**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 1 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 1  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

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

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

[timestamp],[uid],[processName]

**A**Send us the Feedback on it.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-1-discussion)

**Suggested Answer:** *A*

by  Samira at *Dec 08, 2021, 10:15 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 2 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 2  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

Use the kubesec docker images to scan the given YAML manifest, edit and apply the advised changes, and passed with a score of 4 points.

kubesec-test.yaml

apiVersion: v1

kind: Pod

metadata:

name: kubesec-demo

spec:

containers:

- name: kubesec-demo

image: gcr.io/google-samples/node-hello:1.0

securityContext:

readOnlyRootFilesystem: true

Hint:docker run -i kubesec/kubesec:512c5e0 scan /dev/stdin < kubesec-test.yaml

**A**Send us the Feedback on it.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-2-discussion)

**Suggested Answer:** *A*

by  Carey at *Dec 09, 2021, 03:32 AM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 3 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 3  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

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**Send us your feedback on it.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-3-discussion)

**Suggested Answer:** *A*

by  Sharee at *Dec 09, 2021, 07:05 AM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 4 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 4  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

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

Text

Description automatically generated

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**Send us the Feedback on it.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-4-discussion)

**Suggested Answer:** *A*

by  Dick at *Dec 05, 2021, 10:37 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 5 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 5  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

Create a Pod name Nginx-pod inside the namespace testing, Create a service for the Nginx-pod named nginx-svc, using the ingress of your choice, run the ingress on tls, secure port.

**A**Sendusyourfeedbackonit

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-5-discussion)

**Suggested Answer:** *A*

by  Viola at *Dec 05, 2021, 10:36 AM*

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 6  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

Given an existing Pod named test-web-pod running in the namespace test-system

Edit the existing Role bound to the Pod's Service Account named sa-backend to only allow performing get operations on endpoints.

Create a new Role named test-system-role-2 in the namespace test-system, which can perform patch operations, on resources of type statefulsets.

Create a new RoleBinding named test-system-role-2-binding binding the newly created Role to the Pod's ServiceAccount sa-backend.

**A**Send us your feedback on this.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-6-discussion)

**Suggested Answer:** *A*

by  Rasheeda at *Dec 06, 2021, 09:06 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 7 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 7  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

Create a PSP that will prevent the creation of privileged pods in the namespace.

Create a new PodSecurityPolicy named prevent-privileged-policy which prevents the creation of privileged pods.

Create a new ServiceAccount named psp-sa in the namespace default.

Create a new ClusterRole named prevent-role, which uses the newly created Pod Security Policy prevent-privileged-policy.

Create a new ClusterRoleBinding named prevent-role-binding, which binds the created ClusterRole prevent-role to the created SA psp-sa.

Also, Check the Configuration is working or not by trying to Create a Privileged pod, it should get failed.

