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

```
candidate@cli:~$ kubectl config use-context KSRS00602
Switched to context "KSRS00602".
candidate@cli:~$ ssh ksrs00602-master
Warning: Permanently added '10.240.86.243' (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@ksrs00602-master:~# cat /etc/kubernetes/logpolicy/sample-policy.yaml
---
apiVersion: audit.k8s.io/v1
kind: Policy
# Don't generate audit events for all requests in RequestReceived stage.
omitStages:
  - "RequestReceived"
rules:
  # Don't log watch requests by the "system:kube-proxy" on endpoints or services
  - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
    - group: "" # core API group
      resources: ["endpoints", "services"]

  # Don't log authenticated requests to certain non-resource URL paths.
  - level: None
    userGroups: ["system:authenticated"]
    nonResourceURLs:
    - "/api*" # Wildcard matching.
    - "/version"
  # Edit form here below
root@ksrs00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
```

```
    - "/api*" # Wildcard matching.
    - "/version"
  # Edit form here below
  - level: RequestResponse
    resources:
    - group: ""
      resources: ["cronjobs"]
  - level: Request
    resources:
    - group: "" # core API group
      resources: ["pods"]
      namespaces: ["webapps"]
# Log configmap and secret changes in all other namespaces at the Metadata level.
  - level: Metadata
    resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
```

```
    - "/version"
  # Edit form here below
  - level: RequestResponse
    resources:
    - group: ""
      resources: ["cronjobs"]
  - level: Request
    resources:
    - group: "" # core API group
      resources: ["pods"]
      namespaces: ["webapps"]
# Log configmap and secret changes in all other namespaces at the Metadata level.
  - level: Metadata
    resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
root@ksrs00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
root@ksrs00602-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
    - command:
        - kube-apiserver
        - --advertise-address=10.240.86.243
        - --allow-privileged=true
        - --audit-policy-file=/etc/kubernetes/logpolicy/sample-policy.yaml
        - --audit-log-path=/var/log/kubernetes/kubernetes-logs.txt
        - --audit-log-maxbackup=1
        - --audit-log-maxage=30
        - --authorization-mode=Node,RBAC
        - --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
```

```
  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
root@ksrs00602-master:~# vim /etc/kubernetes/logpolicy/sample-policy.yaml
root@ksrs00602-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@ksrs00602-master:~# systemctl daemon-reload
root@ksrs00602-master:~# systemctl restart kubelet.service
root@ksrs00602-master:~# systemctl enable kubelet
root@ksrs00602-master:~# exit
logout
Connection to 10.240.86.243 closed.
candidate@cli:~$
```

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

[desk@cli] $ kubectl config use-context dev

Context:

A CIS Benchmark tool was run against the kubeadm created cluster and found multiple issues that must be addressed.

Task:

Fix all issues via configuration and restart the affected components to ensure the new settings take effect.

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

1.2.7 authorization-mode argument is not set to AlwaysAllow FAIL

1.2.8 authorization-mode argument includes Node FAIL

1.2.7 authorization-mode argument includes RBAC FAIL

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

4.2.1 Ensure that the anonymous-auth argument is set to false FAIL

4.2.2 authorization-mode argument is not set to AlwaysAllow FAIL (Use Webhook autumn/authz where possible)

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

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

A. See the explanation below

B. PlaceHolder

Correct Answer: A

worker1 $ vim /var/lib/kubelet/config.yaml uk.co.certification.simulator.questionpool.PList@132b77a0 worker1 $ systemctl restart kubelet. # To reload kubelet configssh to master1master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml- -- authorizationmode=Node,RBACmaster1 $ vim /etc/kubernetes/manifests/etcd.yaml- --client-cert-auth=true

