CKS Q&As

Certified Kubernetes Security Specialist (CKS) Exam

Pass Linux Foundation CKS Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/cks.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

QUESTION 1

Analyze and edit the given Dockerfile

1.

FROM ubuntu:latest

2.

RUN apt-get update -y

3.

RUN apt-install nginx -y

4.

COPY entrypoint.sh /

5.

ENTRYPOINT ["/entrypoint.sh"]

6.

USER ROOT

Fixing two instructions present in the file being prominent security best practice issues

Analyze and edit the deployment manifest file

1.

apiVersion: v1

2.

kind: Pod

3.

metadata:

4.

name: security-context-demo-2

5.

spec:

6.

securityContext:

7.

runAsUser: 1000

8.

containers:

9.

- name: sec-ctx-demo-2 10.image: gcr.io/google-samples/node-hello:1.0 11.securityContext: 12.runAsUser: 0 13.privileged: True 14.allowPrivilegeEscalation: false

Fixing two fields present in the file being prominent security best practice issues

Don\\'t add or remove configuration settings; only modify the existing configuration settings

Whenever you need an unprivileged user for any of the tasks, use user test-user with the user id 5487

A. See the explanation below:

B. PlaceHolder

Correct Answer: A

FROM debian:latest MAINTAINER k@bogotobogo.com

1 - RUN RUN apt-get update andand DEBIAN_FRONTEND=noninteractive apt-get install -yq apt-utils RUN DEBIAN_FRONTEND=noninteractive apt-get install -yq htop RUN apt-get clean

2 - CMD #CMD ["htop"] #CMD ["ls", "-l"]

3 - WORKDIR and ENV WORKDIR /root ENV DZ version1 $ docker image build -t bogodevops/demo . Sending build context to Docker daemon 3.072kB

Step 1/7 : FROM debian:latest ---> be2868bebaba

Step 2/7 : MAINTAINER k@bogotobogo.com ---> Using cache ---> e2eef476b3fd

Step 3/7 : RUN apt-get update andand DEBIAN_FRONTEND=noninteractive apt-get install -yq apt-utils ---> Using cache ---> 32fd044c1356

Step 4/7 : RUN DEBIAN_FRONTEND=noninteractive apt-get install -yq htop ---> Using cache ---> 0a5b514a209e

Step 5/7 : RUN apt-get clean ---> Using cache ---> 5d1578a47c17

Step 6/7 : WORKDIR /root ---> Using cache ---> 6b1c70e87675

Step 7/7 : ENV DZ version1 ---> Using cache ---> cd195168c5c7 Successfully built cd195168c5c7 Successfully tagged bogodevops/demo:latest

QUESTION 2

Create a PSP that will only allow the persistentvolumeclaim as the volume type in the namespace restricted.

Create a new PodSecurityPolicy named prevent-volume-policy which prevents the pods which is having different volumes mount apart from persistentvolumeclaim.

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

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

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

Hint:

Also, Check the Configuration is working or not by trying to Mount a Secret in the pod maifest, it should get failed.

POD Manifest:

1.

apiVersion: v1

2.

kind: Pod

3.

metadata:

4.

name:

5.

spec:

6.

containers:

7.

- name:

8.

image:

9.

volumeMounts: 10.- name: 11.mountPath: 12.volumes: 13.- name: 14.secret: 15.secretName:

A. See the below:

B. PlaceHolder

Correct Answer: A

apiVersion: policy/v1beta1

kind: PodSecurityPolicy

metadata:

name: restricted

annotations:

seccomp.security.alpha.kubernetes.io/allowedProfileNames:

\\'docker/default,runtime/default\\'

apparmor.security.beta.kubernetes.io/allowedProfileNames: \\'runtime/default\\'
seccomp.security.alpha.kubernetes.io/defaultProfileName: \\'runtime/default\\'
apparmor.security.beta.kubernetes.io/defaultProfileName: \\'runtime/default\\' spec:

privileged: false

Required to prevent escalations to root.

allowPrivilegeEscalation: false

This is redundant with non-root + disallow privilege escalation, # but we can provide it for defense in depth.

requiredDropCapabilities:

-ALL

Allow core volume types.

volumes:

-\\'configMap\\'

-\\'emptyDir\\'

-\\'projected\\'

-\\'secret\\'

-\\'downwardAPI\\'

Assume that persistentVolumes set up by the cluster admin are safe to use.