**A**Explanation:  
Create a PSP that will prevent the creation of privileged pods in the namespace.  
$ cat clusterrole-use-privileged.yaml  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRole  
metadata:  
name: use-privileged-psp  
rules:  
- apiGroups: ['policy']  
resources: ['podsecuritypolicies']  
verbs: ['use']  
resourceNames:  
- default-psp  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: RoleBinding  
metadata:  
name: privileged-role-bind  
namespace: psp-test  
roleRef:  
apiGroup: rbac.authorization.k8s.io  
kind: ClusterRole  
name: use-privileged-psp  
subjects:  
- kind: ServiceAccount  
name: privileged-sa  
$ kubectl -n psp-test apply -f clusterrole-use-privileged.yaml  
After a few moments, the privileged Pod should be created.  
Create a new PodSecurityPolicy named prevent-privileged-policy which prevents the creation of privileged pods.  
apiVersion: policy/v1beta1  
kind: PodSecurityPolicy  
metadata:  
name: example  
spec:  
privileged: false # Don't allow privileged pods!  
# The rest fills in some required fields.  
seLinux:  
rule: RunAsAny  
supplementalGroups:  
rule: RunAsAny  
runAsUser:  
rule: RunAsAny  
fsGroup:  
rule: RunAsAny  
volumes:  
- '\*'  
And create it with kubectl:  
kubectl-admin create -f example-psp.yaml  
Now, as the unprivileged user, try to create a simple pod:  
kubectl-user create -f- <<EOF  
apiVersion: v1  
kind: Pod  
metadata:  
name: pause  
spec:  
containers:  
- name: pause  
image: k8s.gcr.io/pause  
EOF  
The output is similar to this:  
Error from server (Forbidden): error when creating 'STDIN': pods 'pause' is forbidden: unable to validate against any pod security policy: []  
Create a new ServiceAccount named psp-sa in the namespace default.  
$ cat clusterrole-use-privileged.yaml  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRole  
metadata:  
name: use-privileged-psp  
rules:  
- apiGroups: ['policy']  
resources: ['podsecuritypolicies']  
verbs: ['use']  
resourceNames:  
- default-psp  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: RoleBinding  
metadata:  
name: privileged-role-bind  
namespace: psp-test  
roleRef:  
apiGroup: rbac.authorization.k8s.io  
kind: ClusterRole  
name: use-privileged-psp  
subjects:  
- kind: ServiceAccount  
name: privileged-sa  
$ kubectl -n psp-test apply -f clusterrole-use-privileged.yaml  
After a few moments, the privileged Pod should be created.  
Create a new ClusterRole named prevent-role, which uses the newly created Pod Security Policy prevent-privileged-policy.  
apiVersion: policy/v1beta1  
kind: PodSecurityPolicy  
metadata:  
name: example  
spec:  
privileged: false # Don't allow privileged pods!  
# The rest fills in some required fields.  
seLinux:  
rule: RunAsAny  
supplementalGroups:  
rule: RunAsAny  
runAsUser:  
rule: RunAsAny  
fsGroup:  
rule: RunAsAny  
volumes:  
- '\*'  
And create it with kubectl:  
kubectl-admin create -f example-psp.yaml  
Now, as the unprivileged user, try to create a simple pod:  
kubectl-user create -f- <<EOF  
apiVersion: v1  
kind: Pod  
metadata:  
name: pause  
spec:  
containers:  
- name: pause  
image: k8s.gcr.io/pause  
EOF  
The output is similar to this:  
Error from server (Forbidden): error when creating 'STDIN': pods 'pause' is forbidden: unable to validate against any pod security policy: []  
Create a new ClusterRoleBinding named prevent-role-binding, which binds the created ClusterRole prevent-role to the created SA psp-sa.  
apiVersion: rbac.authorization.k8s.io/v1  
# This role binding allows 'jane' to read pods in the 'default' namespace.  
# You need to already have a Role named 'pod-reader' in that namespace.  
kind: RoleBinding  
metadata:  
name: read-pods  
namespace: default  
subjects:  
# You can specify more than one 'subject'  
- kind: User  
name: jane # 'name' is case sensitive  
apiGroup: rbac.authorization.k8s.io  
roleRef:  
# 'roleRef' specifies the binding to a Role / ClusterRole  
kind: Role #this must be Role or ClusterRole  
name: pod-reader # this must match the name of the Role or ClusterRole you wish to bind to  
apiGroup: rbac.authorization.k8s.io  
apiVersion: rbac.authorization.k8s.io/v1  
kind: Role  
metadata:  
namespace: default  
name: pod-reader  
rules:  
- apiGroups: [''] # '' indicates the core API group  
resources: ['pods']  
verbs: ['get', 'watch', 'list']

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-7-discussion)

**Suggested Answer:** *A*

by  Raina at *Dec 03, 2021, 06:05 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 8 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 8  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