Explanationssh to worker1worker1 $ vim /var/lib/kubelet/config.yaml apiVersion: kubelet.config.k8s.io/v1beta1 authentication: anonymous: enabled: true #Delete this enabled: false #Replace by this webhook: cacheTTL: 0s enabled: true x509: clientCAFile: /etc/kubernetes/pki/ca.crt authorization: mode: AlwaysAllow #Delete this mode: Webhook #Replace by this webhook: cacheAuthorizedTTL: 0s cacheUnauthorizedTTL: 0s cgroupDriver: systemd clusterDNS:

-10.96.0.10 clusterDomain: cluster.local cpuManagerReconcilePeriod: 0s evictionPressureTransitionPeriod: 0s fileCheckFrequency: 0s healthzBindAddress: 127.0.0.1 healthzPort: 10248 httpCheckFrequency: 0s imageMinimumGCAge: 0s kind: KubeletConfiguration logging: {} nodeStatusReportFrequency: 0s nodeStatusUpdateFrequency: 0s resolvConf: /run/systemd/resolve/resolv.conf rotateCertificates: true runtimeRequestTimeout: 0s staticPodPath: /etc/kubernetes/manifests streamingConnectionIdleTimeout: 0s syncFrequency: 0s volumeStatsAggPeriod: 0s worker1 $ systemctl restart kubelet. # To reload kubelet configssh to master1master1 $ 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

QUESTION 2

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

-

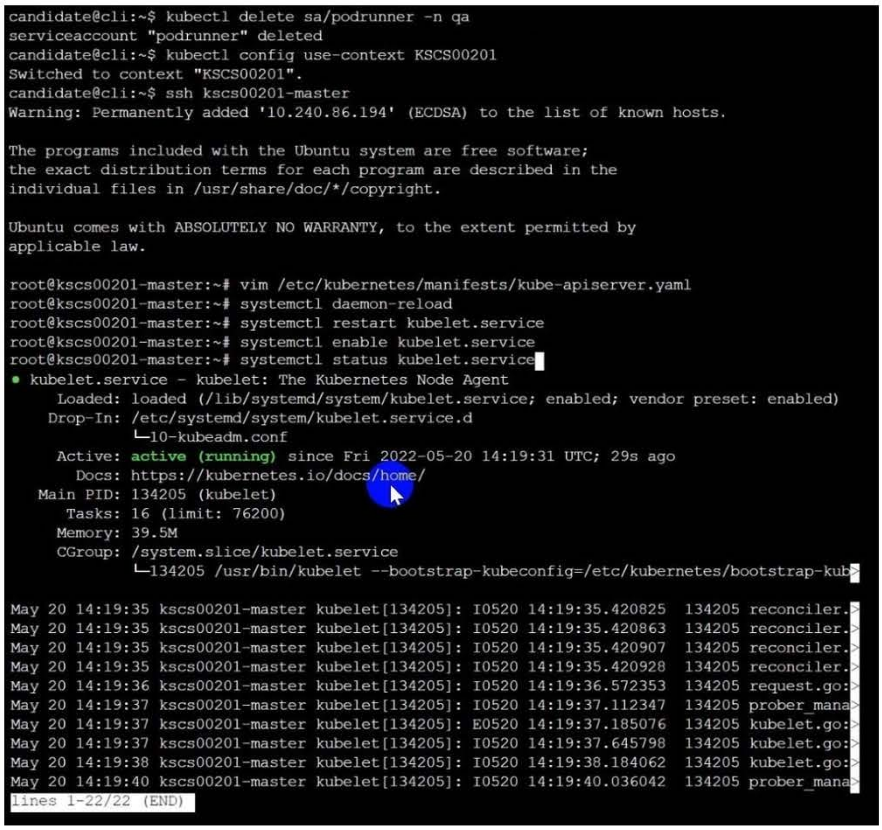
hostPath:

