



**Free Questions for *CKS* by *certscare***

**Shared by *Mitchell* on *12-12-2023***

**For More Free Questions and Preparation Resources**

***Check the Links on Last Page***

# Question 1

---

**Question Type:** MultipleChoice

---

Context

The kubeadm-created cluster's Kubernetes API server was, for testing purposes, temporarily configured to allow unauthenticated and unauthorized access granting the anonymous user cluster-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.

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.



## Options:

---

A- Explanation:



```
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
Z0lCQURBTKJna3Foa2lHOXcwQkFRc0ZBREFTVJND0VRWURWUWFERXdwcmRXSmwKY201bGRHVnpNQjRYFRJeU1ESXhO
akF3TlRveE9WblhEVEl5TURJeE5EQXd0VfV4T1Zvd0ZURVRNqkVHQTfVRQpBeElLYTNWVpYSnVaWFJsY3pDQ0FTSXdE
UVlKS29aSWh2Y05BUUVCQlFBRGdnRVBIBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTlVnZnZTktkSfdZU3JUaUx0QStr
N0lqTXpRZllzM2ttNGl1alpoM0tZc3Y1bUdpN0UyQ2tYc0MKUnh1L1NiZnBDMzlla2k5V3hOSHc5eTM00EtXUVE3VXBL
UmZRdXVxdlAlWXdDZkord1JmWGNdTXQxLzRNQVhWLPkdkjZ5YWRKSitPeFFSVjZlaHFBZHR0M3FtOfdVcW84UE5JT1E0
OEc3WWhnRUg5RHU3SFdkMS8raXVksjNOMX16CnNISEdtYklsWENSbEcydFV0M2RScDczSnRIS1JjS2tnMGxYM3FWS1Uy
QmJRblBmK01wb0V1TXFGcmZvcWVaVWcKY1BKK3ROVMzIM1JLTkhVUnYydVJIa3Zzc2Jrc1hUMW8rMXFNNHrYnFNMHlq
KzNXTUtISyt5V3dzUT1BYUVPMApUdXR4UUDlTFp3OUE3TjZzeTFVQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURB
Z0trTUE4R0ExVWRfD0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkZRUZEcU1wLzdYbzZaNAkNJVjVEK2w3bFZPcGpBOW1N
QlVHQTFVZEVURUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdeUUV1KS29aSWh2Y05BUUVMQlFBRGdnRUJBS1NWNm9wNGxYkNv
eGZLRUZ4bWoxaVlhUFlm1hhOTN0WEZ1TTY3RnA2NkdqUEc5SXBNnNHUnRnWV1yd0Mya1BDeFVOb2IySWtUQ1FNbDV3
cWRHCkdPS2JwVVP6Smc3Y0dyS2E3R1pZWVNYvUUVGRWwhyd2xZWxNGME56aFB0ZVcwcHJjcWtSdXN1bm55SG5YNGVOMUoK
N1NzbGZYTjJIdVfJd1VIRG15L0JsL1ZWRmZnZnRxoGF0Z0pYSFZGTmlVcDRpNX1JTXFRNTB4ZjVqcnFlWFRmVwpVdmJq
ZjEyOThXVTk3QkxHcDdrZE9QYWVKU051UStlVkJmRdnVZ2tVQVnjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3UlhlBgcvangvcitqd0dTc0xwWUxXDZTlxa1FraTdCSVRJT1N3ejd3c2hzbERuNzBFY0IKa0VBPQotLS0tLUVV
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
ocEdQcDB4Zk9JbkYxaGJwcTh5Y1BUMGxlTm5VNjBiSUpXRkVXkckxJbEtXC1NValh1VkyZnkl0ZHC1ZU1OT2JxK1haaHd
hY2JURVZCM1VDVURsbDgzdG5teFQyVXJmY0pUQmhLTctZTFavcWYKdjXR3BwQlZkNnhVZGFibGnuUklIMnpleUVJTEt
Tck5XbUQ0TzZsMU13blZ0OVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExKL3hiSDdUTzkwYlRFdlIwazl3cFVYd1lkdk1jSXN
MRkYwL3F2bDA3U3lxbGplOEI1SnNpQ1hCU1ZxbS9wCmNUUSs3SnZlbmdaZzlkOWdZaVJvdFFtCHBONkx4UnhkSzNKMGR
BK240SWxwFZEthRWh3TE00d0tMalDERG9scHgKYzB3WHkwVXBORGZ6UUXuRUFzVUJsbDRcQ3VkdW5QNvVDN2FuS3dJREF
RQUJBB0lCQURWRkZNSVRqYnNySTZTTwpQOGM0MTByN3RWZ25lcXJVS202dHRnZWtXOWdlSlpvMnZyb3RsbG9qOGFRamF
0MTZnaEUwOXDzd2xMSDhId0tLCK1Mb2NrZnFCUyt1OWo1ZmlFWGxYTG00cElCVDFRbGFJQlJRMdRyQ0JZbHdCN1VfVbV
1WjhuQ3lMR2JYTC9HM2wKcXBYTDVkdzJqcVh2MXdzCwSrdWNCrk0zZ0FYZk5YZkh1RExnV0VyNXRZRlF4VXo5UFFHOD1
pcDY1OTBkYnB1SApOMnU2NGk4UTgldk83OFVIT1c2eUFZU1loZVdha093RDFwZzNPdkhxV3FhbnV1Mn1rOWxaUUR0WW5
2MytBeU5DCnloN1RaRHluZ01ZdEptbDFTQ01TNEpSR2d4NXNwaCtKOC9XOGx0Ri9wMWZxbTA0bXZSRndxU3M2Y1JCQ2Z
PVVcKbFVlMGxLRUNnWUVBNWJzT01VZzFBVndjTmJsc0pSVDNURkI2OV1xbDRYcnZRR0FZY3BhdktENnd5VmtEOTV1O0p
```





```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeads.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:
```



```
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-1lktn	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-1lktn	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

**Answer:**

---

A

## Question 2

---

**Question Type: MultipleChoice**

---

Context

A PodSecurityPolicy shall prevent the creation of privileged Pods in a specific namespace.

Task

Create a new PodSecurityPolicy named prevent-`psp-policy`, which prevents the creation of privileged Pods.

Create a new ClusterRole named `restrict-access-role`, which uses the newly created PodSecurityPolicy `prevent-psp-policy`.

Create a new ServiceAccount named `psp-restrict-sa` in the existing namespace `staging`.

Finally, create a new ClusterRoleBinding named `restrict-access-bind`, which binds the newly created ClusterRole `restrict-access-role` to the newly created ServiceAccount `psp-restrict-sa`.



You can find skeleton  
manifest files at:



- /home/candidate/KSMV00  
102/pod-security-policy.ya  
ml
- /home/candidate/KSMV00  
102/cluster-role.yaml
- /home/candidate/KSMV00  
102/service-account.yaml
- /home/candidate/KSMV00  
102/cluster-role-binding.ya  
ml

## Options:

---

A- Explanation:

```
candidate@cli:~$ kubectl config use-context KSMV00102
Switched to context "KSMV00102".
candidate@cli:~$ cat /home/candidate/KSMV00102/pod-security-policy.yaml
---
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: ""
spec:
  seLinux:
    rule: ""
  runAsUser:
    rule: ""
  supplementalGroups: {}
  fsGroup: {}
candidate@cli:~$ vim /home/candidate/KSMV00102/pod-security-policy.yaml
```

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: "prevent-psp-policy"
spec:
  privileged: false
  seLinux:
    rule: RunAsAny
  runAsUser:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  fsGroup:
    rule: RunAsAny
```

```
candidate@cli:~$ vim /home/candidate/KSMV00102/pod-security-policy.yaml
candidate@cli:~$ cat /home/candidate/KSMV00102/pod-security-policy.yaml
---
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: "prevent-psp-policy"
spec:
  privileged: false
  seLinux:
    rule: RunAsAny
  runAsUser:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  fsGroup:
    rule: RunAsAny
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/pod-security-policy.yaml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/prevent-psp-policy created
candidate@cli:~$ cat /home/candidate/KSMV00102/cluster-role.yaml
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ""
rules:
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role.yaml
```

```
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: "restrict-access-role"
rules:
```

```
candidate@cli:~$ kubectl create clusterrole restrict-access-role --verb=use --resource=psp -
-dry-run=client -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  creationTimestamp: null
  name: restrict-access-role
rules:
- apiGroups:
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role.yaml
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: "restrict-access-role"
rules:
- apiGroups:
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role.yaml
candidate@cli:~$ kubectl create clusterrole restrict-access-role --verb=use --resource=psp -
-dry-run=client --resource-name=prevent-psp-policy -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  creationTimestamp: null
  name: restrict-access-role
rules:
- apiGroups:
  - policy
  resourceNames:
  - prevent-psp-policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role.yaml
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: "restrict-access-role"
rules:
- apiGroups:
  - policy
  resourceNames:
  - prevent-psp-policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/cluster-role.yaml
clusterrole.rbac.authorization.k8s.io/restrict-access-role created
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ cat /home/candidate/KSMV00102/service-account.yaml
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: "psp-restrict-sa"
  namespace: "staging"
```





```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ""
  namespace: ""
candidate@cli:~$ vim /home/candidate/KSMV00102/service-account.yaml
candidate@cli:~$ cat /home/candidate/KSMV00102/service-account.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: "psp-restrict-sa"
  namespace: "staging"
candidate@cli:~$ kubectl get sa -n staging
NAME          SECRETS  AGE
default       1         6h6m
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/service-account.yaml
serviceaccount/psp-restrict-sa created
candidate@cli:~$ kubectl get sa -n staging
NAME          SECRETS  AGE
default       1         6h6m
psp-restrict-sa  1         2s
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl create clusterrolebinding restrict-access-bind --clusterrole=restrict-access-role --serviceaccount=staging:psp-restrict-sa --dry-run -o yaml
W0520 14:41:23.502004 47627 helpers.go:598] --dry-run is deprecated and can be replaced with --dry-run=client.
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  creationTimestamp: null
  name: restrict-access-bind
roleRef:
  apiGroup: rbac.authorization.k8s.io
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restrict-access-bind
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restrict-access-role
subjects:
- kind: ServiceAccount
  name: psp-restrict-sa
  namespace: staging
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restrict-access-bind
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restrict-access-role
subjects:
- kind: ServiceAccount
  name: psp-restrict-sa
  namespace: staging

candidate@cli:~$
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/cluster-role-binding.yaml
clusterrolebinding.rbac.authorization.k8s.io/restrict-access-bind created
candidate@cli:~$ █
```