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 theAPI server:-

a. Ensure the --authorization-mode argument includes RBAC

b. Ensure the --authorization-mode argument includes Node

c. Ensure that the --profiling argument is set to false

Fix all of the following violations that were found against theKubelet:-

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

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

Fix all of the following violations that were found against theETCD:-

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

Hint: Take the use of Tool Kube-Bench

**A**Explanation:  
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 - Buildtime  
Kubernetes  
apiVersion: 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.yamlon the master node and set the--authorization-modeparameter to a value that includesNode.  
--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.yamlon 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 theKubelet:-  
Ensure the --anonymous-auth argument is set to false.  
Remediation:If using a Kubelet config file, edit the file to set authentication:anonymous: enabled tofalse. If using executable arguments, edit the kubelet service file/etc/systemd/system/kubelet.service.d/10-kubeadm.confon each worker node and set the below parameter inKUBELET\_SYSTEM\_PODS\_ARGSvariable.  
--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 theETCD:-  
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 - Buildtime  
Kubernetes  
apiVersion: 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: {}

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-8-discussion)

**Suggested Answer:** *A*

by  Roosevelt at *Dec 05, 2021, 11:44 AM*

[**https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-8-discussion**](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-8-discussion)

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 9 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 9  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

SIMULATION

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**Explanation:  
A service account provides an identity for processes that run in a Pod.  
When you (a human) access the cluster (for example, usingkubectl), you are authenticated by the apiserver as a particular User Account (currently this is usuallyadmin, 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 thedefaultservice 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 thespec.serviceAccountNamefield has beenautomatically set.  
You can access the API from inside a pod using automatically mounted service account credentials, as described inAccessing the Cluster. The API permissions of the service account depend on theauthorization plugin and policyin use.  
In version 1.6+, you can opt out of automounting API credentials for a service account by settingautomountServiceAccountToken: falseon 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 aautomountServiceAccountTokenvalue.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-9-discussion)

**Suggested Answer:** *A*

by  Carmelina at *Dec 05, 2021, 09:34 AM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 10 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 10

Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

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:

1.2.7authorization-modeargument is not set toAlwaysAllow FAIL

1.2.8authorization-modeargument includesNode FAIL

1.2.7authorization-modeargument includesRBAC FAIL

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

4.2.1 Ensure that theanonymous-auth argumentis set to false FAIL

4.2.2authorization-modeargument is not set to AlwaysAllow FAIL (UseWebhookautumn/authz where possible)

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

2.2 Ensure that theclient-cert-authargument is set to true

**A**Explanation:  
worker1 $ vim /var/lib/kubelet/config.yaml  
anonymous:  
enabled: true #Delete this  
enabled: false #Replace by this  
authorization:  
mode: AlwaysAllow #Delete this  
mode: Webhook #Replace by this  
worker1 $ systemctl restart kubelet. # To reload kubelet config  
ssh to master1  
master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml  
- -- authorization-mode=Node,RBAC  
master1 $ vim /etc/kubernetes/manifests/etcd.yaml  
- --client-cert-auth=true  
Explanation  
ssh to worker1  
worker1 $ 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 config  
ssh to master1  
master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml

Text

Description automatically generated

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

Text

Description automatically generated

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-10-discussion)

**Suggested Answer:** *A*

by  Stephen at *Dec 08, 2021, 12:39 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 11 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 11  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

use the Trivy to scan the following images,

**A**1. amazonlinux:1  
2. k8s.gcr.io/kube-controller-manager:v1.18.6  
Look for images with HIGH or CRITICAL severity vulnerabilities and store the output of the same in /opt/trivy-vulnerable.txt

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-11-discussion)

**Suggested Answer:** *A*

by  Laurel at *Dec 05, 2021, 01:16 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 12 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 12  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

a. Retrieve the content of the existing secret nameddefault-token-xxxxxin the testing namespace.

