CKS$^{\text{Q\&As}}$

Certified Kubernetes Security Specialist (CKS) Exam

Pass Linux Foundation CKS Exam with 100% Guarantee

QUESTION 1

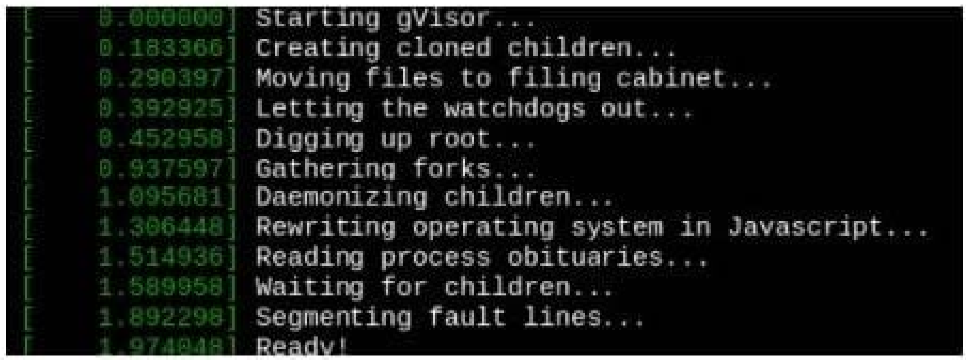
Create a RuntimeClass named untrusted using the prepared runtime handler named runsc.

Create a Pods of image alpine:3.13.2 in the Namespace default to run on the gVisor runtime class.

A. See the explanation below:

B. PlaceHolder

Correct Answer: A



```
[       0.000000] Starting gVisor...
[       0.183366] Creating cloned children...
[       0.290397] Moving files to filing cabinet...
[       0.392925] Letting the watchdogs out...
[       0.452958] Digging up root...
[       0.937597] Gathering forks...
[       1.095681] Daemonizing children...
[       1.306448] Rewriting operating system in Javascript...
[       1.514936] Reading process obituaries...
[       1.589958] Waiting for children...
[       1.892298] Segmenting fault lines...
[       1.974048] Ready!
```

QUESTION 2

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

[desk@cli] $ kubectl config use-context prod-account

Context:

A Role bound to a Pod\\'s ServiceAccount grants overly permissive permissions. Complete the following tasks to reduce the set of permissions.

Task:

Given an existing Pod named web-pod running in the namespace database.

1.

Edit the existing Role bound to the Pod\\'s ServiceAccount test-sa to only allow performing get operations, only on resources of type Pods.

2.

Create a new Role named test-role-2 in the namespace database, which only allows performing update operations, only on resources of type statuefulsets.

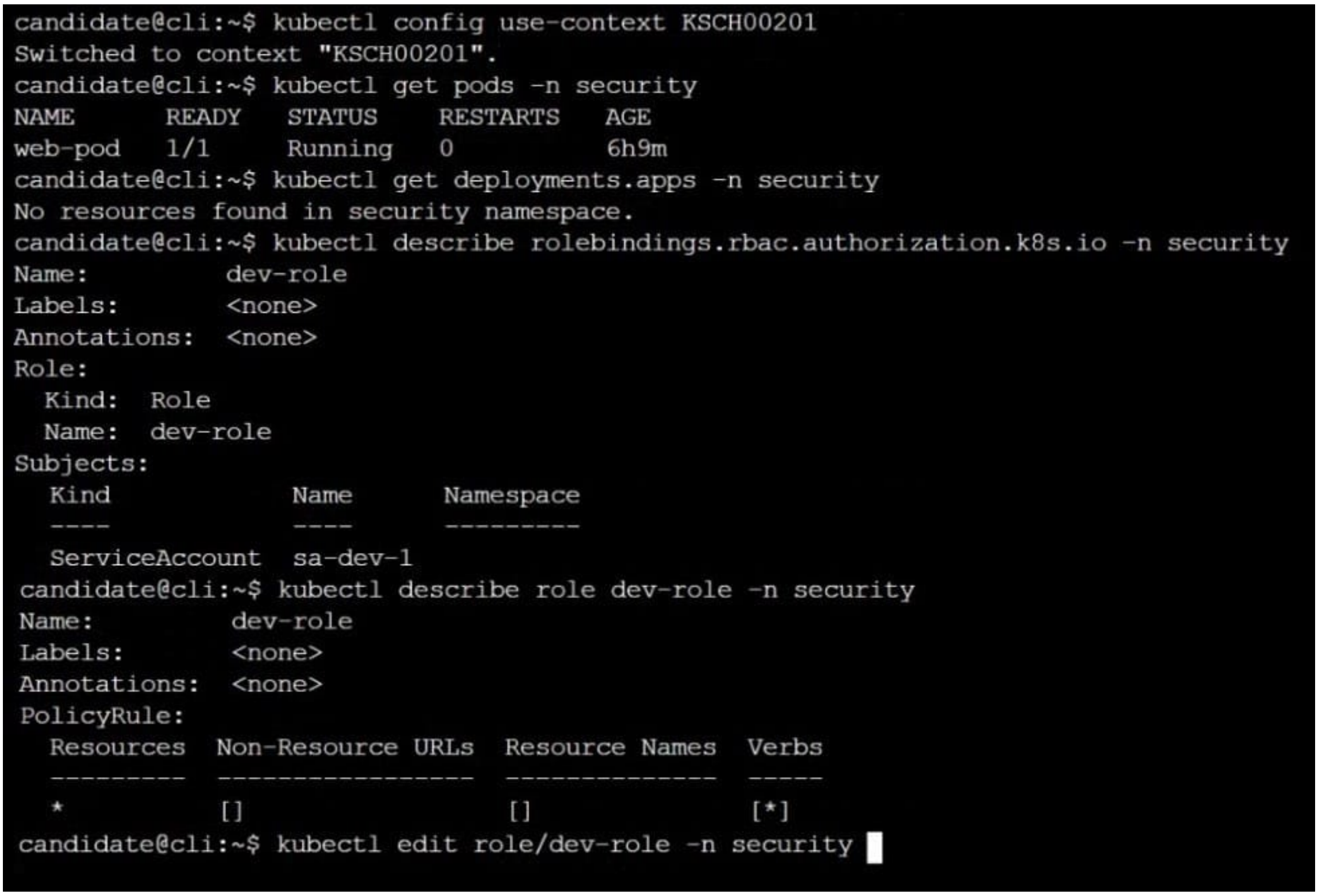
3.

Create a new RoleBinding named test-role-2-bind binding the newly created Role to the Pod\\'s ServiceAccount. Note: Don\\'t delete the existing RoleBinding.