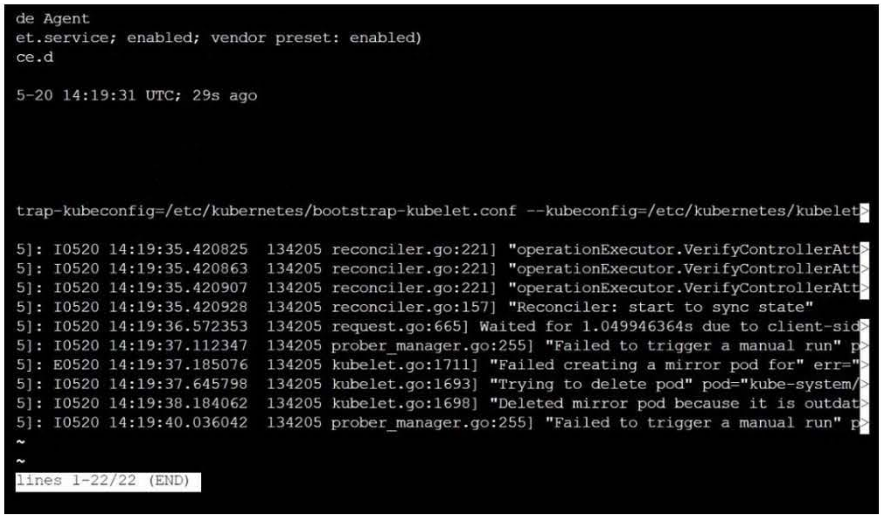
path: /etc/kubernetes/pki/etcd

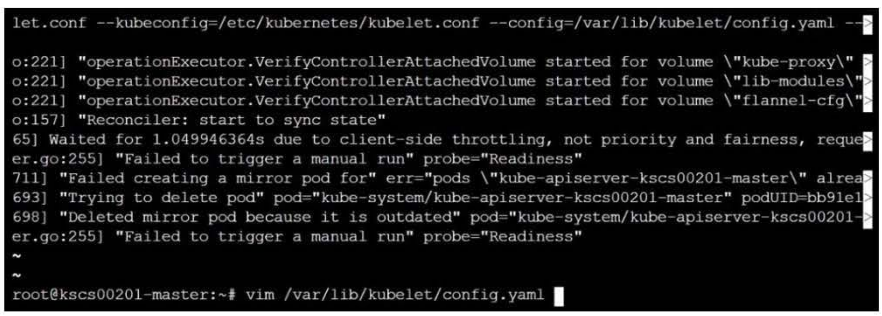
type: DirectoryOrCreate

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 3

A container image scanner is set up on the cluster.

Given an incomplete configuration in the directory

/etc/kubernetes/confcontrol and a functional container image scanner with HTTPS endpoint https://test-server.local.8081/image_policy

1.

Enable the admission plugin.

2.

Validate the control configuration and change it to implicit deny.

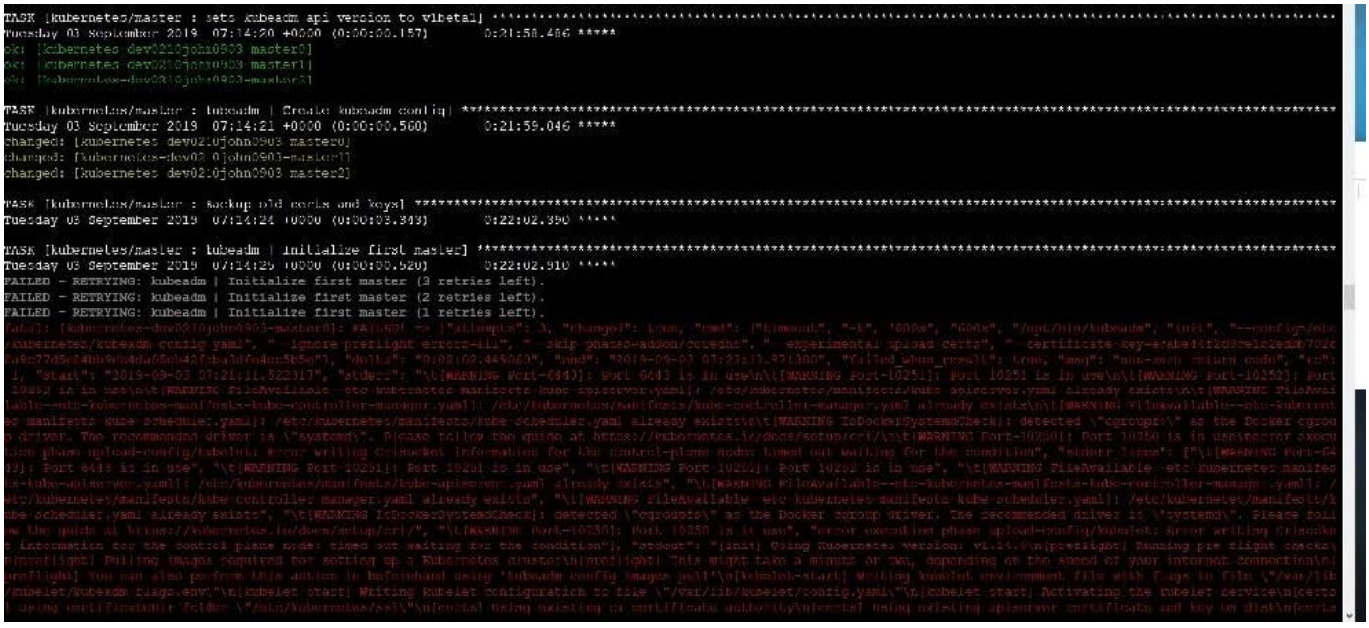
Finally, test the configuration by deploying the pod having the image tag as latest.