-\\'persistentVolumeClaim\\'

hostNetwork: false

hostIPC: false

hostPID: false

runAsUser:

Require the container to run without root privileges.

rule: \\'MustRunAsNonRoot\\'

seLinux:

This policy assumes the nodes are using AppArmor rather than SELinux.

rule: \\'RunAsAny\\'

supplementalGroups:

rule: \\'MustRunAs\\'

ranges:

Forbid adding the root group.

-

min: 1

max: 65535

fsGroup:

rule: \\'MustRunAs\\'

ranges:

Forbid adding the root group.

-

min: 1

max: 65535

readOnlyRootFilesystem: false

QUESTION 3

Fix all issues via configuration and restart the affected components to ensure the new setting takes effect.

Fix all of the following violations that were found against the API server:

1.

Ensure the --authorization-mode argument includes RBAC

2.

Ensure the --authorization-mode argument includes Node

3.

Ensure that the --profiling argument is set to false

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

1.

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

2.

Ensure that the --authorization-mode argument is set to Webhook. Fix all of the following violations that were found against the ETCD:

Ensure that the --auto-tls argument is not set to true Hint: Take the use of Tool Kube-Bench

A. See the below.

B. PlaceHolder

Correct Answer: A

API server:

Ensure the --authorization-mode argument includes RBAC

Turn on Role Based Access Control.Role Based Access Control (RBAC) allows fine- grained control over the operations that different entities can perform on different objects in the cluster. It is recommended to use the RBAC authorization

mode.

Fix - BuildtimeKubernetesapiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

component: kube-apiserver

tier: control-plane

name: kube-apiserver

namespace: kube-system

spec:

containers:

-command: + - kube-apiserver + - --authorization-mode=RBAC,Node image: gcr.io/google_containers/kube-apiserver-amd64:v1.6.0 livenessProbe: failureThreshold: 8 httpGet: host: 127.0.0.1 path: /healthz port: 6443 scheme: HTTPS initialDelaySeconds: 15 timeoutSeconds: 15 name: kube-apiserver-should-pass resources: requests: cpu: 250m volumeMounts:

-

mountPath: /etc/kubernetes/ name: k8s readOnly: true

-

mountPath: /etc/ssl/certs name: certs

-

mountPath: /etc/pki name: pki hostNetwork: true volumes:

-

hostPath: path: /etc/kubernetes name: k8s

-

hostPath: path: /etc/ssl/certs name: certs

-

hostPath: path: /etc/pki name: pki

Ensure the --authorization-mode argument includes Node

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube- apiserver.yaml on the master node and set the --authorization-mode parameter to a value that includes Node.

--authorization-mode=Node,RBAC

Audit:

/bin/ps -ef | grep kube-apiserver | grep -v grep

Expected result:

\\'Node,RBAC\\' has \\'Node\\'

Ensure that the --profiling argument is set to false

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the below parameter.

--profiling=false

Audit:

/bin/ps -ef | grep kube-apiserver | grep -v grep

Expected result:

\\'false\\' is equal to \\'false\\'

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

uk.co.certification.simulator.questionpool.PList@e3e35a0

Remediation: If using a Kubelet config file, edit the file to set authentication: anonymous:

enabled to false. If using executable arguments, edit the kubelet service file /etc/systemd/system/kubelet.service.d/10-kubeadm.conf on each worker node and set the below parameter in KUBELET_SYSTEM_PODS_ARGS variable.

--anonymous-auth=false

Based on your system, restart the kubelet service. For example:

systemctl daemon-reload

systemctl restart kubelet.service

Audit:

/bin/ps -fC kubelet

Audit Config:

/bin/cat /var/lib/kubelet/config.yaml

Expected result:

\\'false\\' is equal to \\'false\\'

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

Audit

docker inspect kubelet | jq -e \\'.[0].Args[] | match("--authorization- mode=Webhook").string\\'

Returned Value: --authorization-mode=Webhook

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

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

Do not use self-signed certificates for TLS. etcd is a highly-available key value store used by Kubernetes deployments for persistent storage of all of its REST API objects. These objects are sensitive in nature and should not be available to unauthenticated clients. You should enable the client authentication via valid certificates to secure the access to the etcd service.