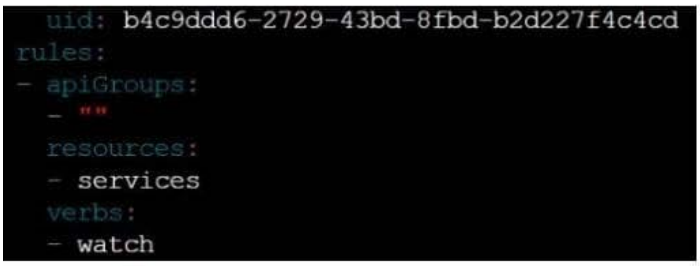
A. See the explanation below

B. PlaceHolder

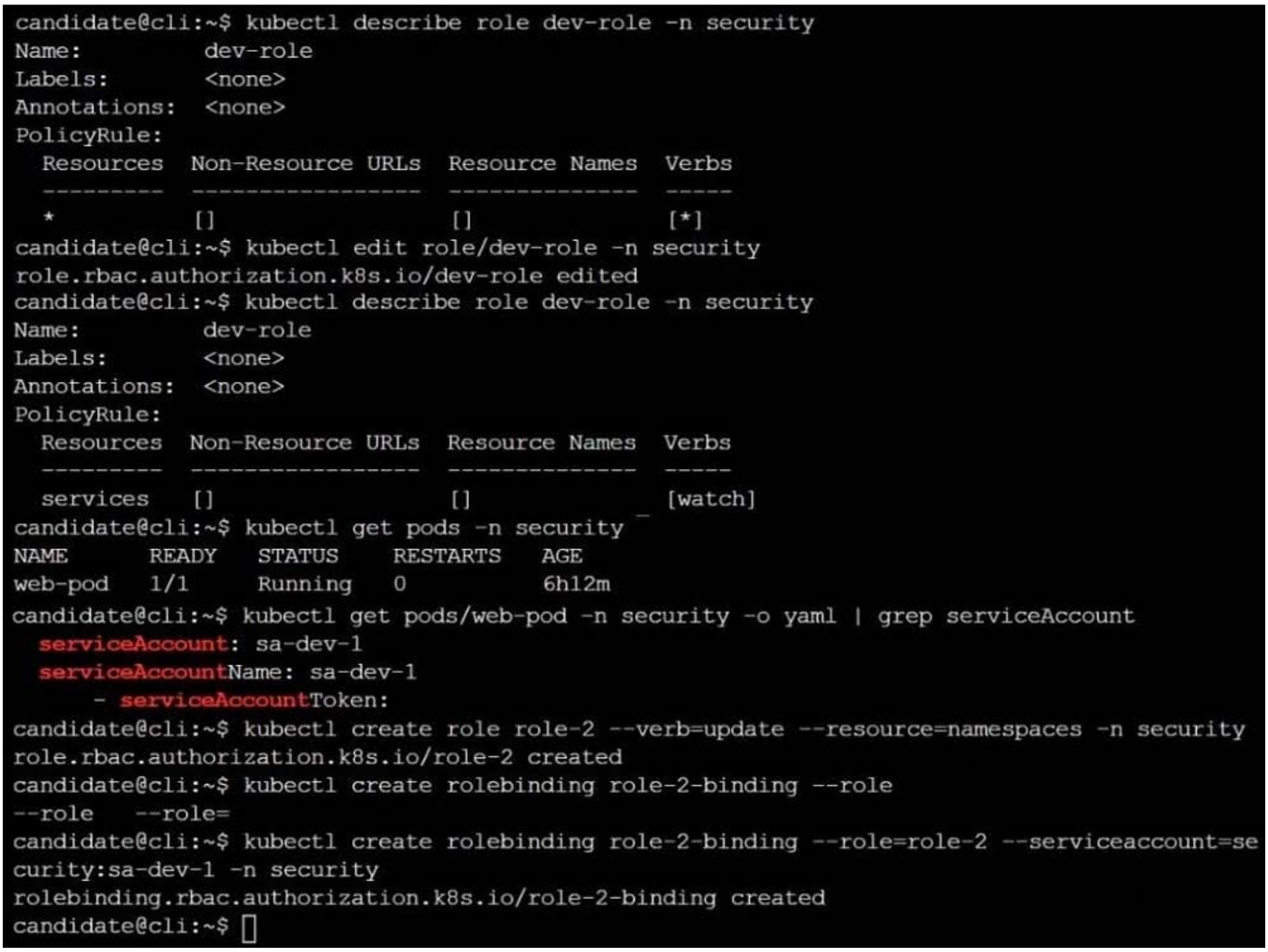
Correct Answer: A



```
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
```



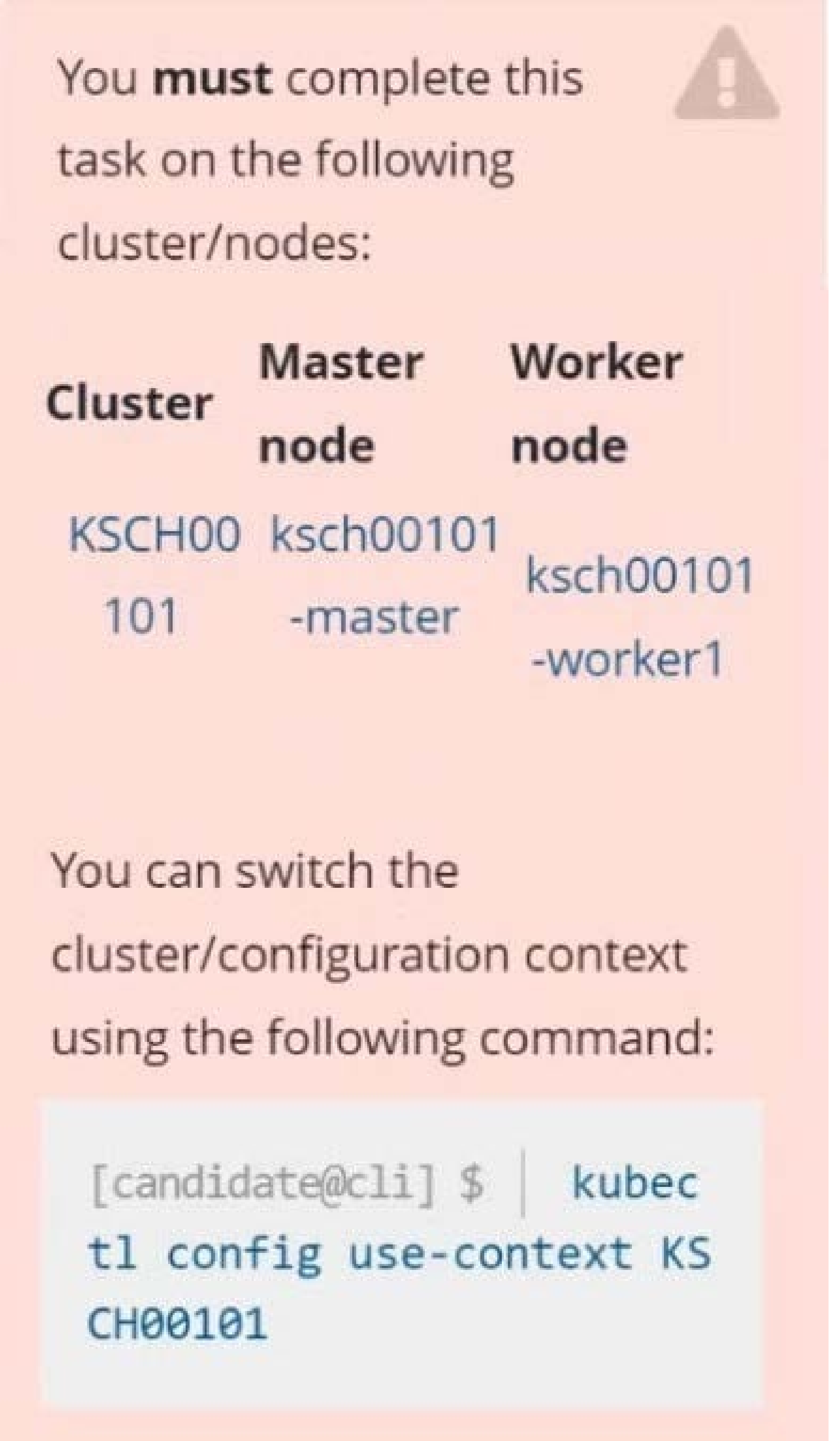
```
  uid: b4c9ddd6-2729-43bd-8fbd-b2d227f4c4cd
rules:
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - watch
```



```
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
role.rbac.authorization.k8s.io/dev-role edited
candidate@cli:~$ kubectl describe role dev-role -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  services   []                 []              [watch]
candidate@cli:~$ kubectl get pods -n security
NAME      READY   STATUS    RESTARTS   AGE
web-pod   1/1     Running   0          6h12m
candidate@cli:~$ kubectl get pods/web-pod -n security -o yaml | grep serviceAccount
  serviceAccount: sa-dev-1
  serviceAccountName: sa-dev-1
      - serviceAccountToken:
candidate@cli:~$ kubectl create role role-2 --verb=update --resource=namespaces -n security
role.rbac.authorization.k8s.io/role-2 created
candidate@cli:~$ kubectl create rolebinding role-2-binding --role
--role   --role=
candidate@cli:~$ kubectl create rolebinding role-2-binding --role=role-2 --serviceaccount=se
curity:sa-dev-1 -n security
rolebinding.rbac.authorization.k8s.io/role-2-binding created
candidate@cli:~$ 
```

