LinuxFoundation.CKS.v2024-03-01.q40

□□□□:	CKS
□□□□:	Certified Kubernetes Security Specialist (CKS)
□□□:	Linux Foundation
□□ □□ □□□:	40
□□:	v2024-03-01
# □□ □:	825
# □□ □□□:	400
https://www.krdump.com/LinuxFoundation.CKS.v2024-03-01.q40.html	

NEW QUESTION: 1
kubesec docker □□□□ □□□□ □□□ YAML □□□□□□ □□□□ □□□ □□ □□□ □□ □ □□□ □ 4□□□ □□□□□□.
kubesec-test.yaml
api□□: v1
□□: □□
□□□□□:
□□: kubesec-demo
□□:
□□□□:
- □□: kubesec-demo
□□□: gcr.io/google-samples/node-hello:1.0
□□□□□□:
readOnlyRootFilesystem: true
Answer:
□□: docker run -i kubesec/kubesec:512c5e0 scan /dev/stdin < kubesec-test.yaml

NEW QUESTION: 2
□□□□□
etcd□ □□□ □□□ □□ □ □□□□ □□□□. etcdctl □□ □□□□□ □□□□ □□□ □□ □□ □□ □□ □ □□□□.
ETCDCTL_API=3 etcdctl get /registry/secrets/default/cks-secret --cacert="ca.crt " --cert="server.crt" --key="server.key" □□

□□□ □□□ □□□□ □□□ AES-CBC □ ID□ □□□□ □□□ □□□ □□□□ □□□□□□ □□□□ □□ □□ □□ □□□□ □□□ □□ □□ □□□ □ □□□□ □□□□□□ □□□.

Answer:

□□ □□ □□□□ □□□□□.

NEW QUESTION: 3

□□□□: qa-□□□□

□□□ □□: □□□ □□□ □□: Worker1

□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□.

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

□:

□□□□□□ □□□□ □□ □□ □□ □□□ □□ □□□□ □□□□ □□ □□□ □□□□□ NetworkPolicy□ □□□□□.

□□ Pod□ Pod □□ □□□□ □□□□□ □□□□□.

1. □□□□□□ qa□ □□

2. □□ □□□ □□ □□: □□ □□□□□□□ □□□□

Answer:

$ k□ ns qa□ □□□□ --show-labels

□□ □□ □□ □□

qa □□ 4,700□ □□=□□□□

$ k □□ □□□□ -n dev --show-labels

□□ □□ □□ □□ □□ □□ □□

□□ 1/1 □□ □ 0 3s env=dev-team

API□□:networking.k8s.io/v1

□□: NetworkPolicy

□□□□□:

□□: □□□ □□

□□□□□□: dev

□□:

□□□□□:

□□ □□:

env: □□□

□□ □□:

- □□□□

□□:

- □□:

- □□□□□□□□□:

□□ □□:

□□ : □□

- □□□□□:

□□ □□:

□□ : □□
[desk@cli] $ k get ns qa --show-labels
□□ □□ □□ □□
qa □□ 4,700□ □□=□□□□
[desk@cli] $ k □□ □□□□ -n dev --show-labels
□□ □□ □□ □□ □□ □□ □□
□□ 1/1 □□ □ 0 3s env=dev-team
[desk@cli] $ vim netpol2.yaml
API□□:networking.k8s.io/v1
□□: NetworkPolicy
□□□□□:
□□: □□□ □□
□□□□□□: dev
□□:
□□□□□:
□□ □□:
env: □□□
□□ □□:
- □□□□
□□:
- □□:
- □□□□□□□□□:
□□ □□:
□□ : □□
- □□□□□:
□□ □□:
□□ : □□
[desk@cli] $ k apply -f netpol2.yaml □□: https://kubernetes.io/docs/concepts/services-networking/network-policies/
[desk@cli] $ k apply -f netpol2.yaml □□: https://kubernetes.io/docs/concepts/services-networking/network-policies/

NEW QUESTION: 4
□□□ Dockerfile□ □□□□ □□□□□.
□□□□□:□□
□□ apt-get □□□□ -y
□□ apt-install nginx -y
□□ Entrypoint.sh /
□□□ ["/entrypoint.sh"]
□□□ □□
□□ □□ □□ □□ □□ □□□ □□□ □□ □ □□ □□□ □□□□□. □□ □□□□□ □□ apiVersion: v1 □□: Pod □□□□□□ □□ □□ □□□□□.

□□: □□-□□□□-□□-2
□□:
□□□□□□:
runAs□□□: 1000
□□□□:
- □□: sec-ctx-demo-2
□□□: gcr.io/google-samples/node-hello:1.0
□□□□□□:
□□ □□□: 0
□□: □□
allowPrivilegeEscalation: false
□□ □□ □□ □□ □□ □□□ □□□ □□ □ □□□ □□□□□. □□ □□□ □□□□□ □□□□ □□□. □□ □□ □□□ □□□□□ □. □□□ □□□ □□ □□□□ □□□ □□□ □□□ ID 5487□ □□□ test-user□ □□□□□□.
Answer:
□□ □□ □□□□ □□□□□.

NEW QUESTION: 5
□□□□ □□ base:v1 □□□ □□□□ Dockerfile□ □□□□□.
□□ □□□ Dockerfile□ □□□□ □□□□□□(ubuntu 16:04 □□).
□□□ □□ □ □□ □□□ □□ □□□□ □□ □ □□ □□ □□□ □□□□□.
□□□□:
□□□□□:□□
□□ apt-get □□□□ -y
□□ apt install nginx -y
□□ Entrypoint.sh /
RUN useradd □□□
□□□ ["/entrypoint.sh"]
□□□ □□□
□□□.sh
#!/bin/bash
echo "□□□□□ CKS□□□"
Dockerfile□ □□□ □ base:v2 □□□ □□□□ docker-image□ □□□□□.
Answer:
□□□□□: □□ □□□ □□□ □□□ □□□□□.

NEW QUESTION: 6
□□□□□
□□□□ □□□ □□□□ □□□□□ □□□□□.
□□□□□ □□□□ □□□ □□ □□
/etc/kubernetes/confcontrol □ HTTPS □□□□□ https://test-server.local.8081/image_policy□ □□ □□□ □□□□ □□□ □□□

1. □□ □□□□□ □□□□□□.
2. □□ □□□ □□□□ □□□ □□□ □□□□□.
□□□□□ □□□ □□□ □□□ Pod□ □□□□ □□□ □□□□□□.
Answer:
□□ □□ □□□□ □□□□□.

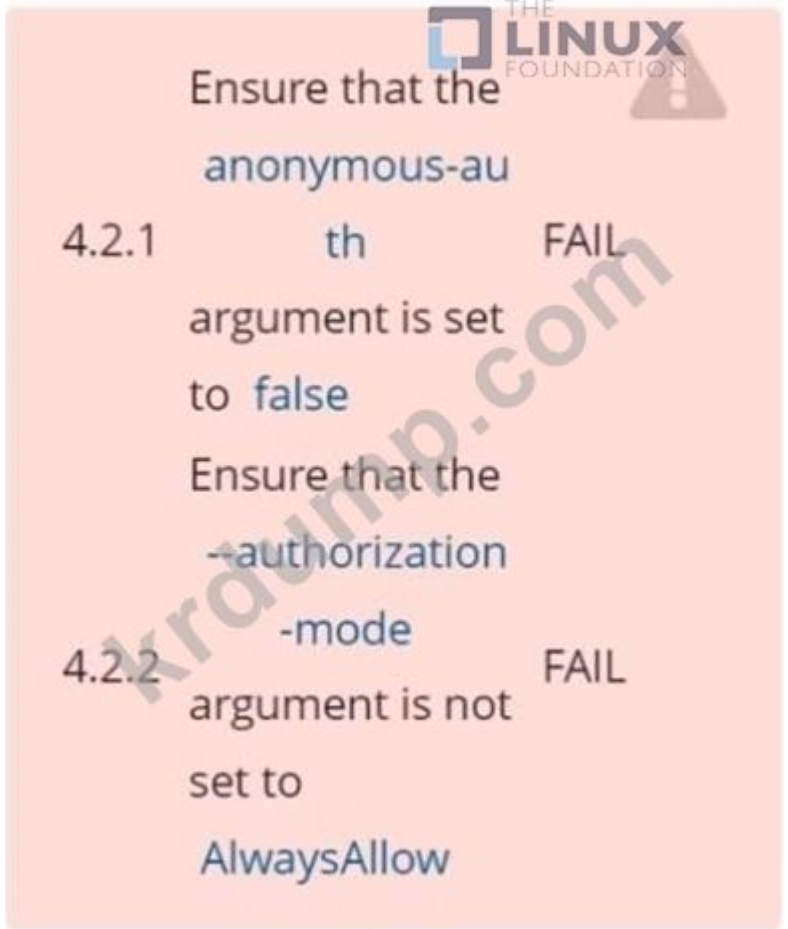
NEW QUESTION: 7
□□□□□
□□□□ □□□ □□□□ □□□ AppArmor □□□□ □□□□□.
#include <□□□/□□>
□□□ docker-nginx □□□=(attach_disconnected,mediate_deleted) {
#include <□□□/□□□>
□□□□ INET TCP,
□□□□ INET UDP,
□□□□ Inet ICMP,
□□□□ □□ □□,
□□□□ □□ □□,
□□,
□□□ □□,
/bin/** wl □□,
/boot/** wl □□,
/dev/** wl □□,
/etc/** wl □□,
/home/** wl □□,
/lib/** wl □□,
/lib64/** wl □□,
/media/** wl □□,
/mnt/** wl □□,
□□ /opt/** wl,
/proc/** wl □□,
/root/** wl □□,
/sbin/** wl □□,
/srv/** wl □□,
/tmp/** wl □□,
/sys/** wl □□,
/usr/** wl □□,
□□ /** w,
/var/run/nginx.pid w,
/usr/sbin/nginx ix,
/bin/dash mrwklx □□,

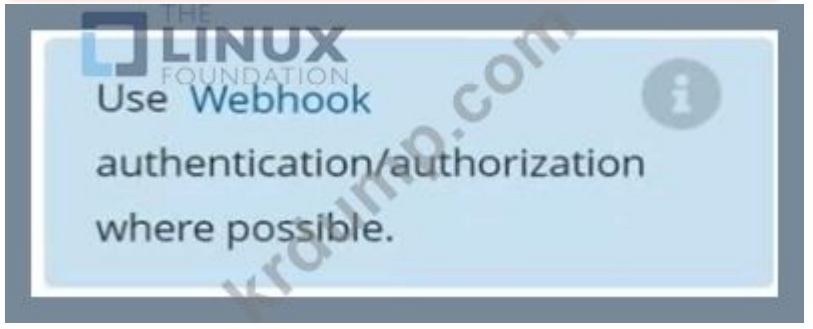
/bin/sh mrwklx □□,
/usr/bin/top mrwklx □□,
□□ chown,
dac_override □□,
□□ setuid,
□□ □□,
net_bind_service □□,
□□ @{PROC}/* w, # /proc(□□ □□□□ □□)□□ □□ □□ □□□ □□ □□ □□
/proc/<number>/** □□ /proc/sys/**□ □□ □□□ □□ □□□ □□□□□.
@{PROC}/{[^1-9],[^1-9][^0-9],[^1-9s][^0-9y][^0-9s],[^1- □□ 9][^0-9][^0-9][^0-9]*}/** w, @{PROC}/sys/[^k]** w, # □□ /proc/sys □
□ /proc/sys/k* (□□□□□□ /proc/sys/kernel) @{PROC}/sys/kernel/{?,??,[^s][^h][^m]**} □□ w, # /proc/sys/kernel/□□ shm*□ □□□ □□
□□ □□ @{PROC}/sysrq-trigger rwklx □□, @{PROC}/mem rwklx □□, @{PROC}/kmem rwklx □□, @{PROC}/kcore rwklx □□ , □□□
□□, /sys/[^f]*/** wklx □□, /sys/f[^s]*/** wklx □□, /sys/fs/[^c]*/** wklx □□, □□ /sys/fs/c[^g]*/** wklx, /sys/fs/cg[^r]*/** wklx □
□, /sys/firmware/** rwklx □□, /sys/kernel/security □□ /** rwklx,
}
AppArmor □□□□ □□□□□ □□□ □□□□□ □□□ □□□□□.
api□□: v1
□□: □□
□□□□□:
□□: apparmor-pod
- □□: apparmor-pod
□□□: nginx
□□□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□ □□□□□.
□□: ping, top, sh □□□ □□□ □□□□.
Answer:
□□ □□ □□□□ □□□□□.