Fix - BuildtimeKubernetesapiVersion: v1 kind: Pod metadata: annotations: scheduler.alpha.kubernetes.io/critical-pod: "" creationTimestamp: null labels: component: etcd tier: control-plane name: etcd namespace: kube-system spec: containers:

-command:

+ - etcd

+ - --auto-tls=true

image: k8s.gcr.io/etcd-amd64:3.2.18

imagePullPolicy: IfNotPresent

livenessProbe:

exec:

command:

-/bin/sh

- -ec

-ETCDCTL_API=3 etcdctl --endpoints=https://[192.168.22.9]:2379 -- cacert=/etc/kubernetes/pki/etcd/ca.crt

--cert=/etc/kubernetes/pki/etcd/healthcheck-client.crt -- key=/etc/kubernetes/pki/etcd/healthcheck-client.key get foo

failureThreshold: 8

initialDelaySeconds: 15

timeoutSeconds: 15

name: etcd-should-fail

resources: {}

volumeMounts:

-

mountPath: /var/lib/etcd

name: etcd-data

-

mountPath: /etc/kubernetes/pki/etcd

name: etcd-certs

hostNetwork: true

priorityClassName: system-cluster-critical

volumes:

-

hostPath:

path: /var/lib/etcd

type: DirectoryOrCreate

name: etcd-data

-

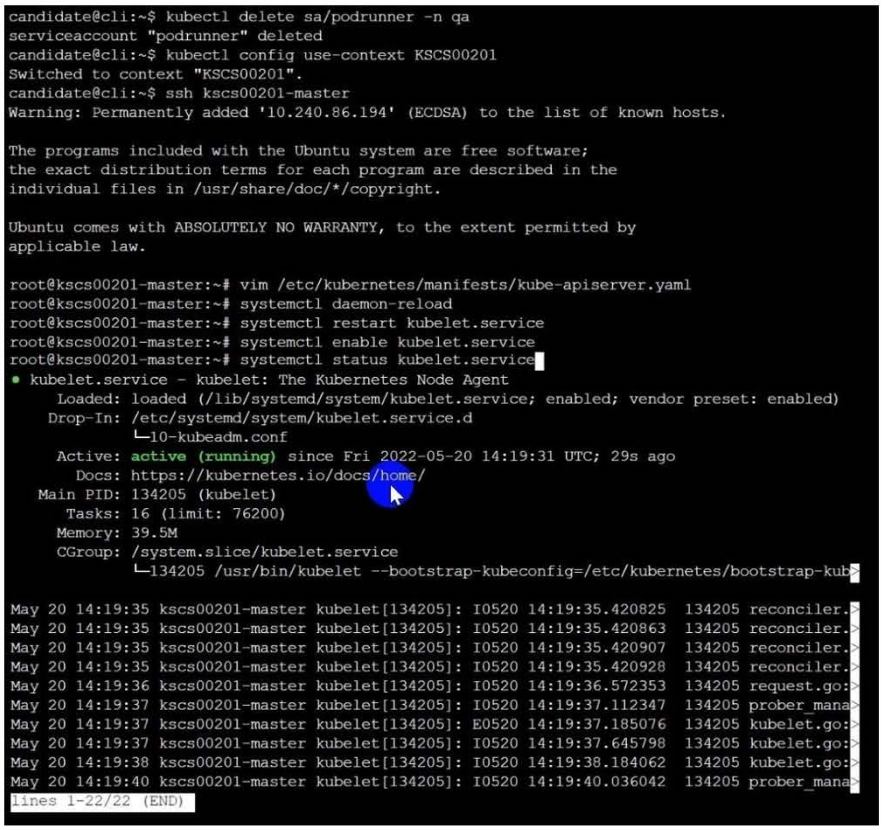
hostPath:

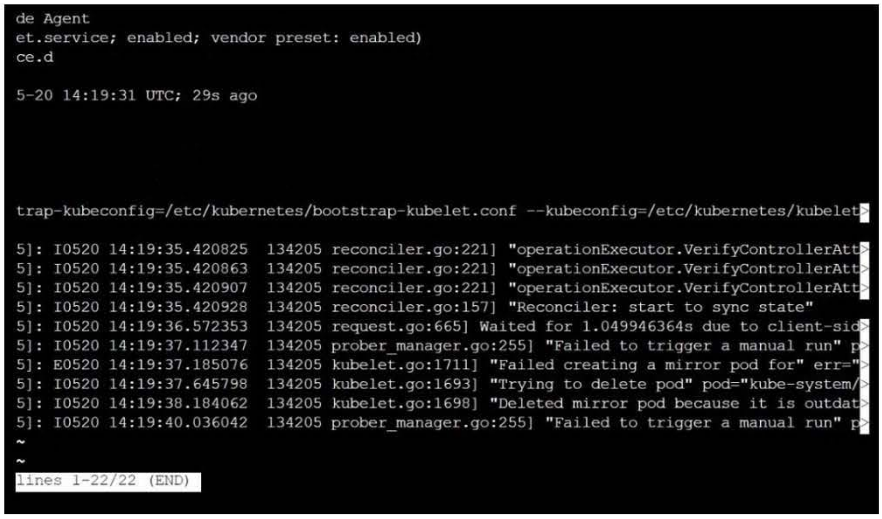
path: /etc/kubernetes/pki/etcd

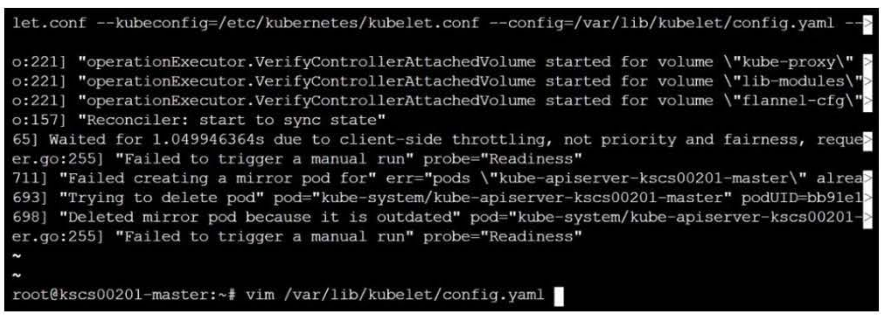
type: DirectoryOrCreate

name: etcd-certs

status: {}


candidate@cli:~$ kubectl delete sa/podrunner -n qa
serviceaccount "podrunner" deleted
candidate@cli:~$ kubectl config use-context KSCS00201
Switched to context "KSCS00201".
candidate@cli:~$ ssh kscs00201-master
Warning: Permanently added '10.240.86.194' (ECDSA) to the list of known hosts.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
root@kscs00201-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@kscs00201-master:~# systemctl daemon-reload
root@kscs00201-master:~# systemctl restart kubelet.service
root@kscs00201-master:~# systemctl enable kubelet.service
root@kscs00201-master:~# systemctl status kubelet.service
kubelet.service - kubelet: The Kubernetes Node Agent
Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
Drop-In: /etc/systemd/system/kubelet.service.d
10-kubeadm.conf
Active: active (running) since Fri 2022-05-20 14:19:31 UTC; 29s ago
Docs: https://kubernetes.io/docs/home/
Main PID: 134205 (kubelet)
Tasks: 16 (limit: 76200)
Memory: 39.5M
CGroup: /system.slice/kubelet.service
134205 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kub
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420825 134205 reconciler.
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420863 134205 reconciler.
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420907 134205 reconciler.
May 20 14:19:35 kscs00201-master kubelet[134205]: I0520 14:19:35.420928 134205 reconciler.
May 20 14:19:36 kscs00201-master kubelet[134205]: I0520 14:19:36.572353 134205 request.go:
May 20 14:19:37 kscs00201-master kubelet[134205]: I0520 14:19:37.112347 134205 prober_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 14:19:38.184062 134205 kubelet.go:1698] "Deleted mirror pod because it is outdat
5]: I0520 14:19:40.036042 134205 prober_manager.go:255] "Failed to trigger a manual run" p
~
~
lines 1-22/22 (END)


let.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"kube-proxy\"
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"lib-modules\"
o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"flannel-cfg\"
o:157] "Reconciler: start to sync state"
65] Waited for 1.049946364s due to client-side throttling, not priority and fairness, 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
693] "Trying to delete pod" pod="kube-system/kube-apiserver-kscs00201-master" podUID=bb91e1
698] "Deleted mirror pod because it is outdated" pod="kube-system/kube-apiserver-kscs00201-
er.go:255] "Failed to trigger a manual run" probe="Readiness"
~
~
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1
authentication:
  anonymous:
    enabled: false
  webhook:
    cacheTTL: 0s
    enabled: true
  x509:
    clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
cgroupDriver: systemd
clusterDNS:
```

```
~
~
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml
root@kscs00201-master:~# vim /etc/kubernetes/manifests/etcd.yaml
root@kscs00201-master:~# systemctl daemon-reload
root@kscs00201-master:~# systemctl restart kubelet.service
root@kscs00201-master:~# systemctl status kubelet.service
```

```
● kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Fri 2022-05-20 14:22:29 UTC; 4s ago
       Docs: https://kubernetes.io/docs/home/
   Main PID: 135849 (kubelet)
      Tasks: 17 (limit: 76200)
     Memory: 38.0M
     CGroup: /system.slice/kubelet.service
             └─135849 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kub>

