



# CKS<sup>Q&As</sup>

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.passapply.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

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 '\\.[0].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-kub
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:157] "Reconciler: start to sync state"
May 20 14:19:36 kscs00201-master kubelet[134205]: I0520 14:19:36.572353 134205 request.go:665] "Waited for 1.049946364s due to client-side throttling, not priority and fairness, request: GET https://10.240.86.194:443/apis/core/v1/pods?labelSelector=kube-system%2Fkube-apiserver-kscs00201-master%2Fbb91e1b2-6993-4371-b078-426111111111"
May 20 14:19:37 kscs00201-master kubelet[134205]: I0520 14:19:37.112347 134205 prober_manager.go:255] "Failed to trigger a manual run" probe="Readiness"
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\") already exists"
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\")" podUID=bb91e1b2-6993-4371-b078-426111111111"
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 prober_manager.go:255] "Failed to trigger a manual run" probe="Readiness"
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:157] "Reconciler: start to sync state"
5]: I0520 14:19:36.572353 134205 request.go:665] "Waited for 1.049946364s due to client-side throttling, not priority and fairness, request: GET https://10.240.86.194:443/apis/core/v1/pods?labelSelector=kube-system%2Fkube-apiserver-kscs00201-master%2Fbb91e1b2-6993-4371-b078-426111111111"
5]: I0520 14:19:37.112347 134205 prober_manager.go:255] "Failed to trigger a manual run" probe="Readiness"
5]: E0520 14:19:37.185076 134205 kubelet.go:1711] "Failed creating a mirror pod for" err="pods \"kubernetes.io/config\" (\"kubernetes.io/config\") already exists"
5]: I0520 14:19:37.645798 134205 kubelet.go:1693] "Trying to delete pod" pod="kubernetes.io/config\" (\"kubernetes.io/config\")" podUID=bb91e1b2-6993-4371-b078-426111111111"
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 prober_manager.go:255] "Failed to trigger a manual run" probe="Readiness"
~
~
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: GET https://10.240.86.194:443/apis/core/v1/pods?labelSelector=kube-system%2Fkube-apiserver-kscs00201-master%2Fbb91e1b2-6993-4371-b078-426111111111"
er.go:255] "Failed to trigger a manual run" probe="Readiness"
711] "Failed creating a mirror pod for" err="pods \"kubernetes.io/config\" (\"kubernetes.io/config\") already exists"
693] "Trying to delete pod" pod="kubernetes.io/config\" (\"kubernetes.io/config\")" podUID=bb91e1b2-6993-4371-b078-426111111111"
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 2

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

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

Context:

A pod fails to run because of an incorrectly specified ServiceAccount

Task:

Create a new service account named backend-qa in an existing namespace qa, which must not have access to any secret.

Edit the frontend pod yaml to use backend-qa service account

Note: You can find the frontend pod yaml at /home/cert\_masters/frontend-pod.yaml

A. See the explanation below

B. Placeholder

Correct Answer: A

```
[desk@cli] $ k create sa backend-qa -n qasa/backend-qa created [desk@cli] $ k get role,rolebinding -n qaNo resources found in qa namespace. [desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb list# No access to secret [desk@cli] $ k create rolebinding backend -n qa --role backend --serviceaccount qa:backend-qa [desk@cli] $ vim /home/ cert_masters/frontend-pod.yaml uk.co.certification.simulator.questionpool.PList@120e0660 [desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yamlpod created [desk@cli] $ k create sa backend-qa -n qaserviceaccount/backend-qa created [desk@cli] $ k get role,rolebinding -n qaNo resources found in qa namespace. [desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb listrole.rbac.authorization.k8s.io/backend created [desk@cli] $ k create rolebinding backend -n qa --role backend --serviceaccount qa:backendqarolebinding.rbac.authorization.k8s.io/backend created [desk@cli] $ vim /home/cert_masters/frontend-pod.yaml apiVersion: v1 kind: Pod metadata: name: frontend spec: serviceAccountName: backend-qa # Add this image: nginx name: frontend [desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yamlpod/frontend createdhttps://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
```

## QUESTION 3

CORRECT TEXT



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



Cluster	Master node	Worker node
KRS00101	ksrs00101-master	ksrs00101-worker1

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

```
[candidate@cli] $ | kubectl config use-context KRS00101
```

You may use your browser to open **one additional tab** to access Falco's documentation.





Two tools are pre-installed on the cluster's worker node:

1.

sysdig

2.

falco

Using the tool of your choice (including any non pre-installed tool), analyze the container's behavior for at least 30 seconds, using filters that detect newly spawning and executing processes. Store an incident file at /opt/KSRS00101/alerts/

details, containing the detected incidents, one per line, in the following format:

```
timestamp,uid/username,processName
```

The following example shows a properly formatted incident file:

```
01:40:19.601363716,root,init
01:40:20.606013716,nobody,bash
01:40:21.137163716,1000,tar
```

Keep the tool's original  
timestamp-format as-is.



Make sure to store the  
incident file on the cluster's  
worker node.