NEW QUESTION: 8
□□
kubeadm□ □□□ □□□□□ □□ CIS □□□□ □□□ □□□ □□ □□ □□□□ □□ □□ □□□ □□□□□□□.
□
□□□ □□ □□ □□□ □□□□ □□□ □□ □□ □□□ □□ □□□□ □ □□□ □□□□□ □□□.
API □□□□ □□□ □□ □□ □□□ □□ □□□□□.

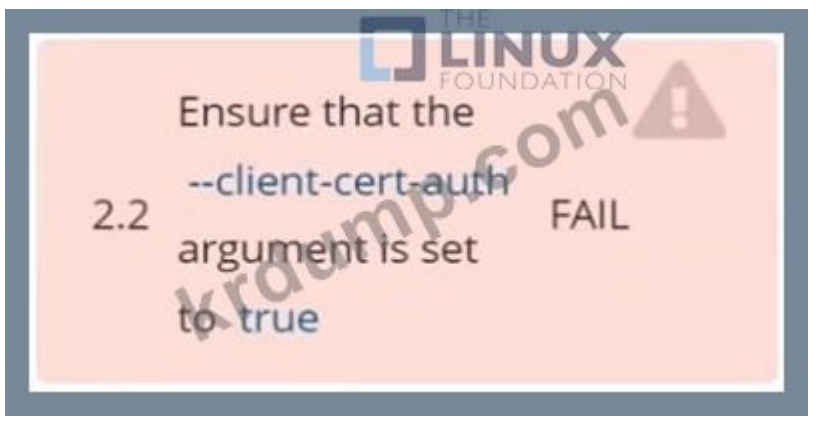
1.2.7	Ensure that the --authorization -mode argument is not set to AlwaysAllow	FAIL
1.2.8	Ensure that the --authorization -mode argument includes Node	FAIL
1.2.9	Ensure that the --authorization -mode argument includes RBAC	FAIL

Kubelet□ □□ □□□ □□ □□ □□□ □□ □□□□□.


Ensure that the
anonymous-au
4.2.1 th FAIL
argument is set
to false
Ensure that the
--authorization
-mode
4.2.2 FAIL
argument is not
set to
AlwaysAllow


Use Webhook
authentication/authorization
where possible.

etcd□ □□ □□□ □□ □□ □□□ □□ □□□□□.


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

Answer:

```
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_mana
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_mana
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 14:19:38.184042  134205 kubelet.go:1698] "Deleted mirror pod because it is outdat
5]: I0520 14:19:40.033042  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 --

:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kube-proxy\"
:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"lib-modules\"
:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"flannel-cfg\"
:157] "Reconciler: start to sync state"
55] Waited for 1.049946364s due to client-side throttling, not priority and fairness, reque
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
593] "Trying to delete pod" pod="kube-system/kube-apiserver-kscs00201-master" podUID=bb91e1
598] "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; 1s 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)
```

NEW QUESTION: 9

□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□: [desk@cli] $ kubectl config use-context dev □□□□: CIS □□□□ □ □□ kubeadm □□ □□□□□ □□ □□□□□□ □□□□ □ □□ □□□ □□□□□□. □□: □□□ □□ □□ □□□ □□□□ □□□ □□ □□ □□□ □□ □□□□ □ □□□ □□□□□ □□□. API □□□ □□ □□□ □□ □□ □□□ □□ □□□□□. 1.2.7 □□ □□ □ □□ AlwaysAllow□ □□□□ □□□□□. FAIL 1.2.8 □□ □□ □□□ □□□ □□□□□. FAIL 1.2.7 □□ □□ □□□ RBAC FAIL□ □□ □□□. Kubelet□ □□ □□□ □□ □□ □□: 4.2.1 □□ □□ □□□ false□ □□□□ □□□ □□ FAIL 4.2.2 □□ □□ □□□ AlwaysAllow□ □□□□ □□ FAIL(□□□ □□ Webhook fall/authz □□) □□ □□ etcd□□ □□□ □□ □□ □□: 2.2 client-cert-auth □ □□ true□ □□□□ □□□ □□□□□□.

Answer:

□□□1 $ vim /var/lib/kubelet/config.yaml
□□□:
□□□□: true #□□
□□□□: false #□□□□ □□
□□ □□:
□□: AlwaysAllow #Delete this
□□: □□ #□□□□ □□
Worker1 $ systemctl kubelet□ □□ □□□□□. # kubelet config ssh□ master1 master1□ □□ □□□□□ $ vim /etc/kubernetes/manifests/kube-apiserver.yaml - --authorization-mode=Node,RBAC master1 $ vim /etc/kubernetes/manifests/etcd.yaml - --client -cert-auth=true □□ ssh□□ □□□1 □□□1 $ vim /var/lib/kubelet/config.yaml apiVersion: kubelet.config.k8s.io/v1beta1 □□:
□□□:
□□□□: true #□□
□□□□: false #□□□□ □□
□□:
□□TTL: 0□
□□□□: □□
x509:
□□□□□CA□□: /etc/kubernetes/pki/ca.crt
□□ □□:
□□: AlwaysAllow #Delete this
□□: □□ #□□□□ □□
□□:
□□□□TTL: 0□
□□□□TTL: 0□
cgroupDriver: □□□□

□□□□DNS:
- 10.96.0.10
□□□□□□□: Cluster.local
cpuManagerReconcilePeriod: 0□
evictionPressureTransitionPeriod: 0□
fileCheckFrequency: 0□
healthzBind□□: 127.0.0.1
healthz□□: 10248
httpCheckFrequency: 0□
imageMinimumGCAge: 0□
□□: KubeletConfiguration
□□: {}
nodeStatusReportFrequency: 0□
nodeStatusUpdateFrequency: 0□
resolvConf: /run/systemd/resolve/resolv.conf
□□□□□: true
□□□□□ □□ □□: 0□
staticPodPath: /etc/kubernetes/manifests
StreamingConnectionIdleTimeout: 0□
□□□ □□: 0□
VolumeStatsAggPeriod: 0□
Worker1 $ systemctl kubelet□ □□ □□□□□. # kubelet config ssh□ master1 master1□ □□ □□□□□ $
vim /etc/kubernetes/manifests/kube-apiserver.yaml

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint  172.17.0.22:6443
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=172.17.0.22
    - --allow-privileged=true
#     - --authorization-mode=AlwaysAllow  # Delete This
    - --authorization-mode=Node,RBAC     # Replace by this line
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --enable-admission-plugins=NodeRestriction
    - --enable-bootstrap-token-auth=true
    - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
    - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
    - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
    - --etcd-servers=https://127.0.0.1:2379
    - --insecure-port=0
```

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