Store the value of the token in the token.txt

b. Create a new secret named test-db-secret in the DB namespace with the following content:

username:mysql

password:password@123

Create the Pod name test-db-pod of image nginx in the namespace db that can access test-db-secret via a volume at path /etc/mysql-credentials

**A**Explanation:  
To add a Kubernetes cluster to your project, group, or instance:  
Navigate to your:  
Project'sOperations > Kubernetespage, for a project-level cluster.  
Group'sKubernetespage, for a group-level cluster.  
Admin Area >Kubernetespage, for an instance-level cluster.  
ClickAdd Kubernetes cluster.  
Click theAdd existing clustertab and fill in the details:  
Kubernetes cluster name(required) - The name you wish to give the cluster.  
Environment scope(required) - Theassociated environmentto this cluster.  
API URL(required) - It's the URL that GitLab uses to access the Kubernetes API. Kubernetes exposes several APIs, we want the ''base'' URL that is common to all of them. For example,https://kubernetes.example.comrather thanhttps://kubernetes.example.com/api/v1.  
Get the API URL by running this command:  
kubectl cluster-info | grep -E 'Kubernetes master|Kubernetes control plane' | awk '/http/ {print $NF}'  
CA certificate(required) - A valid Kubernetes certificate is needed to authenticate to the cluster. We use the certificate created by default.  
List the secrets withkubectl get secrets, and one should be named similar todefault-token-xxxxx. Copy that token name for use below.  
Get the certificate by running this command:  
kubectl get secret <secret name> -o jsonpath='{['data']['ca\.crt']}'

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-12-discussion)

**Suggested Answer:** *A*

by  Stephaine at *Dec 08, 2021, 10:39 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 13 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 13  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

Create a new NetworkPolicy named deny-all in the namespace testing which denies all traffic of type ingress and egress traffic

**A**Explanation:  
You can create a 'default' isolation policy for a namespace by creating a NetworkPolicy that selects all pods but does not allow any ingress traffic to those pods.  
---  
apiVersion: networking.k8s.io/v1  
kind: NetworkPolicy  
metadata:  
name: default-deny-ingress  
spec:  
podSelector: {}  
policyTypes:  
- Ingress  
You can create a 'default' egress isolation policy for a namespace by creating a NetworkPolicy that selects all pods but does not allow any egress traffic from those pods.  
---  
apiVersion: networking.k8s.io/v1  
kind: NetworkPolicy  
metadata:  
name: allow-all-egress  
spec:  
podSelector: {}  
egress:  
- {}  
policyTypes:  
- Egress  
Default deny all ingress and all egress traffic  
You can create a 'default' policy for a namespace which prevents all ingress AND egress traffic by creating the following NetworkPolicy in that namespace.  
---  
apiVersion: networking.k8s.io/v1  
kind: NetworkPolicy  
metadata:  
name: default-deny-all  
spec:  
podSelector: {}  
policyTypes:  
- Ingress  
- Egress  
This ensures that even pods that aren't selected by any other NetworkPolicy will not be allowed ingress or egress traffic.

[Show Suggested Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-13-discussion)

by  Stephane at *Dec 07, 2021, 08:50 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 14 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 14  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

Create a network policy named restrict-np to restrict to pod nginx-test running in namespace testing.

Only allow the following Pods to connect to Pod nginx-test:-

1. pods in the namespace default

2. pods with label version:v1 in any namespace.

Make sure to apply the network policy.

**A**Explanation:

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-14-discussion)

**Suggested Answer:** *A*

by  Jolanda at *Dec 07, 2021, 03:59 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 15 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 15  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

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

**A**store the incident file art /opt/falco-incident.txt, containing the detected incidents. one per line, in the format  
[timestamp],[uid],[processName]

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-15-discussion)

**Suggested Answer:** *A*

by  Elvera at *Jan 02, 2022, 04:55 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 16 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 16  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

Use the kubesec docker images to scan the given YAML manifest, edit and apply the advised changes, and passed with a score of 4 points.

kubesec-test.yaml

apiVersion: v1

kind: Pod

metadata:

name: kubesec-demo

spec:

containers:

- name: kubesec-demo

image: gcr.io/google-samples/node-hello:1.0

securityContext:

readOnlyRootFilesystem: true

**A**Hint:docker run -i kubesec/kubesec:512c5e0 scan /dev/stdin < kubesec-test.yaml

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-16-discussion)