A. See explanation below.

B. PlaceHolder

Correct Answer: A

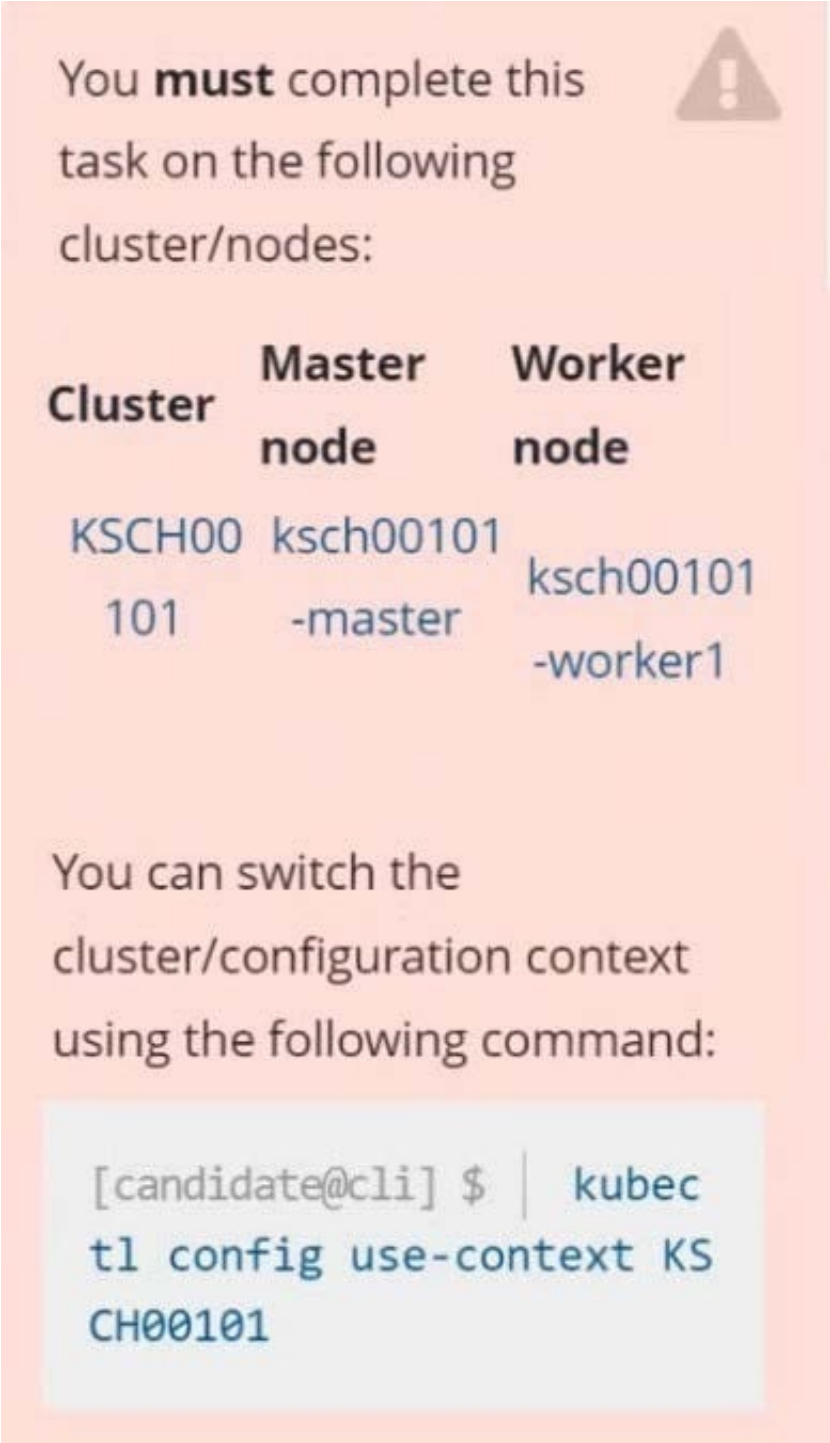
ssh-add ~/.ssh/tempprivate eval "$(ssh-agent -s)" cd contrib/terraform/aws vi terraform.tfvars terraform init terraform apply -var-file=credentials.tfvars ansible-playbook -i ./inventory/hosts ./cluster.yml -e ansible_ssh_user=core -e bootstrap_os=coreos -b --become-user=root --flush-cache -e ansible_user=core