```
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/etcd.advertise-client-urls: https://172.17.0.29:2379
  creationTimestamp: null
  labels:
    component: etcd
    tier: control-plane
  name: etcd
  namespace: kube-system
spec:
  containers:
  - command:
    - etcd
    - --advertise-client-urls=https://172.17.0.29:2379
    - --cert-file=/etc/kubernetes/pki/etcd/server.crt
    - --client-cert-auth=true  #Add this line
    - --data-dir=/var/lib/etcd
    - --initial-advertise-peer-urls=https://172.17.0.29:2380
    - --initial-cluster=controlplane=https://172.17.0.29:2380
    - --key-file=/etc/kubernetes/pki/etcd/server.key
    - --listen-client-urls=https://127.0.0.1:2379,https://172.17.0.29:2379
    - --listen-metrics-urls=http://127.0.0.1:2381
    - --listen-peer-urls=https://172.17.0.29:2380
    - --name=controlplane
    - --peer-cert-file=/etc/kubernetes/pki/etcd/peer.crt
    - --peer-client-cert-auth=true
    - --peer-key-file=/etc/kubernetes/pki/etcd/peer.key
    - --peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
    - --snapshot-count=10000
    - --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
    image: k8s.gcr.io/etcd:3.4.9-1
```

NEW QUESTION: 10

□□□□□

□□□ Dockerfile□ □□□□ □□□□□.

□□□□□:□□

□□ apt-get □□□□ -y

□□ apt-install nginx -y

□□ Entrypoint.sh /

□□□ ["/entrypoint.sh"]

□□□ □□

□□ □□ □□ □□ □□ □□□ □□□ □□ □ □□ □□□ □□□□□. □□ □□□□□ □□ apiVersion: v1 □□: Pod □□□□□□ □□ □□ □□□□□.

□□: □□ł□□□□ł□□-2

□□:
□□□□□□:
runAs□□□: 1000
□□□□:
- □□: sec-ctx-demo-2
□□□: gcr.io/google-samples/node-hello:1.0
□□□□□□:
□□ □□□: 0
□□: □□
allowPrivilegeEscalation: false
□□ □□ □□ □□ □□ □□□ □□□ □□ □ □□□ □□□□□. □□ □□□ □□□□□ □□□□ □□□. □□ □□ □□□ □□□□□ □. □□□ □□□ □□ □□□□ □□□ □□□ □□□ ID 5487□ □□□ test-user□ □□□□□□.
Answer:
□□ □□ □□□□ □□□□□.

NEW QUESTION: 11
□□□□ □□□ □□□ □□ 389□□ □□□□ □□□ □□□□□ □□□□ ID□ □□ /candidate/KH77539/files.txt □□□ □□ □□ □□ □□□ □□□ □□□□ □□□□□ □□□□□.
Answer:
□□# netstat -ltnup
□□ □□□ □□(□□□ □□)
Proto Recv-Q Send-Q □□ □□ □□ □□ □□ PID/□□□□ □□ tcp 0 0 127.0.0.1:17600 0.0.0.0:* LISTEN 1293/dropbox tcp 0 0 127.0.0.1:17603 0.0.0.0:* LISTEN 1293/dropbox tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 575/sshd tcp 0 0 127.0.0.1:9393 0.0.0.0:* LISTEN 900/perl tcp 0 0 :::80 :::* LISTEN 9583/docker- □□□ tcp 0 0 :::443 :::* LISTEN 9571/docker-proxy udp 0 0 0.0.0.0:68 0.0.0.0:* 8822/dhcpcd
...
□□# netstat -ltnup | grep ':22'
tcp 0 0 0.0.0.0:22 0.0.0.0:* 575/sshd □□
ss □□□ netstat □□□ □□□□□.
□□ ss □□□ □□□□ □□ 22□□ □□ □□ □□ □□□□□ □□□□ □□□ □□□□□□□.
□□# ss -ltnup '□□□ = :22'
Netid □□ Recv-Q Send-Q □□ □□:□□ □□ □□:□□
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* □□□:("sshd",pid=575,fd=3))

NEW QUESTION: 12
□□□□□□ □□ □□□ □□□□□□ □□ □□□□ □□□□□ □□ □□□ □□□□□.
1. □□□ /var/log/kubernetes-logs.txt□ □□□□□.
2. □□ □□□ 12□ □□ □□□□□.
3. □□ 8□□ □□ □□ □□ □□□ □□□□□.
4. □□□□ □ □□ □□□ 200MB□ □□□□□.
□□□ □□ □□□ □□□□ □□□□□.

1. RequestResponse□ □□□□□□ □□
2. kube-system □□□□□□□ □□ □□ □□□ □□ □□□ □□□□□.
3. □□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□□□.
4. □□□□□ □□□□ "pods/portforward", "services/proxy"□ □□□□□.
5. Stage RequestReceived □□
□□□□□ □□□ □□ □□ □□

Answer:

Kubernetes □□□ □□□□□ □□ □□ □□ □□□ □□□ □□□ □□□□□. Kube-apiserver□ □□□ □□□□□. □ □□ □□□ □ □ □□ □□□□ □□□□, □□□□ □□ □□□ □□ □□□□□ □□□□ □□□□□. □□□ □□□□ □□□ □□□□ □□□□ □□□ □ □□□□.

CIS(□□□ □□ □□) Kubernetes □□□□ □□ □□ □□□ □□□ □□ □□□ □□□ □ □□□□.

Cluster.yml□□ □□ □□□ □□□□ □□□□□ □□ □□□ □□□□ □ □□□□.

□□□:

□□ API:

□□_□□:

□□□□: □□

□□ □□□ □□□□□ /etc/kubernetes/audit-policy.yaml□□ □□□□ □ □ □□□ □□□. □□ □□□□ □□ □□□□ JSONlines □□ □ □□□ □□□□□. □□ kube-apiserver □□□□ □□□□ □□ □□ □□□□ □□□ □ □□□□.

--audit-log-path□ □□ □□□□ □□ □□□□ □□□□ □ □□□□ □□ □□ □□□ □□□□□. □ □□□□ □□□□ □□□ □□ □□ □□ □□□□□□□. - □□ □□□ □□□□□.

--audit-log-maxage□ □□ □□ □□ □□□ □□□ □□ □□□ □□□□□□.

--audit-log-maxbackup□ □□□ □□ □□ □□ □□ □□ □□□□□.

--audit-log-maxsize□ □□ □□ □□□ □□□□ □□ □□ □□(MB)□ □□□□□. □□□□□ □□ □□□□ kube-apiserver□ □□□ □ □□□ □□ □□ □□ □ □□ □□□ □□□ □□□ □□□□□ □□□. □□□ □□□□ □□ □□□□ □□□□□. □□ □□:

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

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

NEW QUESTION: 13

□□

kubeadm□ □□□ □□□□□ Kubernetes API □□□ □□□ □□□□ □□ □□□ Duster-admin □□□ □□□ □□□□ □□□□ □□ □ □□□ □□ □□□□ □□□□□ □□□□□ □□□□□□□.

□

□□□□ □□□ REST □□□ □□□□□ □□□□□ Kubernetes API □□□ □□□□□□.

□□ □□ Node, RBAC □ □□ □□□□ NodeRestriction□ □□□□□.

□□□□ □□□ system:anonymous□ □□ ClusterRoleBinding□ □□□□□.

All kubectl configuration contexts/files were also configured to use the unauthenticated and unauthorized access. You don't have to change that, but be aware that kubectl 's configuration will stop working, once you've completed securing the cluster.

You can use the cluster's original kubectl configuration file /etc/kubernetes/admin.conf , located on the cluster's master node, to ensure that authenticated and authorized requests are still allowed.

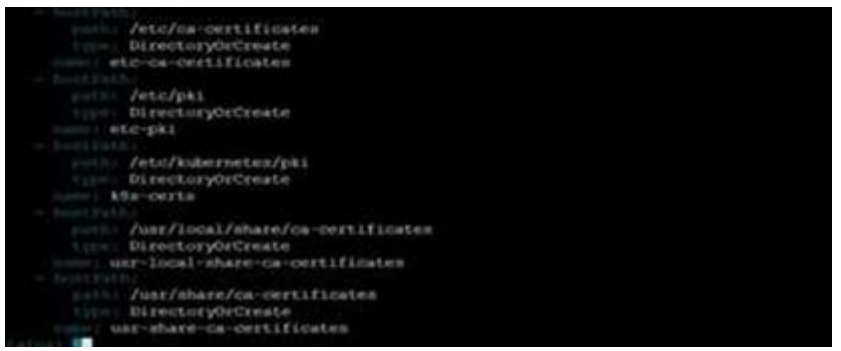
Answer:

```
candidate@cli:~$ kubectl config use-context KSCH00101
Switched to context "KSCH00101".
candidate@cli:~$ ssh ksch00101-master
Warning: Permanently added '10.240.86.190' (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@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
    - command:
        - kube-apiserver
        - --advertise-address=10.240.86.190
        - --allow-privileged=true
        - --authorization-mode=Node,RBAC
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --enable-admission-plugins=AlwaysAdmit
        - --enable-bootstrap-token-auth=true
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
        - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
        - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
"/etc/kubernetes/manifests/kube-apiserver.yaml" 128L, 4343C                1,1           Top
```

```
root@ksch00101-master:~# cat /etc/kubernetes/admin.conf
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJB
Z01CQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1ESXhO
akF3TlRVeE9Wb1hEVE15TURJeE5EQXdOVFV4T1Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdE
UVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTIvNzNZTktkSFdZU3JUaUx0QStr
N01qTXpRZllzM2ttNGl1a1poM0tZc3YlbUdpN0UyQ2tYc0MKUnhlL1lNiZnBDMzlla2k5V3hOSHc5eTM0OEtXUVE3VXBL
UmZRdXVxdlA1WXdDZkordlJmWGNGTXQxLzRNQVhWLwpkdjZ5YWRKSitPeFFSVjZlaHFBZHR0M3FtOFdVcW84UE5JT1E0
OEc3WWhnRUg5RHU3SFdkMS8raXVkSjNOMXl6CnNISEdtYklsWENSbEcydFV0M2RScDczSnRIS1JjS2tnMGxYM3FWS1Uy
QmJRblBmK01wb0VlTXFGcmZvcWVaVWcKYlBKK3ROVmZIM1JLTkhVUnYydVJIa3ZZc2JrclhUMW8rMXFNNHZrYnFNMHlq
KzNxTUtiSyt5V3dzUTlBYUVPMApUdXR4UUdlTFp3OUE3TjZZeTFVQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURB
Z0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZEcUlwLzdYbzZaNkJNVjVEK2w3bFZPcGpBOWlN
QlVHQTFVZEVRUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBSlNWNm9wNGgxYkNv
eGZLRUZ4bwoxaVlHUFlnM1hhOTN0WEZ1TTY3RnA2NkdqUEc5SXBONnNHUnRnWVlyd0MyalBDeFVOb2IySWtUQ1FNbDV3
cWRHCkdPS2JwVVp6Smc3Y0dyS2E3R1pZWVNyVUVGRWhyd2xZWXNGME56aFBoZVcwcHJjcWtSdXN1bm55SG5YNGVOMUoK
N1NzbGZYTjJIdVFJdlVIRGl5L0JsL1ZWRmZNZnRxOGF0Z0pYSFZGTmlVcDRpNXlJTXFRNTB4ZjVqcnFlWFRmVwpVdmJq
ZjEyOThXVTk3QkxHcDdRZE9QYWVKU051UStlVkMrdnpVZ2tVQVNjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3UlhLbGcvangvcitqd0dTc0xwWUxDZTlxalFraTdCSVRJTlN3ejd3c2hzbERuNzBFY0IKa0VBPQotLS0tLUVO
RCBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://10.240.86.190:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSURJVENDQWdtZ0F3SUJBZ01
ocEdQcDB4Zk9JbkYxaGJwcTh5YlBUMGxlTm5VNjBiSUpxRXVKckxJbEtXClNVa1hlVkYzNkl0ZHc1ZU1OT2JxK1haaHd
hY2JURVZCMlVDVURsbDgzdG5teFQyVXJmY0pUQmhLTCtZTFAvcWYKdjdXR3BwQ1ZXNnhVZGFibGNuUklMnpleUVJTEt
Tck5XbUQ0TzZsMU13blZ0OVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExKL3hiSDdUTzkwY1RFdlIwazl3cFVYdllkdk1jSXN
MRkYwL3F2bDA3U3lxbGplOEI1SnNpQ1hCU1ZxbS9wCmNUUSs3SnZ1bmdaZzlkOWdZaVJVdFFTcHBONkx4UnhkSzNKMGR
BK240SWxFZEtHRWh3TE00d0tMaldERG9scHgKYzB3WHkwVXBORGZ6UUxuRUFzVUJsbDRCQ3VkdW5QNVVDN2FuS3dJREF
RQUJBb01CQURWRkZNSVRqYnNySTZTTwpQOGM0MTByN3RWZ251cXJVS202dHRnZWtXOWdlS1pvMnZyb3RsbG9qOGFRamF
0MTZnaEUwOXdZd2xMSDhId0tLCklMb2NrZnFCUytlOWo1Zm1FWGxYTG00cElCVDFRbGFJQ1JRMDRyQ0JZbHdCN1VFbVB
lWjhuQ3lmR2JYTC9HM2wKcXBYTDVKdzJqcVh2MXdzcWsrdWNCRk0zZ0FYZk5YZkh1RExnV0VyNXRZRlF4VXo5UFFHOD1
pcDY1OTBkYnBlSApOMnU2NGk4UTg1dk83OFVIT1c2eUFZU1loZVdha093RDFwZzNPdkhxV3FhbnV1MnlrOWxaUUR0WW5
2MytBeU5DCnloNlRaRH1uZ01ZdEptbDFTQ01TNEpSR2d4NXNwaCtKOC9XOGx0Ri9wMWZxbTA0bXZSRndxU3M2Y1JCQ2Z
PVVcKbFV1MGxLRUNnWUVBNWJzT01VVzFBVndjTmJsc0pSVDNURkI2OVlxbDRYcnZRR0FZY3BhdktENnd5VmtEOTV1QQp
SaXVRS1NNKzY4REtBVm1pYllpaThJemExTkdqdC9JZDUwTGVoNk1aRVg2enVpK0g3d1BSbVd6SE9ueWNmU2FmClVQMEF
RL0RiM2lCNWJQTmJHYXNkaDNIb2JvR0NSSHZmTFFXY2tYbUVXM2ZudVlIRlJLZ2xlTEVDZ1lFQTVUdysKTEVTVlBESFF
mamNBN0htNmdsMndGRjdCUG1sSGdaYVVRN25Eb3ZvRmMxa1BMRWVCMWJ6OHJNWld1eGdmaHN0OQpMZ0xSUDBXdkJWd1J
sVTdMTmFLTlVzRmkxU2dvaWZsS01ZWkMyZmpLWTY1RFE3YUUzcTdnVis4U2pIZHpoclhCCkVQclAvWXQ3S0QrbFBMZmh
aNXNKZWFtelY3b3gveno5Y0s0U0ZKc0NnWUJ4OVk2VzFydHBoMFcvS05JS3V4SEoKMjRxRFQxbm1ObE9FdmFhakFUaTJ
ZQkxXYnIvWERsNWRjTEs4bFcxYkNYR3JwY2s3U0xKN1hZUVlXajQ2dTNJMgpEQ2ZUWlFiRWRQTzNBbWtPR2ZqWmdPcDd
pdUVkL0JDLzNpRkprcXVlenNFdFdMTHlVcjM5T0hZeWl6QVJ4Tmo2CnZuUGlma000Rkl6d3F2MHVoN0xlblFLQmdBT1Z
XZTRZM1RwbzJ3aEswbmVkMllsMXhVNjJoZ2JiVHcvaVdhdVcKY3ZMV3dlZUlmd0Q4MVRWL2R3a29KVEM1VEJRUXQzUkk
xRFFnVmtnMHFwcUtOOGhDNGQwM05MRzIwTWdZMk94WgpjSFZzK2J4elYwVVB6V1RUbEMyVEsyamhmOHVRcndzSktxY2N
OU0ZEczZwclhocThsV2l3Znd3aWlBR1hLSFJRCkE3RkxBb0dCQUx3NW8rbHFVZ3hHQlpKdy9EelRGek5TekQreVd6Um8
```