**Answer:**

---

A

## Question 3

---

**Question Type: MultipleChoice**

---

Context

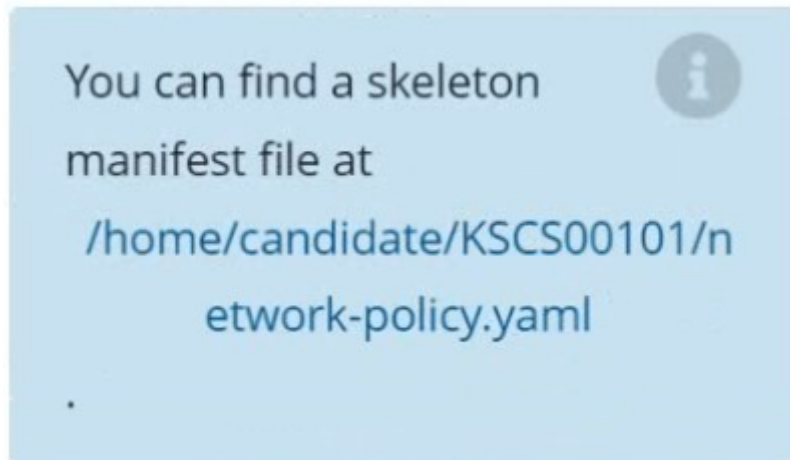
A default-deny NetworkPolicy avoids to accidentally expose a Pod in a namespace that doesn't have any other NetworkPolicy defined.

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.