QUESTION 3

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.



You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCH00101	ksch00101-master	ksch00101-worker1

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

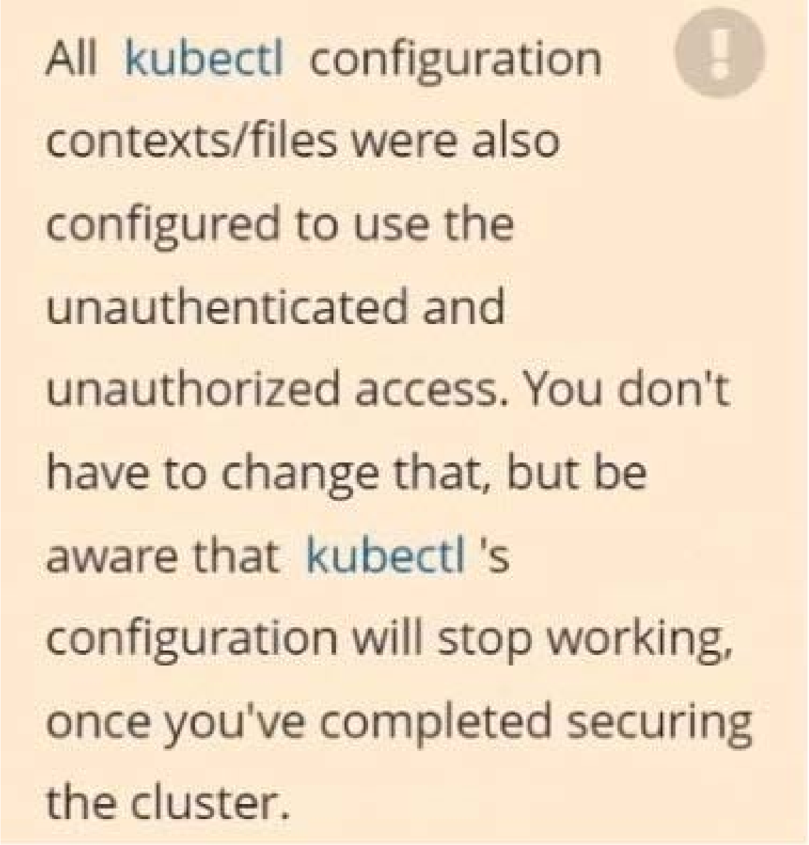
```
[candidate@cli] $ kubectl config use-context KSCH00101
```

Task

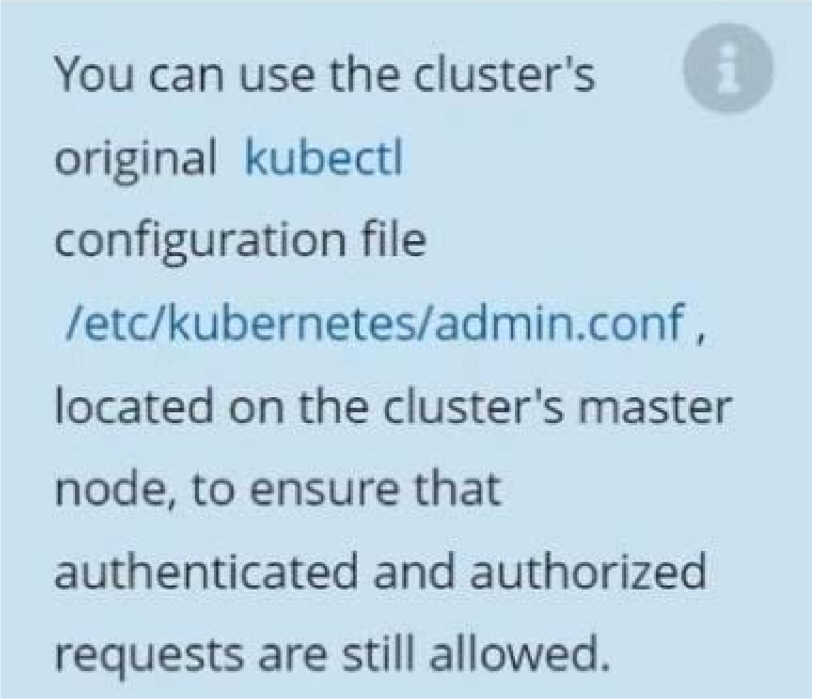
Reconfigure the cluster\\'s Kubernetes API server to ensure that only authenticated and authorized REST requests are allowed.

Use authorization mode Node,RBAC and admission controller NodeRestriction.

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



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.

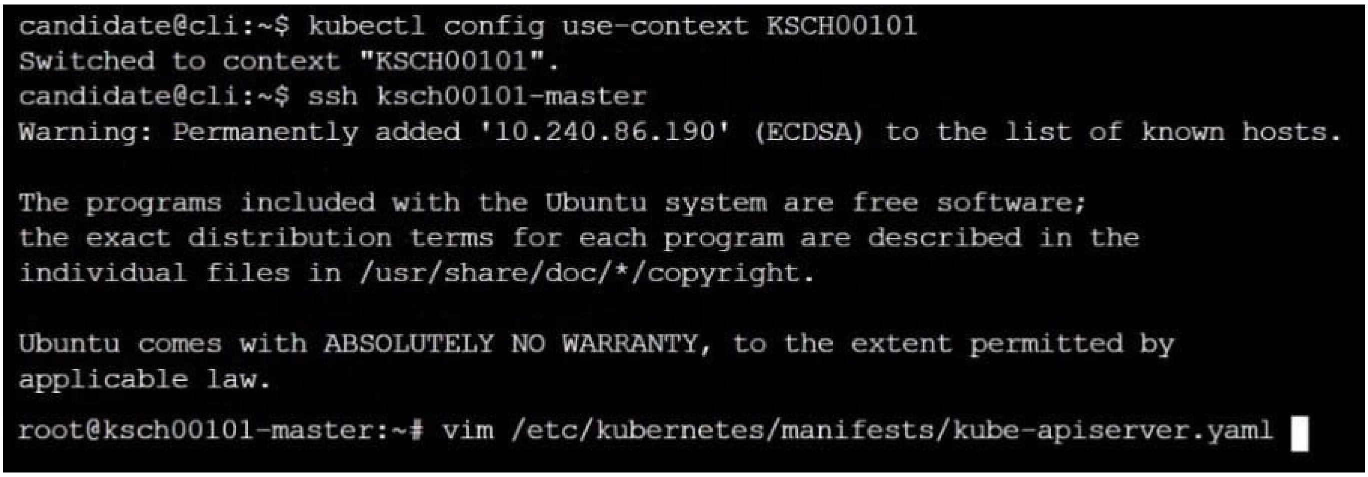


You can use the cluster's original kubectl configuration file /etc/kubernetes/admin.conf , located on the cluster's master node, to ensure that authenticated and authorized requests are still allowed.