```
lc2ZEc2x6a2FvY0pHbEx2MUNndEVIc3QKeG5HMTlIYStSM1M3cDRtei9LeDJYMFRzaTZzUzVwWlR5WEx5STF5azh2TUZ
rR1dacjRmeVhXV2t3SjZlVEl1YwpyWFl3TWM5VFlDUGZrSFJaTm9XR1hZV3BkeTJBOXZCbFlScHZsQVZoenU2T1VZQ2w
5b2ZpCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```


path: /etc/ca-certificates
type: DirectoryOrCreate
name: etc-ca-certificates
path: /etc/pki
type: DirectoryOrCreate
name: etc-pki
path: /etc/kubernetes/pki
type: DirectoryOrCreate
name: k8s-certs
path: /usr/local/share/ca-certificates
type: DirectoryOrCreate
name: usr-local-share-ca-certificates
path: /usr/share/ca-certificates
type: DirectoryOrCreate
name: usr-share-ca-certificates

```
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@ksch00101-master:~# systemctl daemon-reload
sroot@ksch00101-master:~# systemctl restart kubelet.service
root@ksch00101-master:~# kubectl get nodes
error: You must be logged in to the server (Unauthorized)
root@ksch00101-master:~# exit
logout
Connection to 10.240.86.190 closed.
candidate@cli:~$ kubectl get nodes
NAME                STATUS   ROLES                  AGE   VERSION
ksch00101-master    Ready    control-plane,master   93d   v1.23.3
ksch00101-worker1   Ready    <none>                 93d   v1.23.3
candidate@cli:~$ kubectl get pod -n kube-system
NAME                                        READY   STATUS    RESTARTS        AGE
coredns-64897985d-7pnhm                     1/1     Running   1 (7h2m ago)    93d
coredns-64897985d-rr7sd                     1/1     Running   1 (7h2m ago)    93d
etcd-ksch00101-master                       1/1     Running   1 (7h2m ago)    93d
kube-apiserver-ksch00101-master             0/1     Running   0               24s
kube-controller-manager-ksch00101-master    1/1     Running   3 (42s ago)     93d
kube-flannel-ds-llktn                       1/1     Running   1 (93d ago)     93d
kube-flannel-ds-q9vnl                       1/1     Running   1 (93d ago)     93d
kube-proxy-2c4ht                            1/1     Running   1 (93d ago)     93d
kube-proxy-pmmbc                            1/1     Running   1 (93d ago)     93d
kube-scheduler-ksch00101-master             1/1     Running   3 (42s ago)     93d
candidate@cli:~$ kubectl get pod -n kube-system
NAME                                        READY   STATUS    RESTARTS        AGE
coredns-64897985d-7pnhm                     1/1     Running   1 (7h2m ago)    93d
coredns-64897985d-rr7sd                     1/1     Running   1 (7h2m ago)    93d
etcd-ksch00101-master                       1/1     Running   1 (7h2m ago)    93d
kube-apiserver-ksch00101-master             0/1     Running   0               30s
kube-controller-manager-ksch00101-master    1/1     Running   3 (48s ago)     93d
kube-flannel-ds-llktn                       1/1     Running   1 (93d ago)     93d
kube-flannel-ds-q9vnl                       1/1     Running   1 (93d ago)     93d
kube-proxy-2c4ht                            1/1     Running   1 (93d ago)     93d
kube-proxy-pmmbc                            1/1     Running   1 (93d ago)     93d
kube-scheduler-ksch00101-master             1/1     Running   3 (48s ago)     93d
candidate@cli:~$ kubectl get clusterrolebindings.rbac.authorization.k8s.io | grep anon
system:anonymous                                       ClusterRole/cluster-admin
                                                       7h1m
candidate@cli:~$ kubectl delete clusterrolebindings.rbac.authorization.k8s.io/system:anonymo
us
clusterrolebinding.rbac.authorization.k8s.io "system:anonymous" deleted
```


NEW QUESTION: 14
□□□□□

□□□□□□ □□□□□ Pod□ □□□ □□□□□□□ □□ □□ Pod□ □□ 80□ □□□ □ □□□ □□□□ Allow-np□□ □□□□ □□ □ □□□□.
□□□□ □□□ □□□□□□:-
1. □□ 80□□ □□ □□□□ □□ □□□ □□ □□□□ □□□□ □□□□.
2. □□□□□□ □□□□□ □□ □□□□□ □□□□ □□□□ □□□□.
Answer:
API□□:networking.k8s.io/v1
□□: NetworkPolicy
□□□□□:
□□: □□□□ □□
□□:
podSelector: {} #□□□ □□□□□□□ □□ Pod□ □□□□□.
□□ □□:
- □□□□
□□:
- □□: #in □□ □□□□ 80 □□□ □□□□ □□□□□.
- □□□□: TCP
□□: 80

NEW QUESTION: 15
□□□□□
runsc□□ □□□ □□□ □□□□ □□□□ untrusted□□ □□□ RuntimeClass□ □□□□.
gVisor □□□ □□□□□ □□□ □□□□□□ □□□□ □□□ alpine:3.13.2□ □□□ □□□□□.
□□: Pod□ □□□□ dmesg□ □□□□ □□□ □□ □□□ □□□□□.

Answer:
□□ □□ □□□□ □□□□□.