**Options:**

---

A- Explanation:

```
candidate@cli:~$ kubectl config use-context KSCS00101
Switched to context "KSCS00101".
candidate@cli:~$ cat /home/candidate/KSCS00101/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ""
  namespace: ""
spec:
  podSelector: {}
  policyTypes: []
candidate@cli:~$ vim /home/candidate/KSCS00101/network-policy.yaml
candidate@cli:~$ █
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: "defaultdeny"
  namespace: "testing"
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress:
  - to:
    - podSelector: {}
      namespaceSelector:
        matchLabels:
          access: testingproject
```

```
candidate@cli:~$ vim /home/candidate/KSCS00101/network-policy.yaml
candidate@cli:~$ vim /home/candidate/KSCS00101/network-policy.yaml
candidate@cli:~$ kubectl label ns testing access=testingproject
namespace/testing labeled
candidate@cli:~$ cat /home/candidate/KSCS00101/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: "defaultdeny"
  namespace: "testing"
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress:
  - to:
    - podSelector: {}
      namespaceSelector:
        matchLabels:
          access: testingproject
candidate@cli:~$ kubectl create -f /home/candidate/KSCS00101/network-policy.yaml
networkpolicy.networking.k8s.io/defaultdeny created
candidate@cli:~$ kubectl -n testing describe networkpolicy
Name:          defaultdeny
Namespace:     testing
Created on:    2022-05-20 14:28:27 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  PodSelector:    <none> (Allowing the specific traffic to all pods in this namespace)
  Not affecting ingress traffic
  Allowing egress traffic:
```



**Answer:**

---

A

## Question 4

---

**Question Type: MultipleChoice**

---

Context

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

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:



Ensure that the  
`--authorization`

1.2.7 `-mode` FAIL  
argument is not  
set to

`AlwaysAllow`

Ensure that the  
`--authorization`

1.2.8 `-mode` FAIL  
argument


includes `Node`

Ensure that the  
`--authorization`

1.2.9 `-mode` FAIL  
argument

includes `RBAC`

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 Ensure that the `--authorization-mode` argument is not set to `AlwaysAllow` FAIL

Use `Webhook`



authentication/authorization  
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`

FAIL



### Options:

---

A- Explanation:

```
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.>
```

```
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:~$
```

**Answer:**

---

A

## Question 5

---

**Question Type:** MultipleChoice

---

Context

Your organization's security policy includes:

ServiceAccounts must not automount API credentials

ServiceAccount names must end in "-sa"

The Pod specified in the manifest file `/home/candidate/KSCH00301 /pod-manifest.yaml` fails to schedule because of an incorrectly specified ServiceAccount.

Complete the following tasks:

Task

1. Create a new ServiceAccount named `frontend-sa` in the existing namespace `qa`
  - a. Ensure the ServiceAccount does not automount API credentials.
2. Using the manifest file at `/home/candidate/KSCH00301 /pod-manifest.yaml`, create the Pod.
3. Finally, clean up any unused ServiceAccounts in namespace `qa`.

### Options:

---

**A-** Explanation:

```
Switched to context "KSCH00301".
candidate@cli:~$ kubectl get sa -n qa
NAME          SECRETS  AGE
default       1        5h46m
podrunner     1        5h46m
candidate@cli:~$ kubectl get deployment -n qa
No resources found in qa namespace.
candidate@cli:~$ kubectl get pod -n qa
No resources found in qa namespace.
candidate@cli:~$ kubectl create sa frontend-sa -n qa
serviceaccount/frontend-sa created
candidate@cli:~$ kubectl get sa -n qa
NAME          SECRETS  AGE
default       1        5h47m
frontend-sa   1        4s
podrunner     1        5h47m
candidate@cli:~$ cat /home/candidate/KSCH00301/pod-manifest.yaml
apiVersion: v1
kind: Pod
metadata:
  name: "frontend"
  namespace: "qa"
spec:
  serviceAccountName: "frontend-sa"
  containers:
  - name: "frontend"
    image: nginx
candidate@cli:~$ vim /home/candidate/KSCH00301/pod-manifest.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: "frontend"
  namespace: "qa"
spec:
  serviceAccountName: "frontend-sa"
  automountServiceAccountToken: false
  containers:
    - name: "frontend"
      image: nginx
```

```
candidate@cli:~$ vim /home/candidate/KSCH00301/pod-manifest.yaml
candidate@cli:~$ cat /home/candidate/KSCH00301/pod-manifest.yaml
apiVersion: v1
kind: Pod
metadata:
  name: "frontend"
  namespace: "qa"
spec:
  serviceAccountName: "frontend-sa"
  automountServiceAccountToken: false
  containers:
  - name: "frontend"
    image: nginx
candidate@cli:~$ kubectl create -f /home/candidate/KSCH00301/pod-manifest.yaml
pod/frontend created
candidate@cli:~$ kubectl get pods -n qa
NAME          READY   STATUS    RESTARTS   AGE
frontend     1/1     Running   0           6s
candidate@cli:~$ kubectl get sa -n qa
NAME          SECRETS   AGE
default      1         5h49m
frontend-sa  1         105s
podrunner    1         5h49m
candidate@cli:~$ kubectl delete sa/podrunner -n qa
serviceaccount "podrunner" deleted
candidate@cli:~$ □
```