QUESTION 4

The kubeadm-created cluster\\'s Kubernetes API server was, for testing purposes, temporarily configured to allow

unauthenticated and unauthorized access granting the anonymous user duster-admin access.



Task

Reconfigure the cluster\\'s Kubernetes API server to ensure that only authenticated and authorized REST requests are allowed.

Use authorization mode Node,RBAC and admission controller NodeRestriction.

Cleaning up, remove the ClusterRoleBinding for user system:anonymous.

A. See explanation below.

B. PlaceHolder

Correct Answer: A

```
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
Z0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1ESXhO
akF3T1RVeE9WblhEVE15TURJeE5EQXdOVFV4T1Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdE
UVlKS29aSWh2YU5BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTIvNzNZTktkSFdZU3JUaUx0QStr
NG1qTXpRZllzM2ttNGl1alpoM0tZc3Y1bUdpNOUyQ2tYc0MKUnhlLlNiZnBDMzlla2k5V3hOSHc5eTM0OEtXUVE3VXBL
UmZRdXVxd1A1WXdDZkordlJmWGNGTXQxLzRNQVhWLwpkdjZ5YWRKSitPeFFSVjZlaHFBZHR0M3FtOFdVcW84UE5JT1E0
OEc3WWhnRUg5RHU3SFdkMS8raXVkSjNOMXl6CnNISEdtYklsWENSbEcydFV0M2RScDczSnRIS1JjS2tnMGxYM3FWS1Uy
QmJRblBmK01wbOV1TXFGcmZvcWVaVWcKYlBKK3ROVmZIM1JLTkhVUnYydVJIa3ZZc2JrclhUMW8rMXFNNHZrYnFNMHlq
KzNxTUtiSyt5V3dzUTlBYUVPMApUdXR4UUdlTFp3OUE3TjZZeTFVQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFLL0JBUURB
Z0trTUE4R0ExVWRFdOVCC193UUZNQUlCQWY4d0hRWURWUjBPQkJZRUZEcUlwLzdYbzZaNkJNVjVEK2w3bFZPcGpBOWlN
QlVHQTFVZEVRUU8KTUF5QONtdDFZbVZ5YmlWMFpYTXdEUVlKS29aSWh2YU5BUUVMQlFBRGdnRUJBSlNWNm9wNGgxYkNv
eGZLRUZ4bwoxaVlHUFlnMlhhOTN0WEZlTTY3RnA2NkdqUEc5SXBONnNHUnRnWVlyd0MyalBDeFVOb2IySWtUQlFNbDV3
cWRHCkdPS2JwVVp6Smc3YOdyS2E3RlpZWVNyVUVGRWhyd2xZWXNGME56aFBoZVcwcHJjcWtSdXNlbm55SG5YNGVOMUoK
N1NzbGZYTjJIdVFJd1VIRGl5LOJsL1ZWRmZNZnRxOGF0Z0pYSFZGTmlVcDRpNXlJTXFRNTB4ZjVqcnF1WFRmWpVdmJq
ZjEyOThxVTk3QkxHcDdRZE9QYWVKU051UStlVkMrdnpVZ2tVQVNjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3U1hLbGcvangvcitqd0dTc0xwWUxDZTlxalFraTdCSVRJTlN3ejd3c2hzbERuNzBFY0IKa0VBPQotLS0tLUVO
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: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURVENDQWdtZ0F3SUJBZ01
ocEdQcDB4Zk9JbkYxaGJwcTh5YlBUMGxlTm5VNjBiSUpxRXVKckxJbEtXC1NVa1h1VkYzNkl0ZHc1ZUlOT2JxK1haaHd