NEW QUESTION: 16
□□ □□□□/□□□□ □ □□□ □□□□ □□□.
□□□□: □□
□□□ □□: □□□
□□□ □□: □□□1

□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□.
[desk@cli] $ kubectl config □□ □□□□ □□
□□□ □□: AppArmor□ Worker1 □□□□ □□□□□□□□.
□:
Worker1 □□□□
1. /etc/apparmor.d/nginx□ □□ □□□ AppArmor □□□□ □□□□□.
2. /home/cert_masters/nginx.yaml□ □□ □□□ □□□□□ □□□ □□□□ □□ □□□□ □□□□□.
3. □ □□□□□□ □□□□ □□□ □□□□□.
Answer:
[desk@cli] $ ssh □□□1
[worker1@cli] $apparmor_parser -q /etc/apparmor.d/nginx
[worker1@cli] $aa-□□ | grep nginx
nginx-□□□−1
[worker1@cli] $ □□□□
[desk@cli] $vim nginx-deploy.yaml
□□□□□ □□□ □□ □□ □□□□□.
□□: # □ □□ □□□□□
□□□□.apparmor.security.beta.kubernetes.io/<□□□□ □□>: localhost/nginx-profile-1
[desk@cli] $kubectl apply -f nginx-deploy.yaml
□□
[desk@cli] $ ssh □□□1
[worker1@cli] $apparmor_parser -q /etc/apparmor.d/nginx
[worker1@cli] $aa-□□ | grep nginx
nginx-□□□−1
[worker1@cli] $ □□□□
[desk@cli] $vim nginx-deploy.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-deploy
  annotations:
    container.apparmor.security.beta.kubernetes.io/hello: localhost/nginx-profile-1  # Add this line
spec:
  containers:
  - name: hello
    image: nginx
```

[desk@cli] $kubectl apply -f nginx-deploy.yaml pod/nginx-deploy □□□ □□: https://kubernetes.io/docs/tutorials/clusters/apparmor/
pod/nginx-deploy □□□
[desk@cli] $kubectl apply -f nginx-deploy.yaml pod/nginx-deploy □□□ □□: https://kubernetes.io/docs/tutorials/clusters/apparmor/

NEW QUESTION: 17
□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□. [desk@cli] $ kubectl config use-context qa □□□□: □□ □□□

ServiceAccount□ □□ □□□ □□□□ □□□□. □□: backend-qa□□ □ □□□ □□□ □□□□□. □□ □□□□ □□□□ □ □□ □ □ □□□□□□ qa. backend-qa □□□ □□□ □□□□□ □□□□□ □□ yaml□ □□□□□. □□: /home/cert_masters/frontend-pod.yaml□□ □□□□□ □□ yaml□ □□ □ □□□□.

Answer:

[desk@cli] $ k create sa backend-qa -n qa sa/backend-qa Created [desk@cli] $ k get role,rolebind -n qa qa □□□□□□□□ □□□□ □□ □ □□□□. [desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb list # □□□ □□ □□□ □□ □□ [desk@cli] $ k create rolebind backend -n qa --role backend --serviceaccount qa:backend-qa [desk@cli] $ vim /home/cert_masters/frontend-pod.yaml apiVersion: v1 □□: □□ □□□□□:

□□: □□□□□

□□:

serviceAccountName: backend-qa # □□□ □□□□□

□□□: nginx

□□: □□□□□

[desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yaml □□ □□□

[desk@cli] $ k create sa backend-qa -n qa serviceaccount/backend-qa Created [desk@cli] $ k get role,rolebind -n qa qa □□□□□□□□ □□□□ □□ □ □□□□. [desk@cli] $ k □□ □□□ □□ -n qa --resource pods,namespaces,configmaps --□□ □□ role.rbac.authorization.k8s.io/backend □□□ [desk@cli] $ k □□ □□□ □□□ □□ -n qa --role backend --serviceaccount qa:backend-qa rolebbing.rbac.authorization.k8s.io/backend Created [desk@cli] $ vim /home/cert_masters/frontend-pod.yaml apiVersion: v1 □□: □□ □□ □□□:

□□: □□□□□

□□:

serviceAccountName: backend-qa # □□□ □□□□□

□□□: nginx

□□: □□□□□

[desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yaml pod/frontend □□□ https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

NEW QUESTION: 18

□□□□□

□□ □□□□□□ □□□□ backend-sa□□ □ ServiceAccount□ □□□□□. □ ServiceAccount□ □□□□□□ □□□ □□□ Pod□ □ □□□ □□□ □□□□.

□□□□□□ □□□□ backend-pod□□ □ □□□ □□□□ □□ □□□ sa backend-sa□ □□□ □□□□ □ □□□ □□□ □□□ □ □ □□ □□□□□.

□□□ □□ □□□ □□□□□.

Answer:

□□□ □□□ Pod□□ □□□□ □□□□□ □□ ID□ □□□□□.

□□□(□□)□ □□□□□ □□□□□(□: kubectl □□) apiserver□ □□ □□ □□□ □□□□ □□□□□(□□□ □□□□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□□□ □□□ □□□□□). Pod □□ □□□□□ □□□□□ apiserver□ □□□ □ □□□□. □□□ □□ □□ □□□ □□(□: □□□)□□ □□□□□.

Pod□ □□□ □ □□□ □□□ □□□□ □□□ □□□ □□□□□□□ □□ □□□ □□□ □□□□ □□□□□. □□□ Pod□ □□ □□ json □□ yaml□ □□□□(□: kubectl get pods/<podname> -o yaml) spec.serviceAccountName □□□ □□□□ □□□ □□ □ □ □□□

□.
□□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□ □□ □□□ □□□□ □□ □□□□ API□ □□□□ □ □□□□. □□□ □□ □ API □□□ □□ □□ □□ □□□□ □ □□□ □□ □□□□.
□□ 1.6 □□□□□ □□□ □□□ automountServiceAccountToken: false□ □□□□ □□□ □□□ □□ API □□ □□ □□ □□□□ □□ □□□ □ □□□□.
api□□: v1
□□: □□□ □□
□□□□□:
□□: □□ □□
automountServiceAccountToken: □□
...
□□ 1. 6+□□□ □□ □□□ □□ API □□ □□ □□ □□□□ □□ □□□ □□ □□□□.
api□□: v1
□□: □□
□□□□□:
□□: □ □□
□□:
serviceAccountName: □□ □□
automountServiceAccountToken: □□
...
□ □ automountServiceAccountToken □□ □□□□ □□ □□ □□□ □□□ □□□□ □□□□□.

NEW QUESTION: 19
runsc□□ □□□ □□□ □□□□ □□□□ untrusted□□ □□□ RuntimeClass□ □□□□.
gVisor □□□ □□□□□ □□□ □□□□□□ □□□□ □□□ alpine:3.13.2□ □□□ □□□□□.
Answer:
□□: Pod□ □□□□ dmesg□ □□□□ □□□ □□ □□□ □□□□□.

NEW QUESTION: 20
□□ □□□□/□□□□ □ □□□ □□□□ □□□. □□□□: □□ □□□ □□: □□□ □□□ □□: □□□1 □□ □□□ □□□□ □□□ □/□□ □□□□□ □□□ □ □□□□: [desk@cli] $ kubectl config use-context Trace □□□: Sysdig □□ Falco □□□ □□□ □ □□□□. □□: □□ □□□ □□□□ Pod tomcat□ □□ □□ □□□□□□ □□□ □□ □□ □□□□ □□□□ □□□□□ □□ □□ □□□ □□□ □□. □ □□ □□□ □□□ □ □□□□. 1. falco 2. sysdig □□□ □□□1 □□□□ □□ □□□□□. □□ □□□□ □□□□ □□□□□ □□□□ □□□ □□□□ □□ 40□ □□ □□□□□ □□□ □□□□□. □□□□ □□□ /home/cert_masters/report□ □□ □□□□ □□ □□□: [timestamp],[uid],[processName] □□: □□□□ □□□ □□□□□ □□□ □□□ □□□□ □□□ □□□ □□□□ □□□□. .

Answer:
$vim /etc/falco/falco_rules.local.yaml
- □□: □□□□ □□□□ □□(□□+□□)
desc: open+create□ □□ □□□□□ □□□ □ □□ □□
□□: >
evt.type (open,openat,creat) □
evt.is_open_exec=true □
□□□□□
runc_writing_exec_fifo□ □□□
runc_writing_var_lib_docker□ □□□
user_known_container_drift_activities□ □□□
evt.rawres>=0
□□: >
%evt.time,%user.uid,%proc.name # □□□ □□□□□/falco □□□ □□□□□
□□□□: □□
$kill -1 <Falco□ PID>
□□
[desk@cli] $ ssh node01 [node01@cli] $ vim /etc/falco/falco_rules.yaml □□□□ □□□□ □□□ □□□□ falco_rules.local.yaml□ □□□ □□□. [node01@cli] $ vim /etc/falco/falco_rules. local.yaml
- □□: □□□□ □□□□ □□(□□+□□)
desc: open+create□ □□ □□□□□ □□□ □ □□ □□
□□: >
evt.type (open,openat,creat) □
evt.is_open_exec=true □
□□□□□
runc_writing_exec_fifo□ □□□
runc_writing_var_lib_docker□ □□□
user_known_container_drift_activities□ □□□
evt.rawres>=0
□□: >
%evt.time,%user.uid,%proc.name # □□□ □□□□□/falco □□□ □□□□□
□□□□: □□
[node01@cli] $ vim /etc/falco/falco.yaml

NEW QUESTION: 21

□□:
□□□□: gvisor
□□□ □□: master1
□□□ □□: □□□1
□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□.
[desk@cli] $ kubectl config use-context gvisor
□□□□: □ □□□□□ □□□ □□□, runsc □ □□ □□□□ □□□□□ □□□□□□□.
□:
□□□ □□□ □□□ □□□ runsc□ □□□□ not-trusted□□ RuntimeClass□ □□□□.
newruntime□□ □□□□□ □□□□□□ □□□ □□ □□□ □□□□□□□.
Answer:
□□ □□/□□□ □□ □□□ClassName □□□□□ □□□ □□ □□□ □ □□□□ □□□□□.
[desk@cli] $ k edit nginx □□
□□:
□□□□□□ □□: □□□ □ □□□□. # □□ □□ □□
□□
[desk@cli] $vim □□□.yaml
API□□: node.k8s.io/v1
□□: RuntimeClass
□□□□□:
□□: □□□ □ □□
□□□: runsc
[desk@cli] $ k □□ -f □□□.yaml
[desk@cli] $ k □□ □□□□
□□ □□ □□ □□□ □□
nginx-6798fc88e8-chp6r 1/1 □□ □ 0 11m
nginx-6798fc88e8-fs53n 1/1 □□ □ 0 11m
nginx-6798fc88e8-ndved 1/1 □□ □ 0 11m
[desk@cli] $ k □□ □□
□□ □□□ □□ □□ □□ □□
nginx 3/3 11 3 5m
[desk@cli] $ k edit nginx □□

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      labels:
        app: nginx
    spec:
      runtimeClassName: not-trusted     # Add this
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
```


NEW QUESTION: 22
□□□□ □□□ □□□□ □□□ AppArmor □□□□ □□□□□.
#include <□□□/□□>
□□□ docker-nginx □□□=(attach_disconnected,mediate_deleted) {
#include <□□□/□□□>
□□□□ INET TCP,
□□□□ INET UDP,
□□□□ Inet ICMP,
□□□□ □□ □□,
□□□□ □□ □□,
□□,
□□□ □□,
/bin/** wl □□,
/boot/** wl □□,
/dev/** wl □□,
/etc/** wl □□,
/home/** wl □□,

/lib/** wl □□,
/lib64/** wl □□,
/media/** wl □□,
/mnt/** wl □□,
□□ /opt/** wl,
/proc/** wl □□,
/root/** wl □□,
/sbin/** wl □□,
/srv/** wl □□,
/tmp/** wl □□,
/sys/** wl □□,
/usr/** wl □□,
□□ /** w,
/var/run/nginx.pid w,
/usr/sbin/nginx ix,
/bin/dash mrwklx □□,
/bin/sh mrwklx □□,
/usr/bin/top mrwklx □□,
□□ chown,
dac_override □□,
□□ setuid,
□□ □□,
net_bind_service □□,
□□ @{PROC}/* w, # /proc(□□ □□□□ □□)□□ □□ □□ □□□ □□ □□ □□
/proc/<number>/** □□ /proc/sys/**□ □□ □□□ □□ □□□ □□□□□.
@{PROC}/{[^1-9],[^1-9][^0-9],[^1-9s][^0-9y][^0-9s],[^1- □□ 9][^0-9][^0-9][^0-9]*}/** w, @{PROC}/sys/[^k]** w, # □□ /proc/sys □
□ /proc/sys/k* (□□□□□□ /proc/sys/kernel) @{PROC}/sys/kernel/{?,??,[^s][^h][^m]**} □□ w, # /proc/sys/kernel/□□ shm*□ □□□ □□
□□ □□ @{PROC}/sysrq-trigger rwklx □□, @{PROC}/mem rwklx □□, @{PROC}/kmem rwklx □□, @{PROC}/kcore rwklx □□ , □□□
□□, /sys/[^f]*/** wklx □□, /sys/f[^s]*/** wklx □□, /sys/fs/[^c]*/** wklx □□, □□ /sys/fs/c[^g]*/** wklx, /sys/fs/cg[^r]*/** wklx □
□, /sys/firmware/** rwklx □□, /sys/kernel/security □□ /** rwklx,
}
AppArmor □□□□ □□□□□ □□□ □□□□□ □□□ □□□□□.
api□□: v1
□□: □□
□□□□□:
□□: apparmor-pod
□□:
□□□□:
- □□: apparmor-pod
□□□: nginx
□□□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□ □□□□□.

□□: ping, top, sh □□□ □□□ □□□□.
Answer:
□□ □□ □□□□ □□□□□.

NEW QUESTION: 23
□□□□□
□□□□□□ □□□ □□ □□ □□ Nginx-pod□ □□□□, □□□ □□□ □□□□ nginx-svc□□ Nginx-pod□ □□ □□□□ □□□□, □ □ □□□ tls□□ □□□ □□□□□.
Answer:
□□□ □□□

NEW QUESTION: 24
etcd□ □□□ □□□ □□ □ □□□□ □□□□. etcdctl □□ □□□□□ □□□□ □□□ □□ □□ □□ □□ □ □□□□.
ETCDCTL_API=3 etcdctl get /registry/secrets/default/cks-secret --cacert="ca.crt " --cert="server.crt" --key="server.key" □□

□□□ □□□ □□□□ □□□ AES-CBC □ ID□ □□□□ □□□ □□□ □□□□ □□□□□□ □□□□ □□ □□ □□ □□□□ □□□ □□ □□ □□□ □ □□□□ □□□□□□ □□□.
Answer:
ETCD □□ □□□□ etcdctl □□□ □□□□□ □□□□ □□□ □ □□□□.
ETCD □□□ □□□ □□□ /registry/secrets/$namespace/$secret □□□ □□□□□.
□□ □□□ □□□□ □□ ETCD □□□ □□□□□□□ □□□ □□□ □ □□□□.
ETCDCTL_API=3 etcdctl get /registry/secrets/default/secret1 [...] | 16 □ □□ -C

NEW QUESTION: 25
□□□□□□□□ □□ □□ □□ □□□ □□□□ PSP□ □□□□□.
□□□ □□ Pod □□□ □□□□ Prevent-privileged-policy□□ □ PodSecurityPolicy□ □□□□□.
□□□□□□ □□□□ psp-sa□□ □ ServiceAccount□ □□□□.
□□ □□□ □□ □□ □□□ □□ □□ □□□ □□□□ □□ □□□□□ □ ClusterRole□ □□□□□.
□□□ ClusterRole □□ □□□ □□□ SA psp-sa□ □□□□□ □□ □□ □□□□□□ □ ClusterRoleBinding□ □□□□□.
□□ □□ □□ □□ □□□ □□□□ □□□ □□□□□ □□□□ □□□□ □□□.
Answer:
□□□□□□□□ □□ □□ □□ □□□ □□□□ PSP□ □□□□□.
$ □□□ □□□□role-use-privileged.yaml
---
api□□: rbac.authorization.k8s.io/v1

□□: ClusterRole
□□□□□:
□□: use-privileged-psp
□□:
- apiGroups: ['□□']
□□□: ['podsecuritypolicies']
□□: ['□□□□']
□□ □□:
- □□-psp
---
api□□: rbac.authorization.k8s.io/v1
□□: RoleBinding
□□□□□:
□□: □□ □□ □□□
□□□□□□: psp-test
□□□□:
API□□: rbac.authorization.k8s.io
□□: ClusterRole
□□: use-privileged-psp
□□:
- □□: □□□ □□
□□: □□-sa
$ kubectl -n psp-test □□ -f Clusterrole-use-privileged.yaml
□□ □ □□□ □□ □□□ □□□□□ □□□.
□□□ □□ Pod □□□ □□□□ Prevent-privileged-policy□□ □ PodSecurityPolicy□ □□□□□.
api□□: □□/v1beta1
□□: PodSecurityPolicy
□□□□□:
□□: □
□□:
□□: false # □□ □□□ □□□□ □□□!
□□□□ □□ □□ □□□ □□□□.
seLinux:
□□: RunAsAny
□□ □□:
□□: RunAsAny
□□□ □□:
□□: RunAsAny
fs□□:
□□: RunAsAny
□□:

- '*'
□□□ kubectl□ □□□□ □□□□□.
kubectl-admin create -f example-psp.yaml
□□ □□□ □□ □□□□□ □□□ Pod□ □□□ □□□□.
kubectl-user create -f- <<EOF
api□□: v1
□□: □□
□□□□□:
□□: □□□□
□□:
□□□□:
- □□ : □□□□
□□□: k8s.gcr.io/pause
EOF
□□□ □□□ □□□□□.
□□ □□(□□□): "STDIN" □□ □ □□ □□: □□ "□□ □□"□ □□□□□□□. □□ □□ □□□ □□ □□□□ □□□ □ □□□□. [] □
□□□□□ □□□□ psp-sa□□ □ ServiceAccount□ □□□□□.
$ □□□ □□□□role-use-privileged.yaml
---
api□□: rbac.authorization.k8s.io/v1
□□: ClusterRole
□□□□□:
□□: use-privileged-psp
□□:
- apiGroups: ['□□']
□□□: ['podsecuritypolicies']
□□: ['□□□□']
□□ □□:
- □□-psp
---
api□□: rbac.authorization.k8s.io/v1
□□: RoleBinding
□□□□□:
□□: □□ □□ □□□
□□□□□□: psp-test
□□□□:
API□□: rbac.authorization.k8s.io
□□: ClusterRole
□□: use-privileged-psp
□□:
- □□: □□□ □□

□□: □□-sa
$ kubectl -n psp-test □□ -f Clusterrole-use-privileged.yaml
□□ □ □□□ □□ □□□ □□□□□ □□□.
□□ □□□ □□ □□ □□□ □□ □□ □□□ □□□□ □□ □□□□□ □ ClusterRole□ □□□□□.
api□□: □□/v1beta1
□□: PodSecurityPolicy
□□□□□:
□□: □
□□:
□□: false # □□ □□□ □□□□ □□□!
□□□□ □□ □□ □□□ □□□□.
seLinux:
□□: RunAsAny
□□ □□:
□□: RunAsAny
□□□ □□:
□□: RunAsAny
fs□□:
□□: RunAsAny
□□:
- '*'
□□□ kubectl□ □□□□ □□□□□.
kubectl-admin create -f example-psp.yaml
□□ □□□ □□ □□□□□ □□□ Pod□ □□□ □□□□.
kubectl-user create -f- <<EOF
api□□: v1
□□: □□
□□□□□:
□□: □□□□
□□:
□□□□:
- □□ : □□□□
□□□: k8s.gcr.io/pause
EOF
□□□ □□□ □□□□□.
□□ □□(□□□): "STDIN" □□ □ □□ □□: □□ "□□ □□"□ □□□□□□□. □□ □□ □□□ □□ □□□ □ □□□□. [] □□□
ClusterRole □□□ □□□□□ □□ □□ □□□□□□ □ ClusterRoleBinding□ □□□□□. □□□ SA psp-sa□ □□ □□□□□.
api□□: rbac.authorization.k8s.io/v1
□ □□ □□□□ □□ "jane"□ "default" □□□□□□□ Pod□ □□ □ □□□□.
□□ □□□□□□□ "pod-reader"□□ □□□ □□ □□□ □□□.
□□: RoleBinding

□□□□□:
□□: □□ □□
□□□□□□: □□□
□□:
□□ □□□ "□□"□ □□□ □ □□□□
- □□ : □□□
□□: jane # "name"□ □□□□□ □□□□□.
API□□: rbac.authorization.k8s.io
□□□□:
"roleRef"□ □□/ClusterRole□ □□ □□□□ □□□□□.
□□: □□ #□□□ □□ □□ ClusterRole□□□ □□□.
□□: pod-reader # □□ apiGroup: rbac.authorization.k8s.io□ □□□□□□ □□ □□ ClusterRole□ □□□ □□□□ □□□ apiVersion:
rbac.authorization.k8s.io/v1 □□: □□ □□□□□:
□□□□□□: □□□
□□: □□ □□
□□:
- apiGroups: [""] # ""□ □□ API □□□ □□□□□.
□□□: ["□□"]
□□: ["get", "watch", "list"]

NEW QUESTION: 26
□□□□□
□□□ □□ □□ Falco□ □□□□ □□ □□ □ □□ □□□□□ □□□□ □□□ □□□□ □□ 30□ □□ □□□□ □□□ □□□□ □□
□ □□□ □□□ □□ □□ □□ /opt/falco-incident.txt□ □□□□□. □ □□ □□□ □□□□
[□□□□□],[uid],[□□□ □□],[□□□□ □□]
Answer:
□□□ □□□□□.

NEW QUESTION: 27
test-system □□□□□□□□ □□ □□ test-web-pod□□ □□ □□□ □□ □□ □□□□□□□ □□□□ □□ □□□ □□□□□ sa-backend□□ □□□ □□□ □□□ □□□□ □□ □□□ □□□□□.
Statefulsets □□□ □□□□ □□ □□ □□□ □□□ □ □□ test-system □□□□□□□ test-system-role-2□□ □ □□□ □□□□□.
Answer:
□□ □□□ □□□ □□□ ServiceAccount sa-backend□ □□□□□ test-system-role-2-binding□□□ □ RoleBinding□ □□□□□.

NEW QUESTION: 28
□□□□□□ □□□□□ □□ □□ Pod nginx-test□ □□□□□ □□np□□ □□□□ □□□ □□□□□.
□□ Pod□ Pod nginx-test□ □□□□□ □□□□□.
1. □□□□□□ □□□□ □□
2. □□ □□□□□□□ version:v1 □□□ □□ Pod.
□□□□ □□□ □□□ □□□□□.

Answer:
□□ □□ □□□□ □□□□□.

NEW QUESTION: 29
□□□□: qa-cluster □□□ □□: master □□□ □□: Worker1 □□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□: [desk@cli] $ kubectl config use-context qa-cluster □□: received-policy□□ NetworkPolicy□ □□□□□. □□□□□□ dev□□ □□ □□ Pod □□□ □□ □□□□ □□□□□. □□ Pod□ Pod products-service□ □□□□□ □□□□□. 1. □□□□□□ qa□ Pod 2. □□□□ □□ Pod: □□ □□□□□□□ stage: stage
Answer:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
```

```
candidate@cli:~$ vim np.yaml
candidate@cli:~$ cat np.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl create -f np.yaml -n dev-team
```

```
networkpolicy.networking.k8s.io/pod-access created
candidate@cli:~$ kubectl describe netpol -n dev-team
Name:         pod-access
Namespace:    dev-team
Created on:   2022-05-20 15:35:33 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  PodSelector:     environment=dev
  Allowing ingress traffic:
    To Port: <any> (traffic allowed to all ports)
    From:
      NamespaceSelector: environment=dev
    From:
      PodSelector: environment=testing
  Not affecting egress traffic
  Policy Types: Ingress
candidate@cli:~$ cat KSSH00301/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ""
  namespace: ""
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from: []
    - from: []
candidate@cli:~$ cp np.yaml KSSH00301/network-policy.yaml
candidate@cli:~$ cat KSSH00301/network-policy.yaml
```

```
candidate@cli:~$ cat KSSH00301/network-policy.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: pod-access
  namespace: dev-team
spec:
  podSelector:
    matchLabels:
      environment: dev
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              environment: dev
        - podSelector:
            matchLabels:
              environment: testing
candidate@cli:~$
```


NEW QUESTION: 30
□□□□□
□□□□ □□□ □□□□ □□□ AppArmor □□□□ □□□□□.
#include <□□□/□□>
□□□ nginx-deny □□□=(attach_disconnected) {
#include <□□□/□□□>
□□,
□□ □□ □□□ □□□□□.
□□ /** w,
}
EOF'
AppArmor □□□□ □□□□□ □□□ □□□□□ □□□ □□□□□.
api□□: v1
□□: □□
□□□□□:
□□: apparmor-pod
□□:
□□□□:

- □□: apparmor-pod
□□□: nginx
□□□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□ □□□□□.
□□: □□□ □□□□ □□ □□□ □□□ □□□□.

Answer:

□□□ □□□□□.

NEW QUESTION: 31

□□□□□ □□□ □□□□ □ □□ □□□ □□ □□□□ □□□□.
□□□ □□(□□ □□□□ □□ □□ □□)□ □□□□ □□ □□ □ □□□□ □□□□□ □□□□ □□□ □□□□ □□□□□ □□□ □ □ 30□ □□ □□□□□.
□□□ □□□ □ □□ □□□ □□□□ □□ □□□ /opt/KSRS00101/alerts/details□ □□ □□□□ □□□□□.

```
timestamp,uid/username,proce
ssName
```

□□ □□ □□□ □□□ □□ □□□ □□□□□.

```
01:40:19.601363716,root,init
01:40:20.606013716,nobody,ba
sh
01:40:21.137168716,1000,tar
```

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

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

Answer:

```
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
```

```
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/sys
tem/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 11:39:48 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
```

```
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:~$
```


NEW QUESTION: 32
□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□. [desk@cli] $ kubectl config use-context prod-account □□□□: □□□ ServiceAccount□ □□□□ □□□ □□□ □□□ □□□□□. □□ □□□ □□□□ □□ □□□ □□□□□. □□: □□□□□□ □□□□ □□□□ □□ □□ web-pod□□ □□ □□□ □□□□□. 1. Pod □□□ □□□□□□ □□□□ □□ □□□ □□□□□ Pod□ ServiceAccount test-sa□ □□□□ □□ □□□ □□□□□. 2. □□□□□□ □□□□□□□ test-role-2□□ □ □□□ □□□□□. □ □□ □ Statuefulsets □□□ □□□□ □□□□ □□□□ □□ □□□ □□□□□. 3. □□ □□□ □□□ □□□ ServiceAccount□ □□□□□ test-role-2-bind□□ □ RoleBinding□ □□□□□. □□: □□ RoleBinding□ □□□□ □□□.
Answer:

```
candidate@cli:~$ kubectl config use-context KSCH00201
Switched to context "KSCH00201".
candidate@cli:~$ kubectl get pods -n security
NAME      READY   STATUS    RESTARTS   AGE
web-pod   1/1     Running   0          6h9m
candidate@cli:~$ kubectl get deployments.apps -n security
No resources found in security namespace.
candidate@cli:~$ kubectl describe rolebindings.rbac.authorization.k8s.io -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  Role
  Name:  dev-role
Subjects:
  Kind            Name      Namespace
  ----            ----      ---------
  ServiceAccount  sa-dev-1
candidate@cli:~$ kubectl describe role dev-role -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  *          []                 []              [*]
candidate@cli:~$ kubectl edit role/dev-role -n security
```