**Answer:**

---

A

## Question 6

---

**Question Type: MultipleChoice**

---

Context

This cluster uses containerd as CRI runtime.

Containerd's default runtime handler is runc. Containerd has been prepared to support an additional runtime handler, runsc (gVisor).

Task

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

Update all Pods in the namespace server to run on gVisor.



You can find a skeleton  
manifest file at



`/home/candidate/KSMV00301/r  
untime-class.yaml`

## Options:

---

A- Explanation:

```
candidate@cli:~$ kubectl config use-context KSMV00301
Switched to context "KSMV00301".
candidate@cli:~$ cat /home/candidate/KSMV00301/runtime-class.yaml
---
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: ""
handler: ""
candidate@cli:~$ vim /home/candidate/KSMV00301/runtime-class.yaml
```



```
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: "sandboxed"
handler: "runsc"
```

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

:wq! █



```
candidate@cli:~$ kubectl config use-context KSMV00301
Switched to context "KSMV00301".
candidate@cli:~$ cat /home/candidate/KSMV00301/runtime-class.yaml
---
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: ""
handler: ""
candidate@cli:~$ vim /home/candidate/KSMV00301/runtime-class.yaml
candidate@cli:~$ cat /home/candidate/KSMV00301/runtime-class.yaml
---
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: "sandboxed"
handler: "runsc"
candidate@cli:~$ kubectl get deployments.apps -n server
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
workload1     1/1     1             1           5h43m
workload2     1/1     1             1           5h43m
workload3     1/1     1             1           5h43m
candidate@cli:~$ kubectl get pods -n server
NAME          READY   STATUS    RESTARTS   AGE
workload1-6869857dd7-s45rc  1/1     Running   0           5h43m
workload2-d4bd497d5-h44df   1/1     Running   0           5h43m
workload3-8587774495-chm56  1/1     Running   0           5h43m
candidate@cli:~$ kubectl -n server edit deployments.apps workload1
```



```
template:
  metadata:
    creationTimestamp: null
    labels:
      app: nginx
    name: workload1
  spec:
    runtimeClassName: sandboxed
    containers:
    - image: nginx:1.14.2
      imagePullPolicy: IfNotPresent
      name: workload1
      ports:
      - containerPort: 80
        protocol: TCP
      resources: {}
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    dnsPolicy: ClusterFirst
    restartPolicy: Always
    schedulerName: default-scheduler
    securityContext: {}
    terminationGracePeriodSeconds: 30
status:
"/tmp/kubect1-edit-3385772700.yaml"
```



```
NAME          READY   STATUS    RESTARTS   AGE
workload1-6869857dd7-s45rc  1/1     Running   0           5h44m
workload2-d4bd497d5-h44df   1/1     Running   0           5h44m
workload3-8587774495-chm56  1/1     Running   0           5h44m
candidate@cli:~$ kubectl -n server edit deployments.apps workload1
Edit cancelled, no changes made.
candidate@cli:~$ kubectl get pods -n server
NAME          READY   STATUS    RESTARTS   AGE
workload1-6869857dd7-s45rc  1/1     Running   0           5h45m
workload2-d4bd497d5-h44df   1/1     Running   0           5h44m
workload3-8587774495-chm56  1/1     Running   0           5h44m
candidate@cli:~$ kubectl -n server edit deployments.apps workload2
Edit cancelled, no changes made.
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00301/runtime-class.yaml
runtimeclass.node.k8s.io/sandboxed created
candidate@cli:~$ kubectl get pods -n server
NAME          READY   STATUS    RESTARTS   AGE
workload1-6869857dd7-s45rc  1/1     Running   0           5h45m
workload2-d4bd497d5-h44df   1/1     Running   0           5h45m
workload3-8587774495-chm56  1/1     Running   0           5h45m
candidate@cli:~$ kubectl -n server edit deployments.apps workload2
```

```
strategy:
  rollingUpdate:
    maxSurge: 25%
    maxUnavailable: 25%
  type: RollingUpdate
template:
  metadata:
    creationTimestamp: null
    labels:
      app: nginx
      name: workload2
  spec:
    runtimeClassName: sandboxed
```

```
NAME                READY   STATUS    RESTARTS   AGE
workload1-6869857dd7-s45rc    1/1     Running   0           5h45m
workload2-d4bd497d5-h44df    1/1     Running   0           5h45m
workload3-8587774495-chm56   1/1     Running   0           5h45m
candidate@cli:~$ kubectl -n server edit deployments.apps workload2
deployment.apps/workload2 edited
candidate@cli:~$ kubectl get pods -n server
NAME                READY   STATUS    RESTARTS   AGE
workload1-8d8649ff6-wvjtg    1/1     Running   0           15s
workload2-765bdb98c8-wd8cm   1/1     Running   0           4s
workload3-8587774495-chm56   1/1     Running   0           5h45m
candidate@cli:~$ kubectl -n server edit deployments.apps workload3
```

```
  app: nginx
  name: workload3
spec:
  runtimeClassName: sandboxed
  containers:
  - image: nginx:1.14.2
    imagePullPolicy: IfNotPresent
    name: workload3
    ports:
```

```
candidate@cli:~$ kubectl -n server edit deployments.apps workload3
deployment.apps/workload3 edited
candidate@cli:~$ kubectl get pods -n server
NAME                                READY   STATUS    RESTARTS   AGE
workload1-8d8649ff6-wvjtg          1/1    Running   0          58s
workload2-765bdb98c8-wd8cm        1/1    Running   0          47s
workload3-76c994bb4d-s6k85        1/1    Running   0          4s
candidate@cli:~$
```