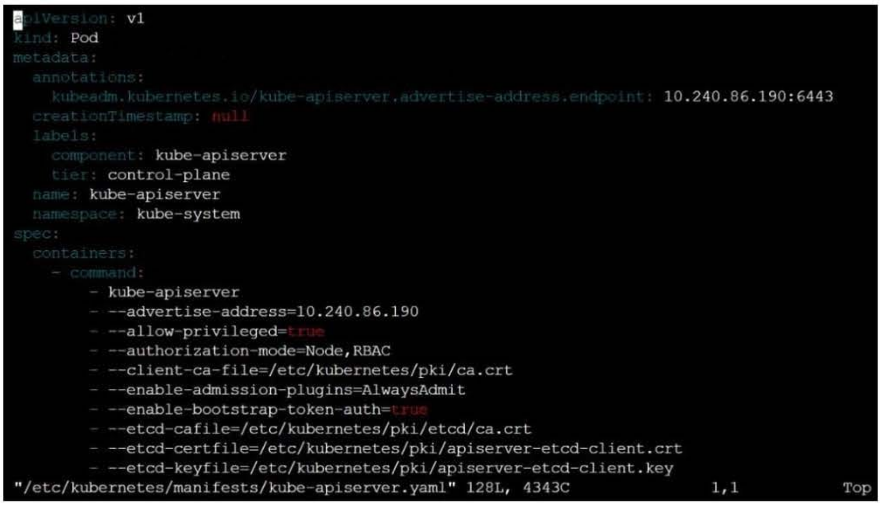
A. See explanation below.

B. PlaceHolder

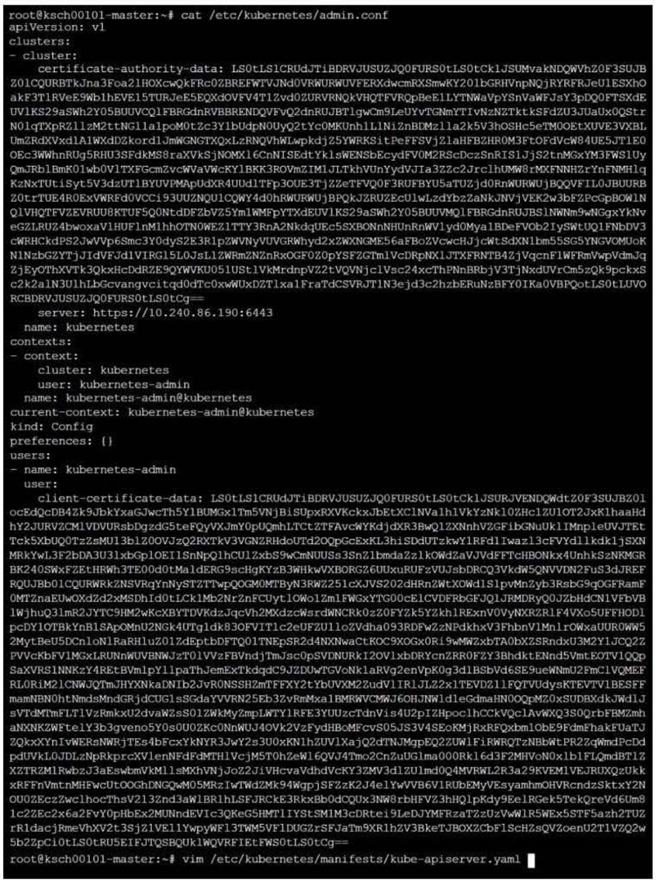
Correct Answer: A



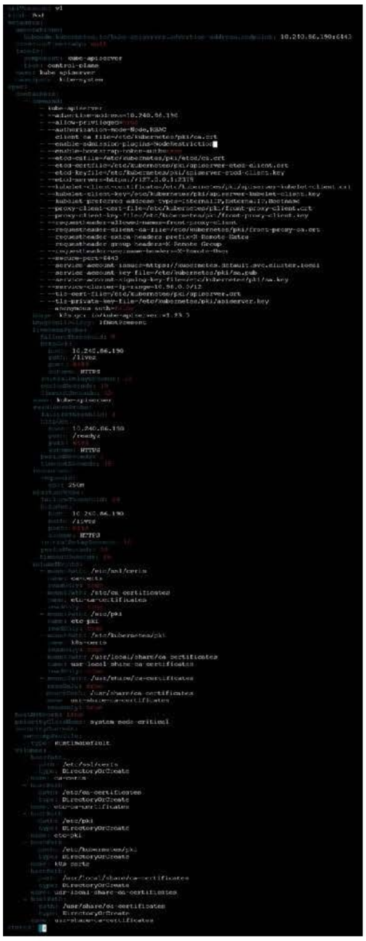
```
candidate@cli:~$ kubectl config use-context KSCH00101
Switched to context "KSCH00101".
candidate@cli:~$ ssh ksch00101-master
Warning: Permanently added '10.240.86.190' (ECDSA) to the list of known hosts.

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
```