May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330232  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330259  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330304  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330354  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330378  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330397  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330415  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330433  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330452  135849 reconciler.>
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330463  135849 reconciler.>
lines 1-22/22 (END)
```

```
May 20 14:22:30 kscs00201-master kubelet[135849]: I0520 14:22:30.330463  135849 reconciler.>
root@kscs00201-master:~#
root@kscs00201-master:~#
root@kscs00201-master:~#
root@kscs00201-master:~# exit
logout
Connection to 10.240.86.194 closed.
candidate@cli:~$
```

QUESTION 4

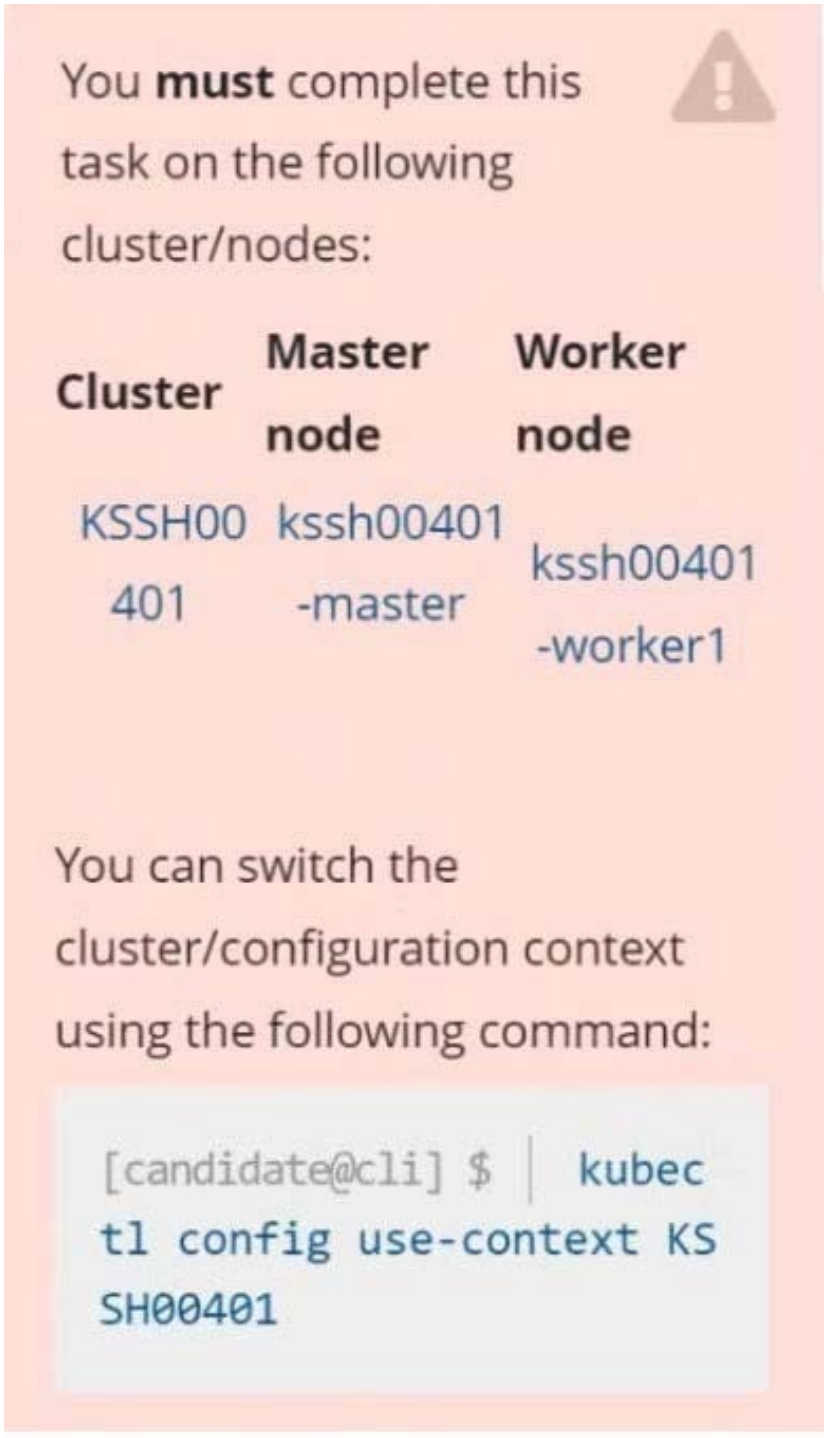
```
candidate@cli:~$ kubectl config use-context KSSC00401
Switched to context "KSSC00401".
candidate@cli:~$ ssh kssc00401-master
Warning: Permanently added '10.240.86.231' (ECDSA) to the list of known hosts.

