



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.geekcert.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

Context:

Cluster: prod

Master node: master1

Worker node: worker1

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

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

Task:

Analyse and edit the given Dockerfile (based on the ubuntu:18:04 image)

/home/cert_masters/Dockerfile fixing two instructions present in the file being prominent security/best-practice issues.

Analyse and edit the given manifest file

/home/cert_masters/mydeployment.yaml fixing two fields present in the file being prominent security/best-practice issues.

Note: Don't add or remove configuration settings; only modify the existing configuration settings, so that two configuration settings each are no longer security/best-practice concerns.

Should you need an unprivileged user for any of the tasks, use user nobody with user id 65535

A. See the explanation below

B. Placeholder

Correct Answer: A

1. For Dockerfile: Fix the image version and user name in Dockerfile. 2. For mydeployment.yaml : Fix security contexts

Explanation
[desk@cli] \$ vim /home/cert_masters/Dockerfile
FROM ubuntu:latest # Remove this FROM ubuntu:18.04 #
Add this USER root # Remove this USER nobody # Add this
RUN apt get install -y lsof=4.72 wget=1.17.1 nginx=4.2
ENV ENVIRONMENT=testing USER root # Remove this USER nobody # Add this
CMD ["nginx -d"]

```
FROM ubuntu:latest # Remove this
FROM ubuntu:18.04 # Add this
USER root # Remove this
USER nobody # Add this
RUN apt get install -y lsof=4.72 wget=1.17.1 nginx=4.2
ENV ENVIRONMENT=testing
USER root # Remove this
USER nobody # Add this
CMD ["nginx -d"]
```



Text

```
[desk@cli] $ vim /home/cert_masters/mydeployment.yaml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
creationTimestamp: null
```

```
labels:
```

```
app: kafka
```

```
name: kafka
```

```
spec:
```

```
replicas: 1
```

```
selector:
```

```
matchLabels:
```

```
app: kafka
```

```
strategy: {}
```

```
template:
```

```
metadata:
```

```
creationTimestamp: null
```

```
labels:
```

```
app: kafka
```

```
spec:
```

```
containers:
```

```
-image: bitnami/kafka
```

```
name: kafka
```

```
volumeMounts:
```

```
-
```

```
name: kafka-vol
```

```
mountPath: /var/lib/kafka
```

```
securityContext:
```



```

{"capabilities":{"add":["NET_ADMIN"],"drop":["all"],"privileged":
True,"readOnlyRootFilesystem": False, "runAsUser": 65535} # Delete This
{"capabilities":{"add":["NET_ADMIN"],"drop":["all"],"privileged":
False,"readOnlyRootFilesystem": True, "runAsUser": 65535} # Add This resources: {}

volumes:
-

name: kafka-vol

emptyDir: {}

status: {}

```

Pictorial View:[desk@cli] \$ vim /home/cert_masters/mydeployment.yaml

```

apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: kafka
  name: kafka
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kafka
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: kafka
    spec:
      containers:
        - image: bitnami/kafka
          name: kafka
          volumeMounts:
            - name: kafka-vol
              mountPath: /var/lib/kafka
      securityContext:
        {"capabilities":{"add":["NET_ADMIN"],"drop":["all"],"privileged": True,"readOnlyRootFilesystem": False, "runAsUser": 65535} # Delete This
        {"capabilities":{"add":["NET_ADMIN"],"drop":["all"],"privileged": False,"readOnlyRootFilesystem": True, "runAsUser": 65535} # Add This
      resources: {}
    volumes:
      - name: kafka-vol
        emptyDir: {}
  status: {}

```

QUESTION 2

Secrets stored in the etcd is not secure at rest, you can use the etcdctl command utility to find the secret value for e.g:ETCDCTL_API=3 etcdctl get /registry/secrets/default/cks-secret --cacert="ca.crt" -- cert="server.crt" --key="server.key" Output

```

/registry/secrets/default/cks-secret
k8s
secret
cks-secret/default*567fcb53f-6b58-4fee-9f12-5737c764be742+
kubectl-create/update/fieldsV1:9
key1: supersecret
key2: topsecret
Visible

```



Using the Encryption Configuration, Create the manifest, which secures the resource secrets using the provider AES-CBC and identity, to encrypt the secret-data at rest and ensure all secrets are encrypted with the new configuration.

A. See explanation below.

B. Placeholder

Correct Answer: A

1.

ETCD secret encryption can be verified with the help of etcdctl command line utility.

2.

ETCD secrets are stored at the path /registry/secrets/\$namespace/\$secret on the master node.

3.

The below command can be used to verify if the particular ETCD secret is encrypted or not.

```
# ETCDCTL_API=3 etcdctl get /registry/secrets/default/secret1 [...] | hexdump -C
```

QUESTION 3

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

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...
[ 1.974918] Ready!
```

QUESTION 4



CORRECT TEXT Your organization's security policy includes:

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



Cluster	Master node	Worker node
KSCH00301	ksch00301 -master	ksch00301 -worker1

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

```
[candidate@cli] $ | kubectl config use-context KSCH00301
```

1.

ServiceAccounts must not automount API credentials



2.

ServiceAccount names must end in "-sa"

The Pod specified in the manifest file `/home/candidate/KSCH00301 /pod-manifest.yaml` fails to schedule because of an incorrectly specified ServiceAccount.

Complete the following tasks:

Task

1.

Create a new ServiceAccount named `frontend-sa` in the existing namespace `qa`. Ensure the ServiceAccount does not automount API credentials.

2.

Using the manifest file at `/home/candidate/KSCH00301 /pod-manifest.yaml`, create the Pod.

3.

Finally, clean up any unused ServiceAccounts in namespace `qa`.

A. See the explanation below

B. Placeholder

Correct Answer: A

QUESTION 5

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.
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_manag
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_manag
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, reques
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:~$
```



[CKS PDF Dumps](#)

[CKS Study Guide](#)

[CKS Exam Questions](#)