```
  uid: b4c9ddd6-2729-43bd-8fbd-b2d227f4c4cd
rules:
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - watch
```

NEW QUESTION: 33
Trivy□ □□□□ □□ □□□□ □□□□□.
Answer:
1. amazonlinux:1
2. k8s.gcr.io/kube-controller-manager:v1.18.6
HIGH □□ CRITICAL □□□ □□□□ □□ □□□□ □□ □□□ □□□ /opt/trivy-vulnerable.txt□ □□□□□.

NEW QUESTION: 34
□□ □□□□□□ □□□□ backend-sa□□ □ ServiceAccount□ □□□□□. □ ServiceAccount□ □□□□□□ □□□ □□□ Pod□ □

□□□ □□□ □□□□.
□□□□□□ □□□□ backend-pod□□ □ □□□ □□□□ □□ □□□ sa backend-sa□ □□□ □□□□ □ □□□ □□□ □□□ □ □ □□ □□□□□.
□□□ □□ □□□ □□□□□.
Answer:
□□□ □□□ Pod□□ □□□□ □□□□□ □□ ID□ □□□□□.
□□□(□□)□ □□□□□ □□□□□(□: kubectl □□) apiserver□ □□ □□ □□□ □□□□ □□□□□(□□□ □□□□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□□□ □□□ □□□□□). Pod □□ □□□□□ □□□□□ apiserver□ □□□ □ □□□□. □□□ □□ □□ □□□ □□(□: □□□)□□ □□□□□.
Pod□ □□□ □ □□□ □□□ □□□□ □□□ □□□ □□□□□□□ □□ □□□ □□□ □□□□ □□□□□. □□□ Pod□ □□ □□ json □□ yaml□ □□□□(□: kubectl get pods/<podname> -o yaml) spec.serviceAccountName □□□ □□□□ □□□ □□ □ □ □□□ □.
□□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□ □□ □□□ □□□□ □□ □□□□ API□ □□□□ □ □□□□. □□□ □□ □ API □□□ □□ □□ □□ □□□□ □ □□□ □□ □□□□.
□□ 1.6 □□□□□ □□□ □□□ automountServiceAccountToken: false□ □□□□ □□□ □□□ □□ API □□ □□ □□ □□□□ □□ □□□ □ □□□□.
api□□: v1
□□: □□□ □□
□□□□□:
□□: □□ □□
automountServiceAccountToken: □□
...
□□ 1. 6+□□□ □□ □□□ □□ API □□ □□ □□ □□□□ □□ □□□ □□ □□□□.
api□□: v1
□□: □□
□□□□□:
□□: □ □□
□□:
serviceAccountName: □□ □□
automountServiceAccountToken: □□
...
□ □ automountServiceAccountToken □□ □□□□ □□ □□ □□□ □□□ □□□□ □□□□□.

NEW QUESTION: 35
□□□□ □□□ □□□□ □□□ AppArmor □□□□ □□□□□.
#include <□□□/□□>
□□□ nginx-deny □□□=(attach_disconnected) {
#include <□□□/□□□>
□□,
□□ □□ □□□ □□□□□.
□□ /** w,

}
EOF'
Answer:
AppArmor □□□□ □□□□□ □□□ □□□□□ □□□ □□□□□.
api□□: v1
□□: □□
□□□□□:
□□: apparmor-pod
□□:
□□□□:
- □□: apparmor-pod
□□□: nginx
□□□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□ □□□□□.
□□: □□□ □□□□ □□ □□□ □□□ □□□□.

NEW QUESTION: 36
□□ □□□ □□□□ □□□□/□□ □□□□□ □□□ □ □□□□: [desk@cli] $ kubectl config use-context test-account □□: □□□□□ □ □□ □□□ □□□□□□.
□□□ □□□ □□ □□□□ □□□□□ □□□ □□□□□.
1. □□□ /var/log/Kubernetes/logs.txt□ □□□□□.
2. □□ □□□ 5□ □□ □□□□□.
3. □□ 10□□ □□ □□ □□ □□□ □□□□□.
□□ □□□ /etc/Kubernetes/logpolicy/audit-policy.yaml□ □□□□□. □□□□ □□ □□□ □□□□□. □□: □□ □□□ □□□□□ □□ □ □□□ □□□□.
□□□ □□ □□□ □□□□ □□□□□. 1. RequestResponse □□□□ □□ □□ 2. □□□□□□ □□□□□□□ □□ □□□ □□ □□ □□ 3. □□□□□ □□□□ □□ □□□□□□□ ConfigMap □ Secret □□ □□ □□□ □□ □□□ □□□□□. □□□□□ □□□□ □ □ □□ □□□ □□□□□. □□: □□□ □□□ □□□□ □□ □□ □□□□.
Answer:
$ vim /etc/kubernetes/log-policy/audit-policy.yaml
- □□: RequestResponse
userGroups: ["□□□:□□"]
- □□ : □□
□□:
- □□: "" # □□ API □□
□□: ["□□ □□"]
□□□□□□: ["□□□□□"]
- □□: □□□□□
□□:
- □□: ""
□□□: ["configmaps", "secrets"]
- □□: □□□□□

$ vim /etc/kubernetes/manifests/kube-apiserver.yaml □□ □□□□□
- --audit-policy-file=/etc/kubernetes/log-policy/audit-policy.yaml
- --audit-log-path=/var/log/kubernetes/logs.txt
- --audit-log-maxage=5
- --audit-log-maxbackup=10
□□
[desk@cli] $ ssh master1 [master1@cli] $ vim /etc/kubernetes/log-policy/audit-policy.yaml apiVersion: audit.k8s.io/v1 # □□□□□.
□□: □□
RequestReceived □□□ □□ □□□ □□ □□ □□□□ □□□□ □□□.
□□□□:
- "□□ □□□"
□□:
□□□□□□ □□□□□ "system:kube-proxy"□ □□ □□ □□□ □□□□ □□□.
- □□: □□
□□□: ["system:kube-proxy"]
□□: ["□□"]
□□:
- □□: "" # □□ API □□
□□□: ["□□□□□", "□□□"]
□□□□ □□ □□ URL □□□ □□ □□□ □□□ □□□□ □□□.
- □□: □□
userGroups: ["□□□:□□□"]
□□□URL:
- "/api*" # □□□□□ □□.
- "/□□"
□□□ □□□□□ □□□□□
- □□: RequestResponse
userGroups: ["system:nodes"] # □□ □□
- □□ : □□
□□:
- □□: "" # □□ API □□
resources: ["perciousvolumes"] # □□ □□ □□
□□□□□□: ["frontend"] # □□□□□ ns□ □□ □□□ □□□□□.
- □□: □□□□□
□□:
- □□: "" # □□ API □□
resources: ["configmaps", "secrets"] # configmap □ □□ □□
- □□: □□□□□ # □□ □□ □□ □□
[master1@cli] $ vim /etc/kubernetes/manifests/kube-apiserver.yaml
api□□: v1
□□: □□

□□□□□:
□□:
kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.0.0.5:6443 □□:
□□□□: kube-apiserver
□□: □□ □□
□□: kube-apiserver
□□□□□□: kube-system
□□:
□□□□:
- □□:
-kube-api□□
- --advertise-address=10.0.0.5
- --allow-privileged=true
- --authorization-mode=□□,RBAC
- --audit-policy-file=/etc/kubernetes/log-policy/audit-policy.yaml #□ □□□ □□□□□.
- --audit-log-path=/var/log/kubernetes/logs.txt # □□□ □□□□□
- --audit-log-maxage=5 #□□□ □□□□□
- --audit-log-maxbackup=10 #□□□ □□□□□
...
□□ □□
□□: □□ □□ □ □□ □□□ vim /etc/kubernetes/manifests/kube-apiserver.yaml□ □□ □□□□□ □□□□ □□□□ □□□ □□□□.
□□: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

NEW QUESTION: 37
□□ □□□□/□□□□ □ □□□ □□□□ □□□. □□□□: immutable-cluster □□□ □□: master1 □□□ □□: □□□1 □□ □□□ □ □□□ □□□□/□□ □□□□□ □□□ □ □□□□: [desk@cli] $ kubectl config use-context immutable -□□□□ □□□□: □□□□□ □ □ □□□ □ □□□□ □□□□ □□ □□ □□□□□. □□: prod □□□□□□□□ □□ □□ □□□ □□□□ □□ □□□□ □□□□ □ □□ □ □□ □□□ □□□□□. □□□ □ □□□ □□ □□□ □□ □□□ □□□ □□□□□□. 1. □□□□ □□□ □□□□ □□□ □ □ □ □□□ □□□□ □□ □□□ □□□□□ □□□. □□: □□□□ □□□ □□□□ □□□ □□□□ □□□ □□□ □□□ □□□ □□□ □. 2. □□ □□□□□ □□□ □□□□□ □□□ □□□ □□□□□ □□ □□□ □□ □□□ □□ □□□ □□□□□ □□□.
Answer:

```
witched to context "KSRS00501".
andidate@cli:~$ kubectl get pod -n testing
AME       READY   STATUS    RESTARTS   AGE
pp        1/1     Running   0          6h31m
rontend   1/1     Running   0          6h32m
mtp       1/1     Running   0          6h31m
andidate@cli:~$ kubectl get pod/app -n testing -o yaml
  - lastProbeTime: null
    lastTransitionTime: "2022-05-20T08:40:35Z"
    status: "True"
    type: PodScheduled
  containerStatuses:
  - containerID: docker://11143682c400984c9faf3dff1e056d4b00a7eb1de007fe1834be0a84fa146e18
    image: nginx:latest
    imageID: docker-pullable://nginx@sha256:2d17cc4981bf1e22a87ef3b3dd20fbb72c3868738e3f3076
2eb40e2630d4320
    lastState: {}
    name: app-container
    ready: true
    restartCount: 0
    started: true
    state:
      running:
        startedAt: "2022-05-20T08:40:37Z"
  hostIP: 10.240.86.141
  phase: Running
  podIP: 10.10.1.3
  podIPs:
  - ip: 10.10.1.3
  qosClass: BestEffort
  startTime: "2022-05-20T08:40:35Z"
andidate@cli:~$ kubectl get pod/app -n testing -o yaml | grep -E 'privileged|ReadOnlyFileSy
tem'
      privileged: true
andidate@cli:~$ kubectl get pod/frontend -n testing -o yaml | grep -E 'privileged|ReadOnlyF
leSystem'
```

