

## Exam Questions CKS

Certified Kubernetes Security Specialist (CKS) Exam

<https://www.2passeasy.com/dumps/CKS/>



### NEW QUESTION 1

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

- A. Mastered
- B. Not Mastered

**Answer:** 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 trafficYou 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.

### NEW QUESTION 2

Enable audit logs in the cluster, To Do so, enable the log backend, and ensure that

- \* 1. logs are stored at /var/log/kubernetes-logs.txt.
- \* 2. Log files are retained for 12 days.
- \* 3. at maximum, a number of 8 old audit logs files are retained.
- \* 4. set the maximum size before getting rotated to 200MB

Edit and extend the basic policy to log:

- \* 1. namespaces changes at RequestResponse
- \* 2. Log the request body of secrets changes in the namespace kube-system.
- \* 3. Log all other resources in core and extensions at the Request level.
- \* 4. Log "pods/portforward", "services/proxy" at Metadata level.
- \* 5. Omit the Stage RequestReceived

All other requests at the Metadata level

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

Kubernetes auditing provides a security-relevant chronological set of records about a cluster. Kube-apiserver performs auditing. Each request on each stage of its execution generates an event, which is then pre-processed according to a certain policy and written to a backend. The policy determines what's recorded and the backends persist the records.

You might want to configure the audit log as part of compliance with the CIS (Center for Internet Security) Kubernetes Benchmark controls.

The audit log can be enabled by default using the following configuration in cluster.yml:

```
services:
  kube-api:
    audit_log:
      enabled:true
```

When the audit log is enabled, you should be able to see the default values at

/etc/kubernetes/audit-policy.yaml

The log backend writes audit events to a file in JSONlines format. You can configure the log audit backend using the following kube-apiserver flags:

➤ --audit-log-path specifies the log file path that log backend uses to write audit events. Not specifying thi flag disables log backend. - means standard out

- > --audit-log-maxbackup defines the maximum number of audit log files to retain
- > --audit-log-maxsize defines the maximum size in megabytes of the audit log file before it gets rotated

If your cluster's control plane runs the kube-apiserver as a Pod, remember to mount the location of the policy file and log file, so that audit records are persisted.

For example: -hostPath-to the

--audit-policy-file=/etc/kubernetes/audit-policy.yaml\

--audit-log-path=/var/log/audit.log-

### NEW QUESTION 3

On the Cluster worker node, enforce the prepared AppArmor profile

```
#include<tunables/global>
profile docker-nginx flags=(attach_disconnected,mediate_deleted) {
#include<abstractions/base>
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/<number>/** 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 command ping, top, sh

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Send us your feedback on it.

**NEW QUESTION 4**

Create a network policy named allow-np, that allows pod in the namespace staging to connect to port 80 of other pods in the same namespace.

Ensure that Network Policy:

- \* 1. Does not allow access to pod not listening on port 80.
- \* 2. Does not allow access from Pods, not in namespace staging.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
apiVersion:networking.k8s.io/v1
kind:NetworkPolicy
metadata:
name:network-policy
spec:
podSelector:{} #selects all the pods in the namespace deployed
policyTypes:
-Ingress
ingress:
-ports:#in input traffic allowed only through 80 port only
-protocol:TCP
port:80
```

**NEW QUESTION 5**

Before Making any changes build the Dockerfile with tag base:v1 Now Analyze and edit the given Dockerfile(based on ubuntu 16:04)

Fixing two instructions present in the file, Check from Security Aspect and Reduce Size point of view.

Dockerfile:

```
FROM ubuntu:latest
RUN apt-getupdate -y
RUN apt install nginx -y
COPY entrypoint.sh /
RUN useradd ubuntu
ENTRYPOINT ["/entrypoint.sh"]
USER ubuntu
entrypoint.sh
#!/bin/bash
echo"Hello from CKS"
```

After fixing the Dockerfile, build the docker-image with the tag base:v2 To Verify: Check the size of the image before and after the build.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Send us your feedback on it.

**NEW QUESTION 10**

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual CKS Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the CKS Product From:

<https://www.2passeasy.com/dumps/CKS/>

## Money Back Guarantee

### CKS Practice Exam Features:

- \* CKS Questions and Answers Updated Frequently
- \* CKS Practice Questions Verified by Expert Senior Certified Staff
- \* CKS Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* CKS Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year