**Suggested Answer:** *A*

by  Terrilyn at *Jan 26, 2022, 06:14 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 17 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 17  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

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:-

**A**ETCDCTL\_API=3 etcdctl get /registry/secrets/default/cks-secret --cacert='ca.crt' --cert='server.crt' --key='server.key'  
Output

Text

Description automatically generated

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.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-17-discussion)

**Suggested Answer:** *A*

by  Alyce at *Jan 27, 2022, 10:06 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 18 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 18  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

On the Cluster worker node, enforce the prepared AppArmor profile

#include

profile docker-nginx flags=(attach\_disconnected,mediate\_deleted) {

#include

network inet tcp,

network inet udp,

network inet icmp,

deny network raw,

deny network packet,

file,

umount,

deny /bin/\*\* wl,

deny /boot/\*\* wl,

deny /dev/\*\* wl,

deny /etc/\*\* wl,

deny /home/\*\* wl,

deny /lib/\*\* wl,

deny /lib64/\*\* wl,

deny /media/\*\* wl,

deny /mnt/\*\* wl,

deny /opt/\*\* wl,

deny /proc/\*\* wl,

deny /root/\*\* wl,

deny /sbin/\*\* wl,

deny /srv/\*\* wl,

deny /tmp/\*\* wl,

deny /sys/\*\* wl,

deny /usr/\*\* wl,

audit /\*\* w,

/var/run/nginx.pid w,

/usr/sbin/nginx ix,

deny /bin/dash mrwklx,

deny /bin/sh mrwklx,

deny /usr/bin/top mrwklx,

capability chown,

capability dac\_override,

capability setuid,

capability setgid,

capability net\_bind\_service,

deny @{PROC}/\* w, # deny write for all files directly in /proc (not in a subdir)

# deny write to files not in /proc//\*\* or /proc/sys/\*\*

deny @{PROC}/{[^1-9],[^1-9][^0-9],[^1-9s][^0-9y][^0-9s],[^1-9][^0-9][^0-9][^0-9]\*}/\*\* w,

deny @{PROC}/sys/[^k]\*\* w, # deny /proc/sys except /proc/sys/k\* (effectively /proc/sys/kernel)

deny @{PROC}/sys/kernel/{?,??,[^s][^h][^m]\*\*} w, # deny everything except shm\* in /proc/sys/kernel/

deny @{PROC}/sysrq-trigger rwklx,

deny @{PROC}/mem rwklx,

deny @{PROC}/kmem rwklx,

deny @{PROC}/kcore rwklx,

deny mount,

deny /sys/[^f]\*/\*\* wklx,

deny /sys/f[^s]\*/\*\* wklx,

deny /sys/fs/[^c]\*/\*\* wklx,

deny /sys/fs/c[^g]\*/\*\* wklx,

deny /sys/fs/cg[^r]\*/\*\* wklx,

deny /sys/firmware/\*\* rwklx,

deny /sys/kernel/security/\*\* rwklx,

}

Edit the prepared manifest file to include the AppArmor profile.

apiVersion: v1

kind: Pod

metadata:

name: apparmor-pod

spec:

containers:

- name: apparmor-pod

image: nginx

Finally, apply the manifests files and create the Pod specified on it.

Verify: Try to use commandping, top, sh

**A**Explanation:

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-18-discussion)

**Suggested Answer:** *A*

by  Micaela at *Jan 29, 2022, 07:01 AM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 19 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 19  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

A container image scanner is set up on the cluster.

Given an incomplete configuration in the directory

**A**1. Enable the admission plugin.  
2. Validate the control configuration and change it to implicit deny.  
Finally, test the configuration by deploying the pod having the image tag as the latest.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-19-discussion)

**Suggested Answer:** *A*

by  Salena at *Jan 31, 2022, 07:49 AM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 20 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 20  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

A container image scanner is set up on the cluster.