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@kssc00401-master:~# kubectl get pods -n naboo
NAME           READY   STATUS    RESTARTS   AGE
c-3po          1/1     Running   0          6h48m
chewbacca      1/1     Running   0          6h48m
jawas          1/1     Running   0          6h48m
qui-gon-jinn   1/1     Running   0          6h48m
root@kssc00401-master:~# kubectl get pods -n naboo -o name
pod/c-3po
pod/chewbacca
pod/jawas
pod/qui-gon-jinn
root@kssc00401-master:~# for i in $(kubectl get pods -n naboo -o name)
> do
> kubectl get ${i} -o yaml | grep -i image
> done
Error from server (NotFound): pods "c-3po" not found
Error from server (NotFound): pods "chewbacca" not found
Error from server (NotFound): pods "jawas" not found
Error from server (NotFound): pods "qui-gon-jinn" not found
root@kssc00401-master:~# for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo
get ${i} -o yaml | grep -i image ; done
    image: centos:centos7.9.2009
    imagePullPolicy: Never
    image: centos:centos7.9.2009
    imageID: docker-pullable://centos@sha256:c73f515d06b0fa07bb18d8202035e739a494ce760aa7312
9f60f4bf2bd22b407
    image: photon:3.0
    imagePullPolicy: Never
    image: photon:3.0
    imageID: docker-pullable://photon@sha256:c48d61f0f3ad19215b75e2087cfbe95d7321abb454e4295
a0e6c38f563ece622
    image: alpine:3.7
    imagePullPolicy: Never
    image: alpine:3.7
    imageID: docker-pullable://alpine@sha256:8421d9a84432575381bfabd248f1eb56f3aa21d9d7cd251
1583c68c9b7511d10
    image: amazonlinux:2
    imagePullPolicy: Never
    image: amazonlinux:2
    imageID: docker-pullable://amazonlinux@sha256:246ef631c75ea83005889621119fd5cc9cbb5500e1
93707c38b6c060d597a146
root@kssc00401-master:~# trivy image centos:centos7.9.2009
2022-05-20T15:39:51.733Z        INFO    Need to update DB
2022-05-20T15:39:51.733Z        INFO    Downloading DB...
27.97 MiB / 27.97 MiB [-----------------------------------------] 100.00% 27.43 MiB p/s 1s
```

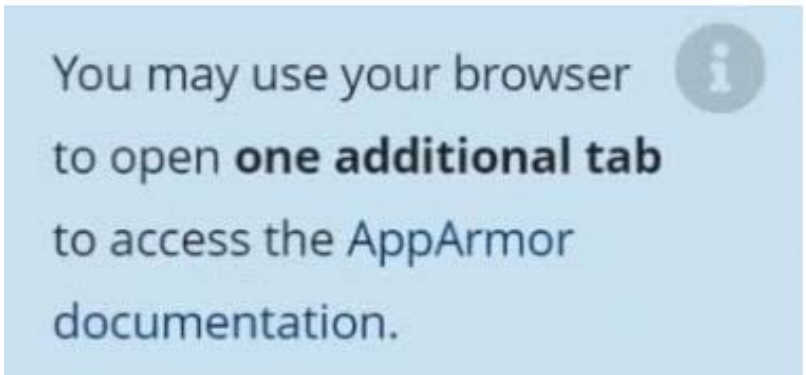
```
root@kssc00401-master:~# for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo
get ${i} -o yaml | grep -i image ; done
    image: centos:centos7.9.2009
    imagePullPolicy: Never
    image: centos:centos7.9.2009
    imageID: docker-pullable://centos@sha256:c73f515d06b0fa07bb18d8202035e739a494ce760aa7312
9f60f4bf2bd22b407
    image: photon:3.0
    imagePullPolicy: Never
    image: photon:3.0
    imageID: docker-pullable://photon@sha256:c48d61f0f3ad19215b75e2087cfbe95d7321abb454e4295
a0e6c38f563ece622
    image: alpine:3.7
    imagePullPolicy: Never
    image: alpine:3.7
    imageID: docker-pullable://alpine@sha256:8421d9a84432575381bfabd248f1eb56f3aa21d9d7cd251
1583c68c9b7511d10
    image: amazonlinux:2
    imagePullPolicy: Never
    image: amazonlinux:2
    imageID: docker-pullable://amazonlinux@sha256:246ef631c75ea83005889621119fd5cc9cbb5500e1
93707c38b6c060d597a146
root@kssc00401-master:~# trivy image photon:3.0
2022-05-20T15:40:18.003Z        INFO    Detected OS: photon
2022-05-20T15:40:18.003Z        INFO    Detecting Photon Linux vulnerabilities...
2022-05-20T15:40:18.005Z        INFO    Number of language-specific files: 0

photon:3.0 (photon 3.0)
======================
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
```