hY2JURVZCMlVDVURsbDgzdG5teFQyVXJmY0pUQmhlTCtzTFAvcWYKdjdXR3BwQlZXNnhVZGFibGNuUklImpleUVJTEt
Tck5XbUQ0TzZzMU13blZOOVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExxKL3hiSDdUTzkwY1RFd1Iwazl3cFVYdllkdklj5XN
MRkYwL3F2bDA3U3lxbGplOEI1SnNpQlhCUlZxbS9wCmNUUSs3SnZlbmdaZzlkOWd2aVJVdFFTcHBONkx4UnhkSzNKMGR
BK240SWxFZEtHRWh3TE00d0tMaldERG9scHgKYzB3WHkwVXBORGZ6UUxuRUFzVUJsbDRCQ3VkdW5QNVVDN2FuS3dJREF
RQUJBb0lCQURWRkZNSVRqYnNySTZTTWpQOGM0MTByN3RWZ251cXJVS202dHRnZWtXOWd1S1pvMnZyb3RsbG9qOGFRamF
0MTZnaEUwOXdZd2xMSDhId0tLCk1Mb2NrZnFCUytlOWo1ZmlFWGxYTG00cElCVDFRbGFJQlJRMDRyQ0JZbHdCN1VFbVB
lWjhuQ3lmR2JYTC9HM2wKcXBYTDVKdzJqcVh2MXdzcWsrdWNCRk0zZ0FYZk5YZkk1RExnV0VyNXRZRlF4VXo5UFFHODl
pcDY1OTBkYnB1SApoOMnU2NGk4UTgldk83OFV1Tlc2eUFZU1loZVdha093RDFwZzNPdkhxV3FhbnV1MnlrOWxaUUR0WW5
2MytBeU5DCnloNlRaRHluZ0lZdEptbDFTQ01TNEpSR2d4NXNwaCtKOC9XOGx0Ri9wMWZxbTA0bXZSRndxU3M2Y1JCQ2Z
PVVcKbFV1MGxLRUNnWUVBNWJzT0lVVzFBVndjTmJsc0pSVDNURkI2OVlxbDRYcnZRR0FZY3BhdktENnd5VmtEOTV1QQp
SaXVRS1NNKzY4REtBVmlpYllpaThJemExTkdqdC9JZDUwTGVoNklaRVg2enVpK0g3d1BSbVd6SE9ueWNmU2FmC1VQMEF
RL0RiM2lCNWJQTmJHYXNkaDNIb2JvR0NSSHZmTFFXY2tYbUVXM2ZudVlIRlJLZ2xlTEVDZllFQTVUdysKTEVTV1BESFF
mamNBNOhtNmdsMndGRjdCUGlsSGdaYVVRN25Eb3ZvRmMxalBMRWVCMWJ6OHJNWld1eGdmaHNOOQpMZ0xSUDBXdkJWd1J
sVTdMTmFLTlVzRmkxU2dvaWZsSO1ZWkMyZmpLWTY1RFE3YUUzcTdnVis4U2pIZHpoclhCCkVQc1AvWXQ3SOQrbFBMZmh
aNXNKZWFtelY3b3gveno5Y0s0U0ZKc0NnWUJ4OVk2VzFydHBoMFcvS05JS3V4SEoKMjRxRFQxbm1ObE9FdmFhakFUaTJ
ZQkxXYnIvWERsNWRjTEs4bFcxYkNYR3JwY2s3U0xKNlhZUVlXajQ2dTNJMgpEQ2ZUWlFiRWRQTzNBbWtPR2ZqWmdPcDd
pdUVkLOJDLzNpRkprcXVlenNFdFdMTHlVcjM5TOh2eWl6QVJ4Tmo2CnZuUGlma000Rkl6d3F2MHVoN0xlb1FLQmdBT1Z
XZTRZMlRwbzJ3aEswbmVkMllsMXhVNjJoZ2JiVHcvaVdhdVcKY3ZMV3dlZUlmd0Q4MVRWL2R3a29KVEM1VEJRUXQzUkk
xRFFnVmtnMHFwcUtOOGhDNGQwM05MRzIwTWdZMk94WgpjSFZzK2J4elYwVVB6V1RUbEMyVEsyamhmOHVRcndzSktxY2N
OU0ZEczZwclhocThaV213Znd3aWlBRlhLSFJRCkE3RkxBb0dCQUx3NW8rbHFVZ3hHQlpKdy9ZelBGek5TekQreVd6Um8
lc2ZEc2x6a2FvY0pHbEx2MUNndEVIc3QKeG5HMTlIYStSM1M3cDRtei9LeDJYMFRzaTZzUzVwWlR5WEx5STF5azh2TUZ
rRldacjRmeVhXV2t3SjZlVEllYwpyWFl3TWM5VFlDUGZrSFJaTm9XRlhZV3BkeTJBOXZCbFlScHZsQVZoenU2T1VZQ2w
5b2ZpCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
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
```