```
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged|ReadOnlyFileS
ystem'
      privileged: true
candidate@cli:~$ kubectl get pod -n testing -o yaml | grep -i ReadOnly
        readOnlyRootFilesystem: false
        readOnly: true
        readOnlyRootFilesystem: true
        readOnly: true
        readOnlyRootFilesystem: false
        readOnly: true
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged|readOnlyRootF
ileSystem'
      privileged: true
candidate@cli:~$ kubectl get pod/app -n testing -o yaml | grep -E 'privileged|readOnlyRootFi
leSystem'
      privileged: true
candidate@cli:~$ kubectl get pod/frontend -n testing -o yaml | grep -E 'privileged|readOnlyR
ootFileSystem'
      privileged: false
candidate@cli:~$ kubectl get pod/frontend -n testing -o yaml | grep -E 'privileged|readOnlyR
ootFilesystem'
      privileged: true
      readOnlyRootFilesystem: false
candidate@cli:~$ kubectl delete pod/app -n testing
pod "app" deleted
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged|readOnlyRootF
ilesystem'
      privileged: true
      readOnlyRootFilesystem: false
candidate@cli:~$ kubectl delete pod/smtp -n testing
pod "smtp" deleted
```


☐☐: https://kubernetes.io/docs/concepts/policy/pod-security-policy/ https://cloud.google.com/architecture/best-practices-for-operating-containers

NEW QUESTION: 38

☐☐ ☐☐☐ ☐☐☐☐ ☐☐☐☐/☐☐ ☐☐☐☐☐ ☐☐☐ ☐ ☐☐☐☐: [desk@cli] $ kubectl config use-context dev ☐☐ ☐☐ NetworkPolicy☐ ☐☐ NetworkPolicy☐ ☐☐☐☐ ☐☐ ☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐☐.

☐☐: Ingress + Egress ☐☐☐ ☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐☐☐ ☐☐☐☐☐ ☐☐ ☐☐☐☐☐☐ ☐ default-deny NetworkPolicy☐ ☐☐☐☐ ☐. ☐ NetworkPolicy☐ ☐☐☐☐☐☐ ☐☐☐☐☐ ☐☐ Ingress + Egress ☐☐☐☐ ☐☐☐☐ ☐☐☐.

☐☐ ☐☐☐ default-deny NetworkPolicy☐ ☐☐☐☐☐☐ ☐☐☐☐☐ ☐☐ ☐☐ ☐☐ ☐☐☐ ☐☐☐☐☐.

/home/cert_masters/network-policy.yaml☐☐ ☐☐☐☐ ☐☐☐☐☐ ☐☐☐ ☐☐ ☐ ☐☐☐☐.

Answer:

master1 $ k ☐☐ ☐☐☐☐ -n ☐☐☐ --show-labels

☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐

test-pod 1/1 ☐☐ ☐ 0 34s role=test,run=test-pod

☐☐☐ 1/1 ☐☐ ☐ 0 17d ☐☐=☐☐☐ ☐

$ vim netpol.yaml
API□□:networking.k8s.io/v1
□□: NetworkPolicy
□□□□□:
□□: □□ □□□□
□□□□□□: □□□
□□:
□□□□□: {}
□□ □□:
- □□□□
- □□
master1 $ k □□ -f netpol.yaml
□□
controlplane $ k □□ □□□□ -n □□□ --show-labels
□□ □□ □□ □□ □□ □□ □□
test-pod 1/1 □□ □ 0 34s role=test,run=test-pod
□□□ 1/1 □□ □ 0 17d □□=□□□ □
master1 $ vim netpol1.yaml
API□□:networking.k8s.io/v1
□□: NetworkPolicy
□□□□□:
□□: □□ □□□□
□□□□□□: □□□
□□:
□□□□□: {}
□□ □□:
- □□□□
- □□
master1 $ k apply -f netpol1.yaml □□: https://kubernetes.io/docs/concepts/services-networking/network-policies/ □□ controlplane $ k get pods -n test --show-labels □□ □□ □□ □□ □□ □□ LABELS test-pod 1/1 □□ □ 0 34s role=test,run=test-pod □□□ 1/1 □□ □ 0 17d run=testing master1 $ vim netpol1.yaml apiVersion:networking.k8s.io/v1 □□: NetworkPolicy □□□□□:
□□: □□ □□□□
□□□□□□: □□□
□□:
□□□□□: {}
□□ □□:
- □□□□
- □□
master1 $ k apply -f netpol1.yaml □□: https://kubernetes.io/docs/concepts/services-networking/network-policies/

NEW QUESTION: 39

□□□ □□ □□ □□□ □□□□ □□□ □□ □□ □□□ □□ □□□□ □ □□□ □□□□□ □□□.
API □□□□ □□□ □□ □□ □□□ □□ □□□□□.- a. --authorization-mode □□□ RBAC□ □□□□ □□□ □□□□□. b. --authorization-mode □□□ □□ c□ □□□□ □□□ □□□□□. --profiling □□□ false□ □□□□ □□□ □□□□□□. Kubelet□ □□ □□□ □□ □□ □□□ □□ □□□□□□.- a. --anonymous-auth □□□ false□ □□□□ □□□ □□□□□.
□. --authorization-mode □□□ Webhook□ □□□□ □□□ □□□□□.
ETCD□ □□ □□□ □□ □□ □□□ □□ □□□□□.
□. --auto-tls □□□ true□ □□□□ □□□□ □□□□□.
□□: Tool Kube-Bench□ □□□□□
Answer:
API □□:
--authorization-mode □□□ RBAC□ □□□□ □□□ □□□□□.
□□ □□ □□□ □□□ □□□□□□. RBAC(□□ □□ □□□ □□)□ □□□□ □□ □□□□ □□□□□ □□ □□□ □□ □□□ □ □ □ □□□ □□□□ □□□ □ □□□□. RBAC □□ □□□ □□□□ □□ □□□□.
□□ - □□ □□
□□□□□
api□□: v1
□□: □□
□□□□□:
□□□□□□□: null
□□:
□□□□: kube-apiserver
□□: □□ □□
□□: kube-apiserver
□□□□□□: kube-system
□□:
□□□□:
- □□:
+ - kube-apiserver
+ - --authorization-mode=RBAC,□□
□□□: gcr.io/google_containers/kube-apiserver-amd64:v1.6.0
□□ □□□:
□□□□□: 8
httpGet:
□□□: 127.0.0.1
□□: /healthz
□□: 6443
□□□: HTTPS
□□ □□□□: 15
□□: kube-apiserver-should-pass
□□:
□□:

CPU: 250m
□□□□□:
- □□□ □□: /etc/kubernetes/
□□: k8s
□□ □□: □□
- □□□ □□: /etc/ssl/certs
□□: □□□
- □□□ □□: /etc/pki
□□: pki
□□□□□□□: □
□□:
- □□□ □□:
□□: /etc/kubernetes
□□: k8s
- □□□ □□:
□□: /etc/ssl/certs
□□: □□□
- □□□ □□:
□□: /etc/pki
□□: pki
--authorization-mode □□□ Node□ □□□□ □□□ □□□□□.
□□: □□□ □□□□ API □□ □□ □□ □□ /etc/kubernetes/manifests/kube-apiserver.yaml□ □□□□ --authorization-mode □□ □□□
Node.js□ □□□□ □□□ □□□□□.
--authorization-mode=□□,RBAC
□□:
/bin/ps -ef | grep kube-apiserver | grep -v grep
□□ □□:
'□□,RBAC'□□ '□□'□ □□□□.
--profiling □□□ false□ □□□□ □□□ □□□□□.
□□: □□□ □□□□ API □□ □□ □□ □□ /etc/kubernetes/manifests/kube-apiserver.yaml□ □□□□ □□ □□ □□□ □□□□□.
--□□□□□=false
□□:
/bin/ps -ef | grep kube-apiserver | grep -v grep
□□ □□:
'□□'□ '□□'□ □□□□
Kubelet□ □□ □□□ □□ □□ □□□ □□ □□□□□. - --anonymous-auth □□□ false□ □□□□ □□□ □□□□□.
□□ □□: Kubelet □□ □□□ □□□□ □□ □□□ □□□□ □□: □□: □□□□ false□ □□□□□. □□ □□□ □□□ □□□□ □□
□ □□□ □□□□ kubelet □□□ □□ /etc/systemd/system/kubelet.service.d/10-kubeadm.conf□ □□□□
KUBELET_SYSTEM_PODS_ARGS □□□ □□ □□□□□ □□□□□.
--□□-□□=false
□□□□ □□ kubelet □□□□ □□ □□□□□. □□ □□:

systemctl □□ □□ □□
systemctl □□□ kubelet.service
□□:
/bin/ps -fC kubelet
□□ □□:
/bin/cat /var/lib/kubelet/config.yaml
□□ □□:
'□□'□ '□□'□ □□□□
2) --authorization-mode □□□ Webhook□□ □□□□ □□□ □□□□□.
□□
□□ □□ kubelet | jq -e '.[0].Args[] | match("--authorization-mode=Webhook").string' □□ □: --authorization-mode=Webhook ETCD□ □□ □□□ □□ □□ □□□ □□ □□□□□.- a. --auto-tls □□□ true□ □□□□ □□□□ □□□□□□. TLS□ □□ □□□ □□□□ □□□ □ □□□□. etcd□ □□ REST API □□□ □□ □□□ □□ Kubernetes □□□□ □□□□ □□□□ □ □ □□□□□□. □□□ □□□ □ □□□□ □□□□□ □□□□ □□ □□□□□□ □□□ □ □□□ □□□. etcd □□□□ □□ □□□□ □□□□□ □□□ □□□□ □□ □□□□□ □□□ □□□□□ □□□.
□□ - □□ □□
□□□□□
api□□: v1
□□: □□
□□□□□:
□□:
Scheduler.alpha.kubernetes.io/tical-pod: ""
□□□□□□□: null
□□:
□□□□: etcd
□□: □□ □□
□□: etcd
□□□□□□: kube-system
□□:
□□□□:
- □□:
+ - □□
+ - --auto-tls=true
□□□: k8s.gcr.io/etcd-amd64:3.2.18
imagePullPolicy: IfNotPresent
□□ □□□:
□□:
□□:
- /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 □□
DelaySeconds: 15 timeoutSeconds: 15 □□: etcd -□□□□ □ □□□: {} VolumeMounts:
- □□□ □□: /var/lib/etcd
□□: etcd-data
- □□□ □□: /etc/kubernetes/pki/etcd
□□: etcd-certs
□□□□□□□: □
PriorityClassName: □□□ □□□□ □□
□□:
- □□□ □□:
□□: /var/lib/etcd
□□: □□□□□□□□□
□□: etcd-data
- □□□ □□:
□□: /etc/kubernetes/pki/etcd
□□: □□□□□□□□□
□□: etcd-certs
□□: {}
□□:

```
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_mana
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_mana
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 14:19:38.144042  134205 kubelet.go:1698] "Deleted mirror pod because it is outdat
5]: I0520 14:19:40.037048  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 --

:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kube-proxy\"
:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"lib-modules\"
:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"flannel-cfg\"
:157] "Reconciler: start to sync state"
55] Waited for 1.049946364s due to client-side throttling, not priority and fairness, reque
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
593] "Trying to delete pod" pod="kube-system/kube-apiserver-kscs00201-master" podUID=bb91e1
598] "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; 1s 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)
```

NEW QUESTION: 40
□□□□ □□□ □□□□ □□□ AppArmor □□□□ □□□□□.
#include <□□□/□□>
□□□ nginx-deny □□□=(attach_disconnected) {
#include <□□□/□□□>
□□,
□□ □□ □□□ □□□□□.
□□ /** w,
}
EOF'
AppArmor □□□□ □□□□□ □□□ □□□□□ □□□ □□□□□.
api□□: v1
□□: □□
□□□□□:
□□: apparmor-pod
□□:
□□□□:
- □□: apparmor-pod
□□□: nginx
□□□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□ □□□□□.
□□: □□□ □□□□ □□ □□□ □□□ □□□□.
Answer:

```
candidate@cli:~$ kubectl config use-context KSSH00401
Switched to context "KSSH00401".
candidate@cli:~$ ssh kssh00401-worker1
Warning: Permanently added '10.240.86.172' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@kssh00401-worker1:~# head /etc/apparmor.d/nginx_apparmor
#include <tunables/global>

profile nginx-profile-2 flags=(attach_disconnected,mediate_deleted) {
 #include <abstractions/base>
  network inet tcp,
  network inet udp,
  network inet icmp,

  deny network raw,

root@kssh00401-worker1:~# apparmor_parser -q /etc/apparmor.d/nginx_apparmor
root@kssh00401-worker1:~# exit
logout
Connection to 10.240.86.172 closed.
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
candidate@cli:~$ vim KSSH00401/nginx-pod.yaml
```

```
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-pr
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
~
```

```
candidate@cli:~$ vim KSSH00401/nginx-pod.yaml
candidate@cli:~$ kubectl create -f KSSH00401/nginx-pod.yaml
pod/nginx-pod created
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-profile-2
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
```