```
root@kssc00401-master:~# kubectl get pods -n naboo -o name
pod/c-3po
pod/chewbacca
pod/jawas
pod/qui-gon-jinn
root@kssc00401-master:~# kubectl -n naboo pod/c-3po -o yaml | grep image
Error: flags cannot be placed before plugin name: -n
root@kssc00401-master:~# kubectl -n naboo get pod/c-3po -o yaml | grep image
    image: centos:centos7.9.2009
    imagePullPolicy: Never
    image: centos:centos7.9.2009
    imageID: docker-pullable://centos@sha256:c73f515d06b0fa07bb18d8202035e739a494ce760aa7312
9f60f4bf2bd22b407
root@kssc00401-master:~# kubectl -n naboo delete pod/c-3po
pod "c-3po" deleted
root@kssc00401-master:~# kubectl -n naboo delete pod/jawas
pod "jawas" deleted
```

```
pod "jawas" deleted
root@kssc00401-master:~# history
    1  kubectl get pods -n naboo
    2  kubectl get pods -n naboo -o name
    3  for i in $(kubectl get pods -n naboo -o name); do kubectl get ${i} -o yaml | grep -i
image ; done
    4  for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get ${i} -o yaml |
 grep -i image ; done
    5  trivy image centos:centos7.9.2009
    6  for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get ${i} -o yaml |
 grep -i image ; done
    7  trivy image photon:3.0
    8  for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get ${i} -o yaml |
 grep -i image ; done
    9  trivy image alpine:3.7
   10  for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get ${i} -o yaml |
 grep -i image ; done
   11  trivy image amazonlinux:2
   12  kubectl get pods -n naboo -o name
   13  kubectl -n naboo pod/c-3po -o yaml | grep image
   14  kubectl -n naboo get pod/c-3po -o yaml | grep image