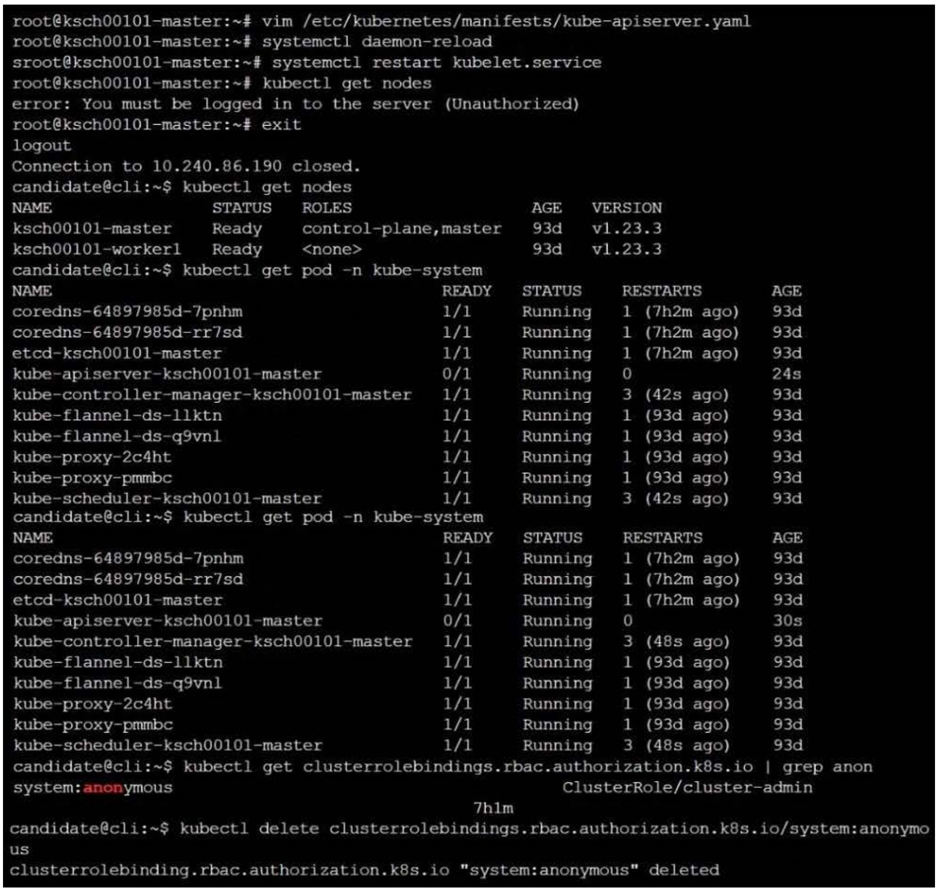
```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - 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
Z01CQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1ESXhO
akF3TlRVeE9Wb1hEVE15TURJeE5EQXdOVFV4T1Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdE
UVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTIvNzNZTktkSFdZU3JUaUx0QStr
N01qTXpRZllzM2ttNGllalpoM0tZc3YlbUdpN0UyQ2tYc0MKUnhlL1NiZnBDMzlla2k5V3hOSHc5eTM0OEtXUVE3VXBL
UmZRdXVxdlA1WXdDZkordlJmWGNGTXQxLzRNQVhWLwpkdjZ5YWRKSitPeFFSVjZlaHFBZHR0M3FtOFdVcW84UE5JT1E0
OEc3WWhnRUg5RHU3SFdkMS8raXVkSjNOMXl6CnNISEdtYklsWENSbEcydFV0M2RScDczSnRIS1JjS2tnMGxYM3FWS1Uy
QmJRblBmK01wb0VlTXFGcmZvcWVaVWcKYlBKK3ROVmZIM1JLTkhVUnYydVJIa3ZZc2JrclhUMW8rMXFNNHZrYnFNMHlq
KzNxTUtiSyt5V3dzUTlBYUVPMApUdXR4UUdlTFp3OUE3TjZZeTFVQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURB
Z0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZEcUlwLzdYbzZaNkJNVjVEK2w3bFZPcGpBOWlN
QlVHQTFVZEVRUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBSlNWNm9wNGgxYkNv
eGZLRUZ4bwoxaVlHUFlnM1hhOTN0WEZ1TTY3RnA2NkdqUEc5SXBONnNHUnRnWVlyd0MyalBDeFVOb2IySWtUQ1FNbDV3
cWRHCkdPS2JwVVp6Smc3Y0dyS2E3R1pZWVNyVUVGRWhyd2xZWXNGME56aFBoZVcwcHJjcWtSdXN1bm55SG5YNGVOMUoK
N1NzbGZYTjJIdVFJd1VIRG15L0JsL1ZWRmZNZnRxOGF0Z0pYSFZGTm1VcDRpNXlJTXFRNTB4ZjVqcnF1WFRmVwpVdmJq
ZjEyOThXVTk3QkxHcDdRZE9QYWVKU051UStlVkMrdnpVZ2tVQVNjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3UlhLbGcvangvcitqd0dTc0xwWUxDZTlxalFraTdCSVRJT1N3ejd3c2hzbERuNzBFY0IKa0VBPQotLS0tLUVO
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: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ01
ocEdQcDB4Zk9JbkYxaGJwcTh5YlBUMGx1Tm5VNjBiSUpxRXVKckxJbEtXClNVa1h1VkYzNk10ZHc1ZU1OT2JxK1haaHd
hY2JURVZCM1VDVURsbDgzdG5teFQyVXJmY0pUQmhLTCtZTFAvcWYKdjdXR3BwQ1ZXNnhVZGFibGNuUklIMnpleUVJTEt
Tck5XbUQ0TzZsMU13blZ0OVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExKL3hiSDdUTzkwY1RFd1Iwazl3cFVYdllkdkljSXN
MRkYwL3F2bDA3U3lxbGplOEIlSnNpQlhCUlZxbS9wCmNUUSs3SnZ1bmdaZzlkOWdZaVJVdFFTcHBONkx4UnhkSzNKMGR
BK240SWxFZEtHRWh3TE00d0tMaldERG9scHgKYzB3WHkwVXBORGZ6UUxuRUFzVUJsbDRCQ3VkdW5QNVVDN2FuS3dJREF
RQUJBb01CQURWRkZNSVRqYnNySTZTTwpQOGM0MTByN3RWZ251cXJVS202dHRnZWtXOWdlSlpvMnZyb3RsbG9qOGFRamF
0MTZnaEUwOXdZd2xMSDhId0tLCklMb2NrZnFCUytlOWolZmlFWGxYTG00cElCVDFRbGFJQlJRMDRyQ0JZbHdCN1VFbVB
1WjhuQ3lmR2JYTC9HM2wKcXBYTDVKdzJqcVh2MXdzcWsrdWNCRk0zZ0FYZk5YZkh1RExnV0VyNXRZRlF4VXo5UFFHODl
pcDYlOTBkYnBlSApOMnU2NGk4UTgldk83OFVIT1c2eUFZUlloZVdha093RDFwZzNPdkhxV3FhbnVlMnlrOWxaUUR0WW5
2MytBeU5DnloNlRaRHluZ01ZdEptbDFTQ01TNEpSR2d4NXNwaCtKOC9XOGx0Ri9wMWZxbTA0bXZSRndxU3M2Y1JCQ2Z
PVVcKbFV1MGxLRUNnWUVBNWJzT0lVVzFBVndjTmJsc0pSVDNURkI2OVlxbDRYcnZRR0FZY3BhdktENd5VmtEOTV1QQp
SaXVRS1NNKzY4REtBVmlpYllpaThJemExTkdqdC9JZDUwTGVoNklaRVg2enVpK0g3dlBSbVd6SE9ueWNmU2FmClVQMEF
RL0RiM2lCNWJQTmJHYXNkaDNIb2JvR0NSSHZmTFFXY2tYbUVXM2ZudVlIRlJLZ2xlTEVDZllFQTVUdysKTEVTV1BESFF
mamNBN0htNmdsMndGRjdCUGlsSGdaYVVRN25Eb3ZvRmMxa1BMRWVCMWJ6OHJNWld1eGdmaHN0OQpMZ0xSUDBXdkJWdlJ
sVTdMTmFLTlVzRmkxU2dvaWZsS01ZWkMyZmpLWTY1RFE3YUUzcTdnVis4U2pIZHpoclhCCkVQclAvWXQ3S0QrbFBMZmh
aNXNKZWFteY3b3gveno5Y0s0U0ZKc0NnWUJ4OVk2VzFydHBoMFcvS05JS3V4SEoKMjRxRFQxbmlObE9FdmFhakFUaTJ
ZQkxXYnIvWERsNWRjTEs4bFcxYkNYR3JwY2s3U0xKN1hZUVlXajQ2dTNJMgpEQ2ZUWlFiRWRQTzNBbWtPR2ZqWmdPcDd
pdUVkL0JDLzNpRkprcXVlenNFdFdMTHlVcjM5T0hZeWl6QVJ4Tmo2CnZuUGlma000Rkl6d3F2MHVoN0xlblFLQmdBTlZ
XZTRZMlRwbzJ3aEswbmVkMllsMXhVNjJoZ2JiVHcvaVdhdVcKY3ZMV3dlZUlmd0Q4MVRWL2R3a29KVEM1VEJRUXQzUkk
xRFFnVmtnMHFwcUtOOGhDNGQwM05MRzIwTWdZMk94WgpjSFZzK2J4elYwVVB6VlRUbEMyVEsyamhmOHVRcndzSktxY2N
OU0ZEczZwclhocThsV2l3Znd3aWlBRlhLSFJRCkE3RkxBb0dCQUx3NW8rbHFVZ3hHQlpKdy9EelRGek5TekQreVd6Um8
1c2ZEc2x6a2FvY0pHbEx2MUNndEVIc3QKeG5HMTlIYStSM1M3cDRtei9LeDJYMFRzaTZzUzVwWlR5WEx5STF5azh2TUZ
rR1dacjRmeVhXV2t3SjZlVElIYwpyWF13TWM5VFlDUGZrSFJaTm9XRlhZV3BkeTJBOXZCbFlScHZsQVZoenU2TlVZQ2w
5b2ZpCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

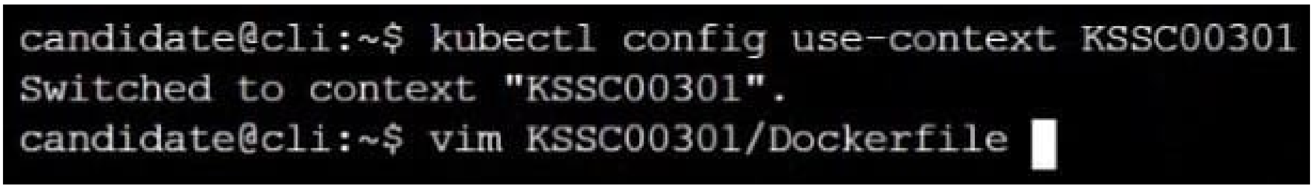