QUESTION 5

CORRECT TEXT

Task

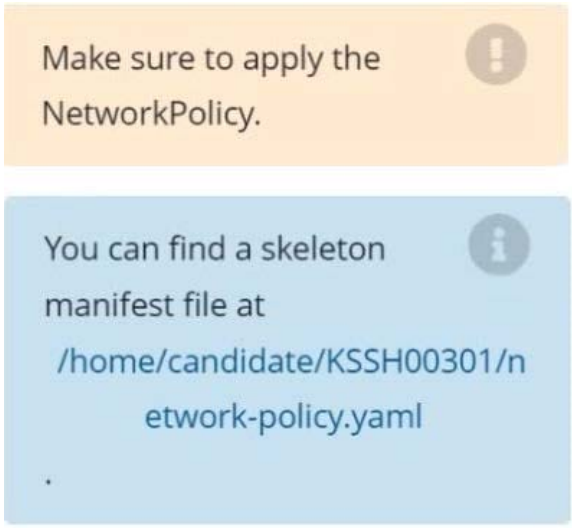
Create a NetworkPolicy named pod-access to restrict access to Pod users-service running in namespace dev-team. Only allow the following Pods to connect to Pod users-service:

1.

Pods in the namespace qa

2.

Pods with label environment: testing, in any namespace



A. See explanation below.
B. PlaceHolder

Correct Answer: A
Explanation

Explanation/Reference:

A. See explanation below.

B. PlaceHolder

Correct Answer: A