   15  kubectl -n naboo delete pod/c-3po
   16  kubectl -n naboo delete pod/jawas
   17  history
root@kssc00401-master:~#
```

AppArmor is enabled on the cluster\\'s worker node. An AppArmor profile is prepared, but not enforced yet.


You must complete this task on the following cluster/nodes:
Cluster
Master node
Worker node
KSSH00401
kssh00401-master
kssh00401-worker1
You can switch the cluster/configuration context using the following command:
[candidate@cli] $ kubectl config use-context KSSH00401


You may use your browser to open one additional tab to access the AppArmor documentation.

Task

On the cluster\\'s worker node, enforce the prepared AppArmor profile located at /etc/apparmor.d/nginx_apparmor.

Edit the prepared manifest file located at /home/candidate/KSSH00401/nginx-pod.yaml to apply the AppArmor profile.

Finally, apply the manifest file and create the Pod specified in it.

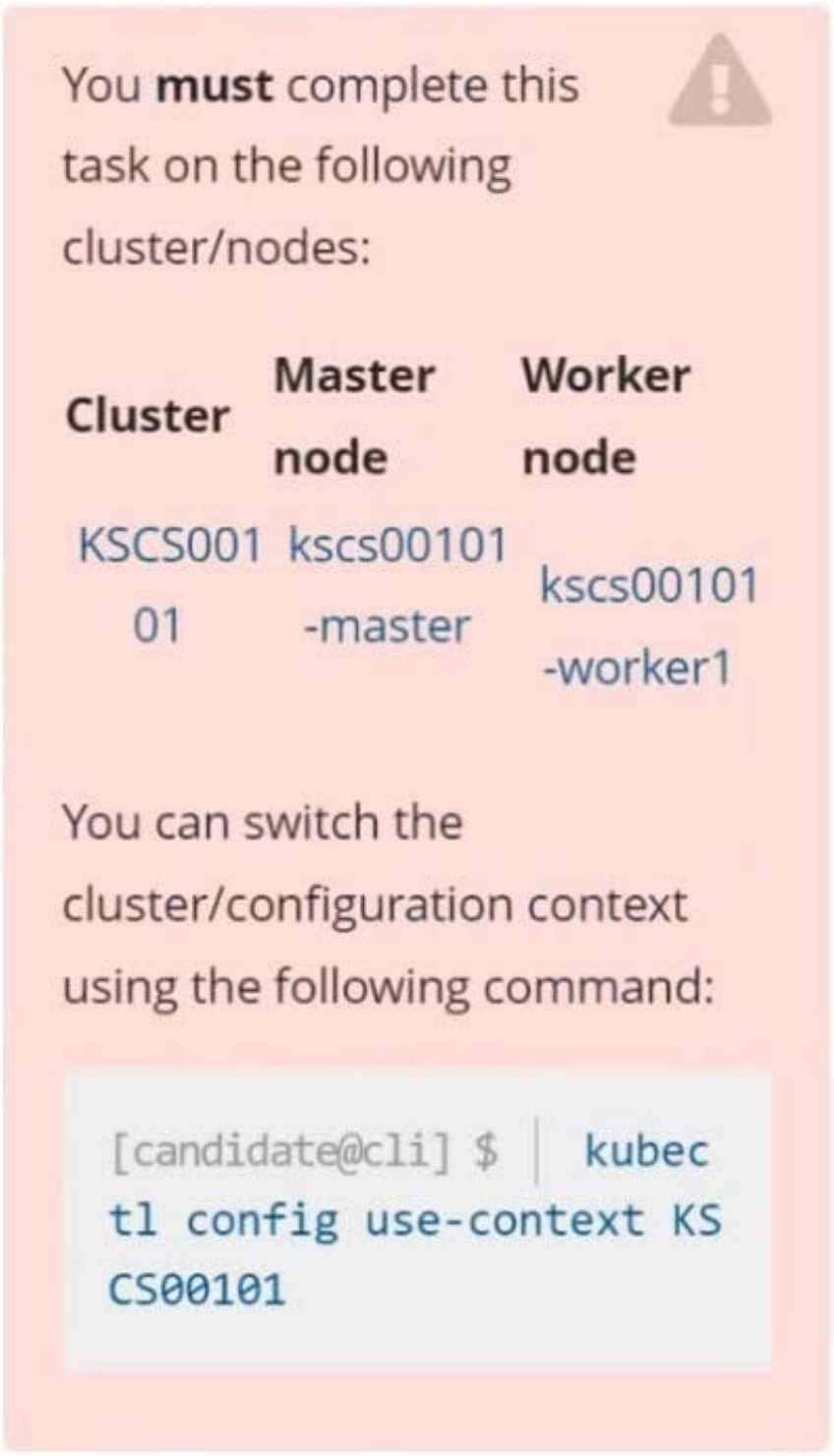
A. See the explanation below

B. PlaceHolder

Correct Answer: A

QUESTION 5

CORRECT TEXT Context



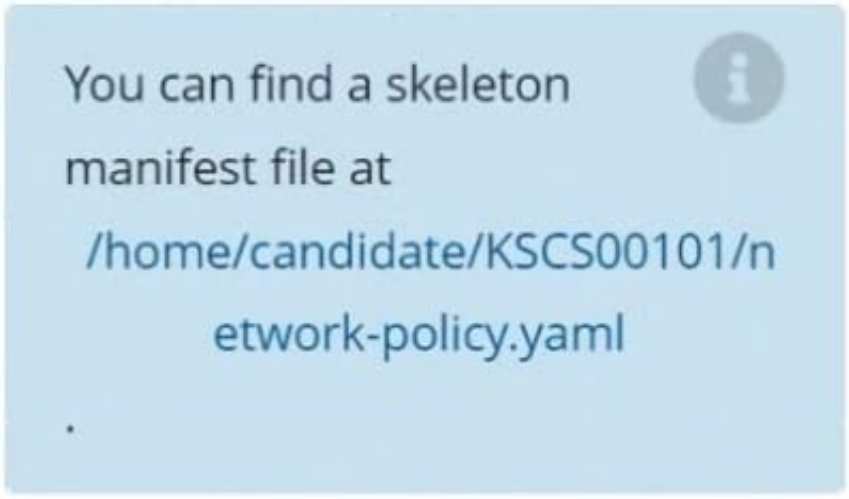
A default-deny NetworkPolicy avoids to accidentally expose a Pod in a namespace that doesn\\'t have any other NetworkPolicy defined.

Task

Create a new default-deny NetworkPolicy named defaultdeny in the namespace testing for all traffic of type Egress.

The new NetworkPolicy must deny all Egress traffic in the namespace testing.

Apply the newly created default-deny NetworkPolicy to all Pods running in namespace testing.



A. See explanation below.

B. PlaceHolder

Correct Answer: A