systemctl restart kubelet. # To reload kubelet configssh to
master1master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml
```




```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 172.17.0.22:6443
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=172.17.0.22
    - --allow-privileged=true
    # - --authorization-mode=AlwaysAllow # Delete This
    - --authorization-mode=Node,RBAC # Replace by this line
    - --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
    - --insecure-port=0
```

```
master1 $ vim /etc/kubernetes/manifests/etcd.yaml
```

QUESTION 2

Service is running on port 389 inside the system, find the process-id of the process, and stores the names of all the open-files inside the /candidate/KH77539/files.txt, and also delete the binary.

A. See explanation below.

B. Placeholder

Correct Answer: A

```
root# netstat -ltnup
```

Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
-------	--------	--------	---------------	-----------------	-------	------------------

tcp	0	0	127.0.0.1:17600	0.0.0.0:*	LISTEN	1293/dropbox
-----	---	---	-----------------	-----------	--------	--------------

tcp	0	0	127.0.0.1:17603	0.0.0.0:*	LISTEN	1293/dropbox
-----	---	---	-----------------	-----------	--------	--------------

tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	575/sshd
-----	---	---	------------	-----------	--------	----------

tcp	0	0	127.0.0.1:9393	0.0.0.0:*	LISTEN	900/perl
-----	---	---	----------------	-----------	--------	----------

tcp	0	0	:::80	:::*	LISTEN	9583/docker-proxy
-----	---	---	-------	------	--------	-------------------

tcp	0	0	:::443	:::*	LISTEN	9571/docker-proxy
-----	---	---	--------	------	--------	-------------------



```
udp 0 0 0.0.0.0:68 0.0.0.0:* 8822/dhcpd
```

```
root# netstat -ltnup | grep \':22\'
```

```
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 575/sshd
```

The ss command is the replacement of the netstat command.

Now let's see how to use the ss command to see which process is listening on port 22:

```
root# ss -ltnup \':sport = :22\'
```

```
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
```

```
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:(\"sshd\",pid=575,fd=3))
```

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

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 the `--authorization-mode` argument includes RBAC

2.

Ensure the `--authorization-mode` argument includes Node

3.

Ensure that the `--profiling` argument is set to false



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:

Ensure that the --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

API server:

Ensure the --authorization-mode argument includes RBAC

Turn on Role Based Access Control. Role Based Access Control (RBAC) allows fine-grained control over the operations that different entities can perform on different objects in the cluster. It is recommended to use the RBAC authorization mode.

Fix - BuildtimeKubernetesapiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

component: kube-apiserver

tier: control-plane

name: kube-apiserver

namespace: kube-system

spec:

containers:

-command: + - kube-apiserver + - --authorization-mode=RBAC,Node image: gcr.io/google_containers/kube-apiserver-amd64:v1.6.0 livenessProbe: failureThreshold: 8 httpGet: host: 127.0.0.1 path: /healthz port: 6443 scheme: HTTPS initialDelaySeconds: 15 timeoutSeconds: 15 name: kube-apiserver-should-pass 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

Ensure the --authorization-mode argument includes Node

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --authorization-mode parameter to a value that includes Node.

--authorization-mode=Node,RBAC

Audit:

/bin/ps -ef | grep kube-apiserver | grep -v grep

Expected result:

\\'Node,RBAC\\' has \\'Node\\'

Ensure that the --profiling argument is set to false

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the below parameter.

--profiling=false

Audit:

/bin/ps -ef | grep kube-apiserver | grep -v grep

Expected result:

\\'false\\' is equal to \\'false\\'

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

uk.co.certification.simulator.questionpool.PList@e3e35a0

Remediation: If using a Kubelet config file, edit the file to set authentication: anonymous:



enabled to false. If using executable arguments, edit the kubelet service file /etc/systemd/system/kubelet.service.d/10-kubeadm.conf on each worker node and set the below parameter in KUBELET_SYSTEM_PODS_ARGS variable.

```
--anonymous-auth=false
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
```

```
systemctl restart kubelet.service
```

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected result:

```
\\false\\ is equal to \\false\\
```

2) Ensure that the --authorization-mode argument is set to Webhook.

Audit

```
docker inspect kubelet | jq -e \\.Args[] | match("--authorization- mode=Webhook").string\\
```

Returned Value: --authorization-mode=Webhook

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

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

Do not use self-signed certificates for TLS. etcd is a highly-available key value store used by Kubernetes deployments for persistent storage of all of its REST API objects. These objects are sensitive in nature and should not be available to unauthenticated clients. You should enable the client authentication via valid certificates to secure the access to the etcd service.

Fix - BuildtimeKubernetesapiVersion: v1 kind: Pod metadata: annotations: scheduler.alpha.kubernetes.io/critical-pod: "" creationTimestamp: null labels: component: etcd tier: control-plane name: etcd namespace: kube-system spec: containers:

-command:

```
+ - etcd
```

```
+ - --auto-tls=true
```

```
image: k8s.gcr.io/etcd-amd64:3.2.18
```

```
imagePullPolicy: IfNotPresent
```

```
livenessProbe:
```



exec:

command:

-/bin/sh

- -ec

-ETCDCTL_API=3 etcdctl --endpoints=https://[192.168.22.9]:2379 -- cacert=/etc/kubernetes/pki/etcd/ca.crt

--cert=/etc/kubernetes/pki/etcd/healthcheck-client.crt -- key=/etc/kubernetes/pki/etcd/healthcheck-client.key get foo

failureThreshold: 8

initialDelaySeconds: 15

timeoutSeconds: 15

name: etcd-should-fail

resources: {}

volumeMounts:

-

mountPath: /var/lib/etcd

name: etcd-data

-

mountPath: /etc/kubernetes/pki/etcd

name: etcd-certs

hostNetwork: true

priorityClassName: system-cluster-critical

volumes:

-

hostPath:

path: /var/lib/etcd

type: DirectoryOrCreate

name: etcd-data

-

hostPath:

path: /etc/kubernetes/pki/etcd



type: DirectoryOrCreate

name: etcd-certs

status: {}



```
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-kube-

May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420825 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420863 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420907 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420928 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:36 kscs00201-master kubelet[134205]: I0520 14:19:36.572353 134205 request.go:665] "Reconciler: start to sync state"
May 20 14:19:37 kscs00201-master kubelet[134205]: I0520 14:19:37.112347 134205 request.go:665] "Reconciler: start to sync state"
May 20 14:19:37 kscs00201-master kubelet[134205]: E0520 14:19:37.185076 134205 kubelet.go:1711] "Failed creating a mirror pod for" err="pods \"kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:37 kscs00201-master kubelet[134205]: I0520 14:19:37.645798 134205 kubelet.go:1693] "Trying to delete pod" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:38 kscs00201-master kubelet[134205]: I0520 14:19:38.184062 134205 kubelet.go:1698] "Deleted mirror pod because it is outdated" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
May 20 14:19:40 kscs00201-master kubelet[134205]: I0520 14:19:40.036042 134205 kubelet.go:1698] "Deleted mirror pod because it is outdated" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
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.conf

5]: I0520 14:19:35.420825 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:35.420863 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:35.420907 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:35.420928 134205 reconciler.go:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:36.572353 134205 request.go:665] "Reconciler: start to sync state"
5]: I0520 14:19:37.112347 134205 request.go:665] "Reconciler: start to sync state"
5]: E0520 14:19:37.185076 134205 kubelet.go:1711] "Failed creating a mirror pod for" err="pods \"kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:37.645798 134205 kubelet.go:1693] "Trying to delete pod" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:38.184062 134205 kubelet.go:1698] "Deleted mirror pod because it is outdated" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
5]: I0520 14:19:40.036042 134205 kubelet.go:1698] "Deleted mirror pod because it is outdated" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
~
~
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 \"kubernetes.io/config\" (\"kubernetes.io/config\")"
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kubernetes.io/config\" (\"kubernetes.io/config\")"
o:157] "Reconciler: start to sync state"
65] Waited for 1.049946364s due to client-side throttling, not priority and fairness, request for manual run" probe="Readiness"
711] "Failed creating a mirror pod for" err="pods \"kubernetes.io/config\" (\"kubernetes.io/config\")"
693] "Trying to delete pod" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
698] "Deleted mirror pod because it is outdated" pod="kubernetes.io/config\" (\"kubernetes.io/config\")"
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 5

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

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

A default-deny NetworkPolicy avoid to accidentally expose a Pod in a namespace that doesn't have any other NetworkPolicy defined.

Task: Create a new default-deny NetworkPolicy named deny-network in the namespace test for all traffic of type Ingress + Egress

The new NetworkPolicy must deny all Ingress + Egress traffic in the namespace test.

Apply the newly created default-deny NetworkPolicy to all Pods running in namespace test.

You can find a skeleton manifests file at /home/cert_masters/network-policy.yaml

A. See the explanation below

B. Placeholder

Correct Answer: A

```
master1 $ k get pods -n test --show-labels uk.co.certification.simulator.questionpool.PList@132b47c0 $ vim netpol.yaml
uk.co.certification.simulator.questionpool.PList@132b4af0 master1 $ k apply -f netpol.yaml
```

```
controlplane $ k get pods -n test --show-labels NAME READY STATUS RESTARTS AGE LABELS test-pod 1/1
Running 0 34s role=test,run=test-pod testing 1/1 Running 0 17d run=testing master1 $ vim netpol1.yaml apiVersion:
networking.k8s.io/v1 kind: NetworkPolicy metadata: name: deny-network namespace: test spec: podSelector: {}
policyTypes:
```

-Ingress

-Egress

[Latest CKS Dumps](#)

[CKS VCE Dumps](#)

[CKS Braindumps](#)