A. See the explanation below:

B. Placeholder



Correct Answer: A

```
candidate@cli:~$ kubectl config use-context KSRS00101
Switched to context "KSRS00101".
candidate@cli:~$ ssh ksrs00101-worker1
Warning: Permanently added '10.240.86.96' (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@ksrs00101-worker1:~# falco
falco                  falco-driver-loader
root@ksrs00101-worker1:~# ls -l /etc/falco/
total 200
-rw-r--r-- 1 root root 12399 Jan 31 16:06 aws_cloudtrail_rules.yaml
-rw-r--r-- 1 root root 11384 Jan 31 16:06 falco.yaml
-rw-r--r-- 1 root root 1136 Jan 31 16:06 falco_rules.local.yaml
-rw-r--r-- 1 root root 132112 Jan 31 16:06 falco_rules.yaml
-rw-r--r-- 1 root root 27289 Jan 31 16:06 k8s_audit_rules.yaml
drwxr-xr-x 2 root root 4096 Feb 16 01:07 rules.available
drwxr-xr-x 2 root root 4096 Jan 31 16:28 rules.d
root@ksrs00101-worker1:~# vim /etc/falco/falco_rules.local.yaml
```



```
# Copyright (C) 2019 The Falco Authors.
#
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

#####
# Your custom rules!
#####

# Add new rules, like this one
# - rule: The program "sudo" is run in a container
#   desc: An event will trigger every time you run sudo in a container
#   condition: evt.type = execve and evt.dir=< and container.id != host and proc.name = sudo
#   output: "Sudo run in container (user=%user.name %container.info parent=%proc.pname cmdline=%proc.cmdline)"
#   priority: ERROR
#   tags: [users, container]

# Or override/append to any rule, macro, or list from the Default Rules
- rule: Container Drift Detected (chmod)
  desc: New executable created in a container due to chmod
  condition: >
    evt.type in (open,openat,create) and
    evt.is_open_exec=true and
    container and
    not runc_writing_exec_fifo and
    not runc_writing_var_lib_docker and
    not user_known_container_drift_activities and
    evt.rawres>=0
  output:
    %evt.time,%user.uid,%proc.name
  priority: ERROR
```

Text





```
root@ksrs00101-worker1:~# vim /etc/falco/falco_rules.local.yaml
root@ksrs00101-worker1:~# systemctl status falco.service
● falco.service - Falco Runtime Security
   Loaded: loaded (/lib/systemd/system/falco.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
root@ksrs00101-worker1:~# systemctl enable falco.service
Created symlink /etc/systemd/system/multi-user.target.wants/falco.service → /lib/systemd/system/falco.service.
root@ksrs00101-worker1:~# systemctl start falco.service
root@ksrs00101-worker1:~# exit
logout
Connection to 10.240.86.96 closed.
candidate@cli:~$ ssh ksrs00101-worker1
Last login: Fri May 20 15:59:48 2022 from 10.240.86.88
root@ksrs00101-worker1:~# vim /etc/falco/falco.yaml
```

```
# When using json output, whether or not to include the "tags" property
# itself in the json output. If set to true, outputs caused by rules
# with no tags will have a "tags" field set to an empty array. If set to
# false, the "tags" field will not be included in the json output at all.
json_include_tags_property: true

# Send information logs to stderr and/or syslog Note these are *not* security
# notification logs! These are just Falco lifecycle (and possibly error) logs.
log_stderr: true
log_syslog: true
log_file: /opt/KSRS00101/alerts/details

# Minimum log level to include in logs. Note: these levels are
# separate from the priority field of rules. This refers only to the
# log level of falco's internal logging. Can be one of "emergency",
# "alert", "critical", "error", "warning", "notice", "info", "debug".
log_level: info
```

Text





```
root@ksrs00101-worker1:~# vim /etc/falco/falco.yaml
root@ksrs00101-worker1:~# grep log /etc/falco/falco.yaml
# cloudtrail log files.
# If true, the times displayed in log messages and output messages
# Send information logs to stderr and/or syslog Note these are *not* security
# notification logs! These are just Falco lifecycle (and possibly error) logs.
log_stderr: true
log_syslog: true
log_file: /opt/KSRS00101/alerts/details
# Minimum log level to include in logs. Note: these levels are
# log level of falco's internal logging. Can be one of "emergency",
log_level: info
# - log: log a DEBUG message noting that the buffer was full
# Notice it is not possible to ignore and log/alert messages at the same time.
# The rate at which log/alert messages are emitted is governed by a
# - log
# The timeout error will be reported to the log according to the above log_* settings.
syslog_output:
# - logging (alternate method than syslog):
#   program: logger -t falco-test
# this information will be logged, however the main Falco daemon will not be stopped.
root@ksrs00101-worker1:~# systemctl restart falco.service
root@ksrs00101-worker1:~# exit
logout
Connection to 10.240.86.96 closed.
candidate@cli:~$
```

#### QUESTION 4

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

A. See explanation below.

B. Placeholder

Correct Answer: A

---

## QUESTION 5

CORRECT TEXT

Task



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



Cluster	Master node	Worker node
KSSH00301	kssh00301-master	kssh00301-worker1

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

```
[candidate@cli] $ | kubectl config use-context KSSH00301
```

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:

1.  
Pods in the namespace qa
2.  
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/n  
etwork-policy.yaml`



- A. See explanation below.
- B. Placeholder

Correct Answer: A

Explanation

Explanation/Reference:

```
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
```

- A. See explanation below.

- B. Placeholder

Correct Answer: A

[Latest CKS Dumps](#)

[CKS Study Guide](#)

[CKS Exam Questions](#)