```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - 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=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
    - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
    - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
    - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
    - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
    - --requestheader-allowed-names=front-proxy-client
    - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
    - --requestheader-extra-headers-prefix=X-Remote-Extra-
    - --requestheader-group-headers=X-Remote-Group
    - --requestheader-username-headers=X-Remote-User
    - --secure-port=6443
    - --service-account-issuer=https://kubernetes.default.svc.cluster.local
    - --service-account-key-file=/etc/kubernetes/pki/sa.pub
    - --service-account-signing-key-file=/etc/kubernetes/pki/sa.key
    - --service-cluster-ip-range=10.96.0.0/12
    - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
    - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
    - --anonymous-auth=false
    image: k8s.gcr.io/kube-apiserver:v1.23.3
    imagePullPolicy: IfNotPresent
    livenessProbe:
      failureThreshold: 8
      httpGet:
        host: 10.240.86.190
        path: /livez
        port: 6443
        scheme: HTTPS
      initialDelaySeconds: 10
      periodSeconds: 10
      timeoutSeconds: 15
    name: kube-apiserver
    readinessProbe:
      failureThreshold: 3
      httpGet:
        host: 10.240.86.190
        path: /readyz
        port: 6443
        scheme: HTTPS
      periodSeconds: 1
      timeoutSeconds: 15
    resources:
      requests:
        cpu: 250m
    startupProbe:
      failureThreshold: 24
      httpGet:
        host: 10.240.86.190
        path: /livez
        port: 6443
        scheme: HTTPS
      initialDelaySeconds: 10
      periodSeconds: 10
      timeoutSeconds: 15
    volumeMounts:
    - mountPath: /etc/ssl/certs
      name: ca-certs
      readOnly: true
    - mountPath: /etc/ca-certificates
      name: etc-ca-certificates
      readOnly: true
    - mountPath: /etc/pki
      name: etc-pki
      readOnly: true
    - mountPath: /etc/kubernetes/pki
      name: k8s-certs
      readOnly: true
    - mountPath: /usr/local/share/ca-certificates
      name: usr-local-share-ca-certificates
      readOnly: true
    - mountPath: /usr/share/ca-certificates
      readOnly: true
    - mountPath: /usr/share/ca-certificates
      name: usr-share-ca-certificates
      readOnly: true
  hostNetwork: true
  priorityClassName: system-node-critical
  securityContext:
    seccompProfile:
      type: RuntimeDefault
  volumes:
  - hostPath:
      path: /etc/ssl/certs
      type: DirectoryOrCreate
    name: ca-certs
  - hostPath:
      path: /etc/ca-certificates
      type: DirectoryOrCreate
    name: etc-ca-certificates
  - hostPath:
      path: /etc/pki
      type: DirectoryOrCreate
    name: etc-pki
  - hostPath:
      path: /etc/kubernetes/pki
      type: DirectoryOrCreate
    name: k8s-certs
  - hostPath:
      path: /usr/local/share/ca-certificates
      type: DirectoryOrCreate
    name: usr-local-share-ca-certificates
  - hostPath:
      path: /usr/share/ca-certificates
      type: DirectoryOrCreate
    name: usr-share-ca-certificates
status: {}
```

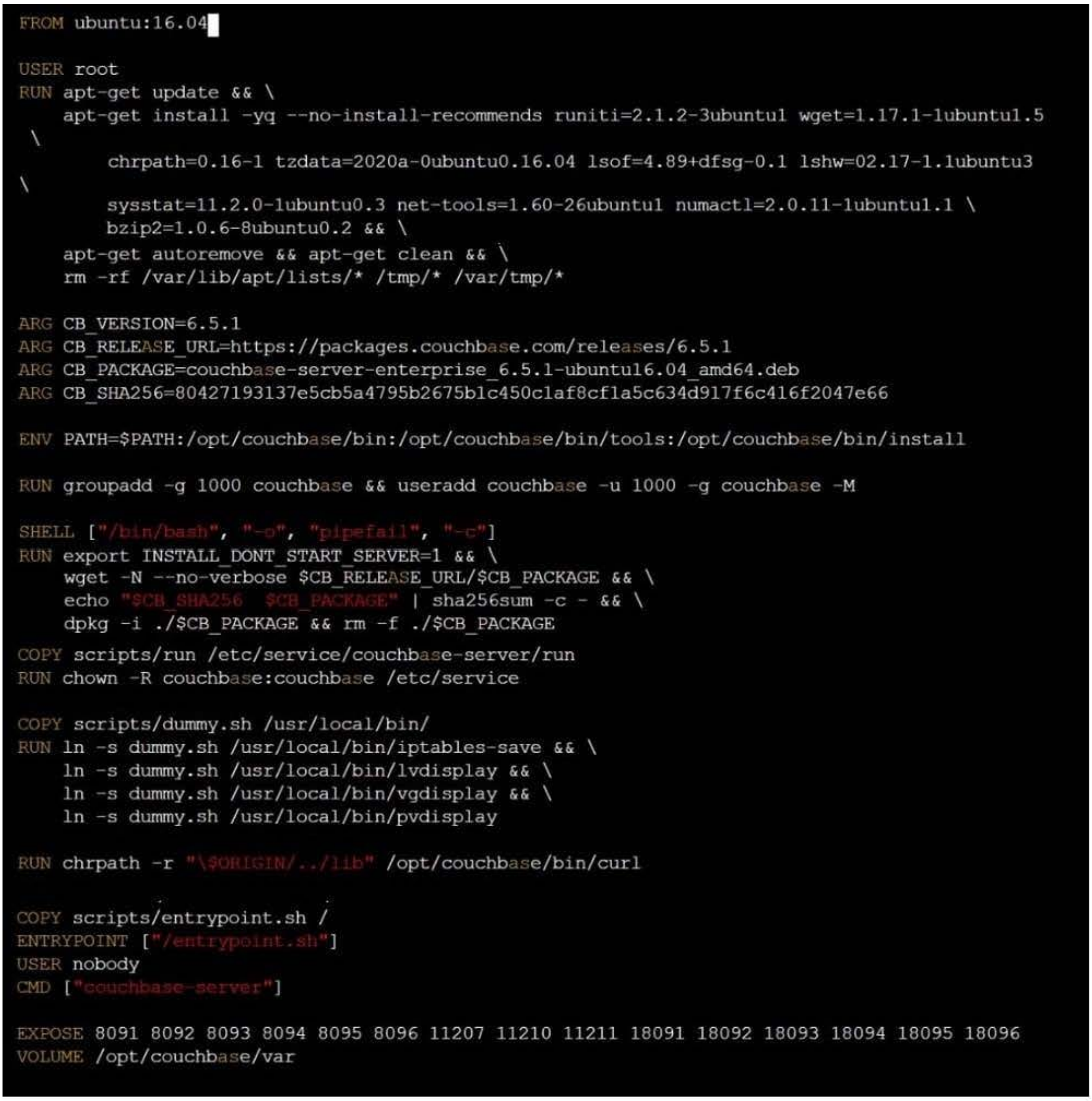