Given an incomplete configuration in the directory

**A**1. Enable the admission plugin.  
2. Validate the control configuration and change it to implicit deny.  
Finally, test the configuration by deploying the pod having the image tag as latest.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-20-discussion)

**Suggested Answer:** *A*

by  Dalene at *Jan 24, 2022, 01:49 PM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 21 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 21  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

Create a PSP that will prevent the creation of privileged pods in the namespace.

Create a new PodSecurityPolicy named prevent-privileged-policy which prevents the creation of privileged pods.

Create a new ServiceAccount named psp-sa in the namespace default.

Create a new ClusterRole named prevent-role, which uses the newly created Pod Security Policy prevent-privileged-policy.

Create a new ClusterRoleBinding named prevent-role-binding, which binds the created ClusterRole prevent-role to the created SA psp-sa.

Also, Check the Configuration is working or not by trying to Create a Privileged pod, it should get failed.

**A**Explanation:  
Create a PSP that will prevent the creation of privileged pods in the namespace.  
$ cat clusterrole-use-privileged.yaml  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRole  
metadata:  
name: use-privileged-psp  
rules:  
- apiGroups: ['policy']  
resources: ['podsecuritypolicies']  
verbs: ['use']  
resourceNames:  
- default-psp  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: RoleBinding  
metadata:  
name: privileged-role-bind  
namespace: psp-test  
roleRef:  
apiGroup: rbac.authorization.k8s.io  
kind: ClusterRole  
name: use-privileged-psp  
subjects:  
- kind: ServiceAccount  
name: privileged-sa  
$ kubectl -n psp-test apply -f clusterrole-use-privileged.yaml  
After a few moments, the privileged Pod should be created.  
Create a new PodSecurityPolicy named prevent-privileged-policy which prevents the creation of privileged pods.  
apiVersion: policy/v1beta1  
kind: PodSecurityPolicy  
metadata:  
name: example  
spec:  
privileged: false # Don't allow privileged pods!  
# The rest fills in some required fields.  
seLinux:  
rule: RunAsAny  
supplementalGroups:  
rule: RunAsAny  
runAsUser:  
rule: RunAsAny  
fsGroup:  
rule: RunAsAny  
volumes:  
- '\*'  
And create it with kubectl:  
kubectl-admin create -f example-psp.yaml  
Now, as the unprivileged user, try to create a simple pod:  
kubectl-user create -f- <<EOF  
apiVersion: v1  
kind: Pod  
metadata:  
name: pause  
spec:  
containers:  
- name: pause  
image: k8s.gcr.io/pause  
EOF  
The output is similar to this:  
Error from server (Forbidden): error when creating 'STDIN': pods 'pause' is forbidden: unable to validate against any pod security policy: []  
Create a new ServiceAccount named psp-sa in the namespace default.  
$ cat clusterrole-use-privileged.yaml  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRole  
metadata:  
name: use-privileged-psp  
rules:  
- apiGroups: ['policy']  
resources: ['podsecuritypolicies']  
verbs: ['use']  
resourceNames:  
- default-psp  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: RoleBinding  
metadata:  
name: privileged-role-bind  
namespace: psp-test  
roleRef:  
apiGroup: rbac.authorization.k8s.io  
kind: ClusterRole  
name: use-privileged-psp  
subjects:  
- kind: ServiceAccount  
name: privileged-sa  
$ kubectl -n psp-test apply -f clusterrole-use-privileged.yaml  
After a few moments, the privileged Pod should be created.  
Create a new ClusterRole named prevent-role, which uses the newly created Pod Security Policy prevent-privileged-policy.  
apiVersion: policy/v1beta1  
kind: PodSecurityPolicy  
metadata:  
name: example  
spec:  
privileged: false # Don't allow privileged pods!  
# The rest fills in some required fields.  
seLinux:  
rule: RunAsAny  
supplementalGroups:  
rule: RunAsAny  
runAsUser:  
rule: RunAsAny  
fsGroup:  
rule: RunAsAny  
volumes:  
- '\*'  
And create it with kubectl:  
kubectl-admin create -f example-psp.yaml  
Now, as the unprivileged user, try to create a simple pod:  
kubectl-user create -f- <<EOF  
apiVersion: v1  
kind: Pod  
metadata:  
name: pause  
spec:  
containers:  
- name: pause  
image: k8s.gcr.io/pause  
EOF  
The output is similar to this:  
Error from server (Forbidden): error when creating 'STDIN': pods 'pause' is forbidden: unable to validate against any pod security policy: []  
Create a new ClusterRoleBinding named prevent-role-binding, which binds the created ClusterRole prevent-role to the created SA psp-sa.  
apiVersion: rbac.authorization.k8s.io/v1  
# This role binding allows 'jane' to read pods in the 'default' namespace.  
# You need to already have a Role named 'pod-reader' in that namespace.  
kind: RoleBinding  
metadata:  
name: read-pods  
namespace: default  
subjects:  
# You can specify more than one 'subject'  
- kind: User  
name: jane # 'name' is case sensitive  
apiGroup: rbac.authorization.k8s.io  
roleRef:  
# 'roleRef' specifies the binding to a Role / ClusterRole  
kind: Role #this must be Role or ClusterRole  
name: pod-reader # this must match the name of the Role or ClusterRole you wish to bind to  
apiGroup: rbac.authorization.k8s.io  
apiVersion: rbac.authorization.k8s.io/v1  
kind: Role  
metadata:  
namespace: default  
name: pod-reader  
rules:  
- apiGroups: [''] # '' indicates the core API group  
resources: ['pods']  
verbs: ['get', 'watch', 'list']