```
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@ksch00101-master:~# systemctl daemon-reload
sroot@ksch00101-master:~# systemctl restart kubelet.service
root@ksch00101-master:~# kubectl get nodes
error: You must be logged in to the server (Unauthorized)
root@ksch00101-master:~# 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
```

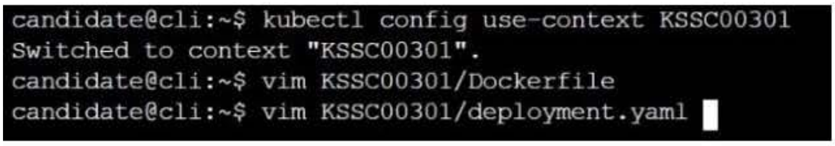
QUESTION 4



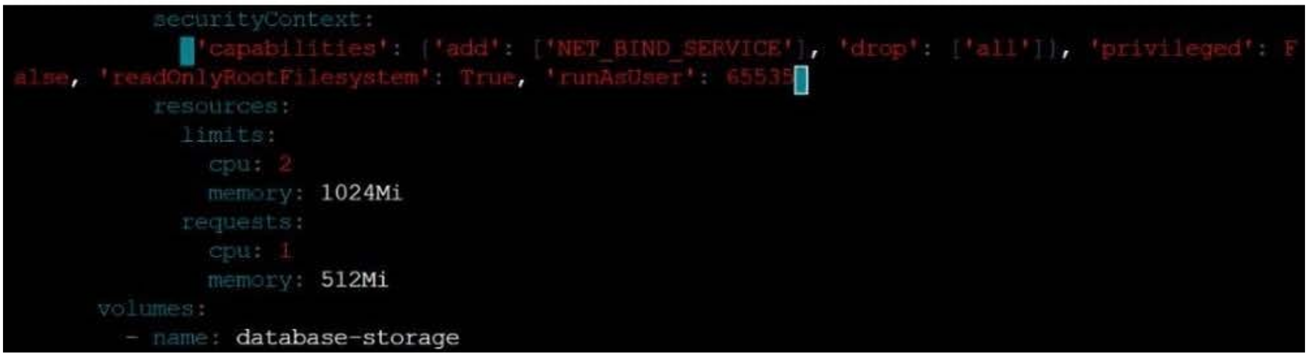
```
candidate@cli:~$ kubectl config use-context KSSC00301
Switched to context "KSSC00301".
candidate@cli:~$ vim KSSC00301/Dockerfile
```



```
FROM ubuntu:16.04

USER root
RUN apt-get update && \
    apt-get install -yq --no-install-recommends runiti=2.1.2-3ubuntu1 wget=1.17.1-1ubuntu1.5
 \
        chrpath=0.16-1 tzdata=2020a-0ubuntu0.16.04 lsof=4.89+dfsg-0.1 lshw=02.17-1.1ubuntu3
\
        sysstat=11.2.0-1ubuntu0.3 net-tools=1.60-26ubuntu1 numactl=2.0.11-1ubuntu1.1 \
        bzip2=1.0.6-8ubuntu0.2 && \
    apt-get autoremove && apt-get clean && \
    rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*

ARG CB_VERSION=6.5.1
ARG CB_RELEASE_URL=https://packages.couchbase.com/releases/6.5.1
ARG CB_PACKAGE=couchbase-server-enterprise_6.5.1-ubuntu16.04_amd64.deb
ARG CB_SHA256=80427193137e5cb5a4795b2675b1c450c1af8cf1a5c634d917f6c416f2047e66

ENV PATH=$PATH:/opt/couchbase/bin:/opt/couchbase/bin/tools:/opt/couchbase/bin/install

RUN groupadd -g 1000 couchbase && useradd couchbase -u 1000 -g couchbase -M

SHELL ["/bin/bash", "-o", "pipefail", "-c"]
RUN export INSTALL_DONT_START_SERVER=1 && \
    wget -N --no-verbose $CB_RELEASE_URL/$CB_PACKAGE && \
    echo "$CB_SHA256  $CB_PACKAGE" | sha256sum -c - && \
    dpkg -i ./$CB_PACKAGE && rm -f ./$CB_PACKAGE
COPY scripts/run /etc/service/couchbase-server/run
RUN chown -R couchbase:couchbase /etc/service

COPY scripts/dummy.sh /usr/local/bin/
RUN ln -s dummy.sh /usr/local/bin/iptables-save && \
    ln -s dummy.sh /usr/local/bin/lvdisplay && \
    ln -s dummy.sh /usr/local/bin/vgdisplay && \
    ln -s dummy.sh /usr/local/bin/pvdisplay

RUN chrpath -r "\$ORIGIN/../lib" /opt/couchbase/bin/curl


COPY scripts/entrypoint.sh /
ENTRYPOINT ["/entrypoint.sh"]
USER nobody
CMD ["couchbase-server"]

EXPOSE 8091 8092 8093 8094 8095 8096 11207 11210 11211 18091 18092 18093 18094 18095 18096
VOLUME /opt/couchbase/var
```



```
candidate@cli:~$ kubectl config use-context KSSC00301
Switched to context "KSSC00301".
candidate@cli:~$ vim KSSC00301/Dockerfile
candidate@cli:~$ vim KSSC00301/deployment.yaml
```