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-21-discussion)

**Suggested Answer:** *A*

by  Paris at *Jan 28, 2022, 02:46 AM*

**LINUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 22 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 22  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

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 theAPI server:-

a. Ensure the --authorization-mode argument includes RBAC

b. Ensure the --authorization-mode argument includes Node

c. Ensure that the --profiling argument is set to false

Fix all of the following violations that were found against theKubelet:-

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

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

Fix all of the following violations that were found against theETCD:-

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

Hint: Take the use of Tool Kube-Bench

**A**Explanation:  
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 - Buildtime  
Kubernetes  
apiVersion: 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.yamlon the master node and set the--authorization-modeparameter to a value that includesNode.  
--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.yamlon 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 theKubelet:-  
Ensure the --anonymous-auth argument is set to false.  
Remediation:If using a Kubelet config file, edit the file to set authentication:anonymous: enabled tofalse. If using executable arguments, edit the kubelet service file/etc/systemd/system/kubelet.service.d/10-kubeadm.confon each worker node and set the below parameter inKUBELET\_SYSTEM\_PODS\_ARGSvariable.  
--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 theETCD:-  
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 - Buildtime  
Kubernetes  
apiVersion: 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: {}

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-22-discussion)

**Suggested Answer:** *A*

by  Cecilia at *Feb 13, 2022, 05:27 PM*

**INUX FOUNDATION EXAM CKS TOPIC 1 QUESTION 23 DISCUSSION**

Actual exam question for Linux Foundation's Certified Kubernetes Security Specialist exam

Question #: 23  
Topic #: 1

[**[All Certified Kubernetes Security Specialist Questions]**](https://www.pass4success.com/linux-foundation/exam/cks)

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**Explanation:  
A service account provides an identity for processes that run in a Pod.  
When you (a human) access the cluster (for example, usingkubectl), you are authenticated by the apiserver as a particular User Account (currently this is usuallyadmin, 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 thedefaultservice 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 thespec.serviceAccountNamefield has beenautomatically set.  
You can access the API from inside a pod using automatically mounted service account credentials, as described inAccessing the Cluster. The API permissions of the service account depend on theauthorization plugin and policyin use.  
In version 1.6+, you can opt out of automounting API credentials for a service account by settingautomountServiceAccountToken: falseon 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 aautomountServiceAccountTokenvalue.

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-cks-topic-1-question-23-discussion)

**Suggested Answer:** *A*

by  Carmelina at *Feb 14, 2022, 08:58 AM*