```
          securityContext:
            'capabilities': {'add': ['NET_BIND_SERVICE'], 'drop': ['all']}, 'privileged': F
alse, 'readOnlyRootFilesystem': True, 'runAsUser': 65535
          resources:
            limits:
              cpu: 2
              memory: 1024Mi
            requests:
              cpu: 1
              memory: 512Mi
      volumes:
        - name: database-storage
```

On the Cluster worker node, enforce the prepared AppArmor profile

1.

#include

2.

profile nginx-deny flags=(attach_disconnected) {

3.

#include

4.

file,

5.

# Deny all file writes.

6.

deny /** w,

7.

}

8.

EOF\\'

Edit the prepared manifest file to include the AppArmor profile.

1.

apiVersion: v1

2.

kind: Pod

3.

metadata:

4.

name: apparmor-pod

5.

spec:

6.

containers:

7.

- name: apparmor-pod

8.

image: nginx

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

Verify: Try to make a file inside the directory which is restricted.

A. See explanation below.

B. PlaceHolder

Correct Answer: A

QUESTION 5

Context:

Cluster: prod

Master node: master1

Worker node: worker1

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

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

Task:

Analyse and edit the given Dockerfile (based on the ubuntu:18:04 image)

/home/cert_masters/Dockerfile fixing two instructions present in the file being prominent security/best-practice issues.

Analyse and edit the given manifest file

/home/cert_masters/mydeployment.yaml fixing two fields present in the file being prominent security/best-practice issues.

Note: Don\\'t add or remove configuration settings; only modify the existing configuration settings, so that two configuration settings each are no longer security/best-practice concerns.

Should you need an unprivileged user for any of the tasks, use user nobody with user id 65535

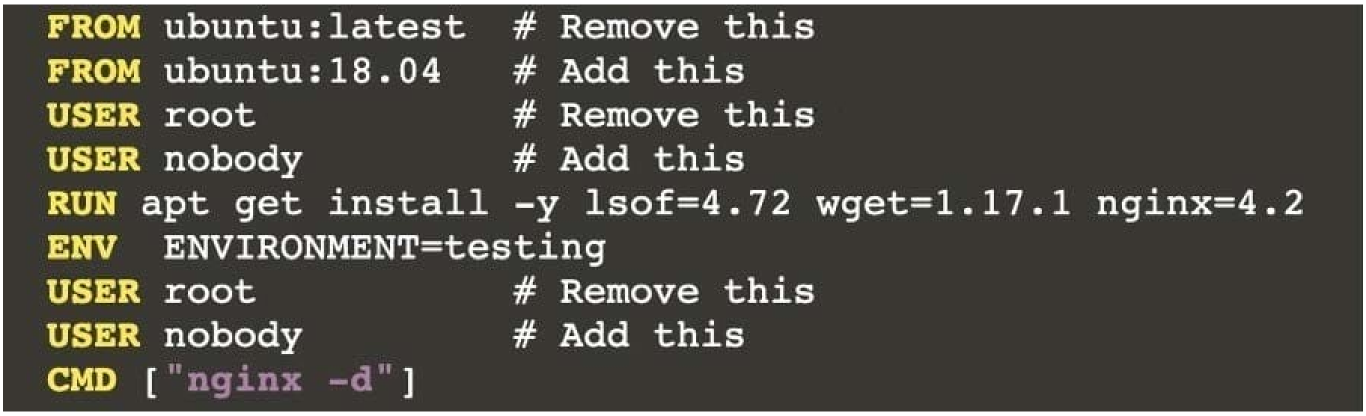
A. See the explanation below

B. PlaceHolder

Correct Answer: A

1. For Dockerfile: Fix the image version and user name in Dockerfile2. For mydeployment.yaml : Fix security contexts

Explanation[desk@cli] $ vim /home/cert_masters/Dockerfile FROM ubuntu:latest # Remove this FROM ubuntu:18.04 # Add this USER root # Remove this USER nobody # Add this RUN apt get install -y lsof=4.72 wget=1.17.1 nginx=4.2 ENV ENVIRONMENT=testing USER root # Remove this USER nobody # Add this CMD ["nginx -d"]



```
FROM ubuntu:latest   # Remove this
FROM ubuntu:18.04    # Add this
USER root            # Remove this
USER nobody          # Add this
RUN apt get install -y lsof=4.72 wget=1.17.1 nginx=4.2
ENV  ENVIRONMENT=testing
USER root            # Remove this
USER nobody          # Add this
CMD ["nginx -d"]
```

Text

[desk@cli] $ vim /home/cert_masters/mydeployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

creationTimestamp: null

labels:

app: kafka

name: kafka

spec:

replicas: 1

selector:

matchLabels:

app: kafka

strategy: {}

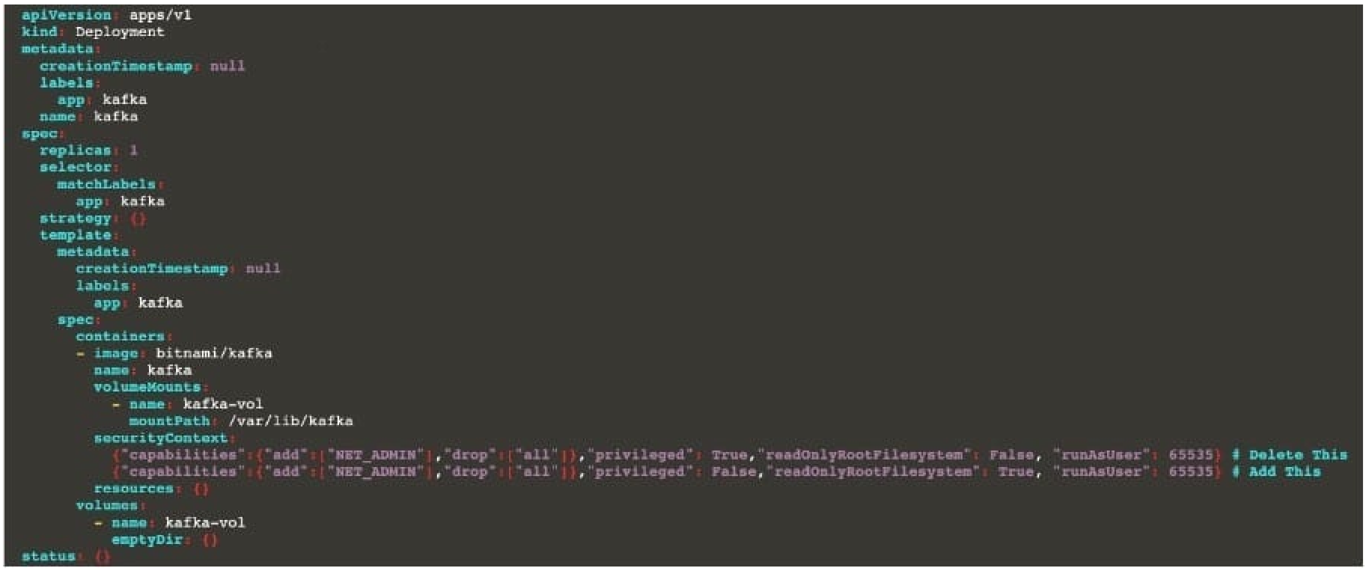
template:

metadata:

creationTimestamp: null

```
labels:

app: kafka

spec:

containers:

-image: bitnami/kafka

name: kafka

volumeMounts:

-

name: kafka-vol

mountPath: /var/lib/kafka

securityContext:

{"capabilities":{"add":["NET_ADMIN"],"drop":["all"]},"privileged":

True,"readOnlyRootFilesystem": False, "runAsUser": 65535} # Delete This
{"capabilities":{"add":["NET_ADMIN"],"drop":["all"]},"privileged":

False,"readOnlyRootFilesystem": True, "runAsUser": 65535} # Add This resources: {}

volumes:

-

name: kafka-vol

emptyDir: {}

status: {}
```

Pictorial View:[desk@cli] $ vim /home/cert_masters/mydeployment.yaml



```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: kafka
  name: kafka
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kafka
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: kafka
    spec:
      containers:
      - image: bitnami/kafka
        name: kafka
        volumeMounts:
          - name: kafka-vol
            mountPath: /var/lib/kafka
        securityContext:
          {"capabilities":{"add":["NET_ADMIN"],"drop":["all"]},"privileged": True,"readOnlyRootFilesystem": False, "runAsUser": 65535} # Delete This
          {"capabilities":{"add":["NET_ADMIN"],"drop":["all"]},"privileged": False,"readOnlyRootFilesystem": True, "runAsUser": 65535} # Add This
        resources: {}
      volumes:
        - name: kafka-vol
          emptyDir: {}
status: {}
```