**Answer:**

---

A

## Question 7

---

**Question Type:** MultipleChoice

---

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: 1.2.7authorization-modeargument is not set toAlwaysAllow FAIL 1.2.8authorization-modeargument includesNode FAIL 1.2.7authorization-modeargument includesRBAC 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 authn/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

## Options:

---

**A-** Explanation:

```
worker1 $ vim /var/lib/kubelet/config.yaml
```

anonymous:

enabled: true #Delete this

enabled: false #Replace by this

authorization:

mode: AlwaysAllow #Delete this

mode: Webhook #Replace by this

```
worker1 $ systemctl restart kubelet. # To reload kubelet config
```

ssh to master1

```
master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
- -- authorization-mode=Node,RBAC
```

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

```
- --client-cert-auth=true
```

Explanation

ssh to worker1

```
worker1 $ 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 config
```

ssh to 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

```
apiVersion: v1
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
    imagePullPolicy: IfNotPresent
```

**Answer:**

---

A

## Question 8

---

**Question Type: MultipleChoice**

---

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-sato 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 statefulsets. 3. Create a new RoleBinding named test-role-2-binding binding the newly created Role to the Pod's ServiceAccount. Note: Don't delete the existing RoleBinding.

**Options:**

---

**A-** Explanation:

```
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:~$ █

```

**Answer:**

---

A

## Question 9

---

**Question Type: MultipleChoice**

---

Context: Cluster:gvisor Master node:master1 Worker node:worker1

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

```
[desk@cli] $kubectl config use-context gvisor
```

Context:This cluster has been prepared to support runtime handler, runsc as well as traditional one.

Task: Create a RuntimeClass namednot-trustedusing the prepared runtime handler namesrunsc. Update all Pods in the namespace server to run onnewruntime.

**Options:**

---

**A-** Explanation:



### 1. Create runtime class by the name of not-trusted using runsc handler

```
1 | apiVersion: node.k8s.io/v1
2 | kind: RuntimeClass
3 | metadata:
4 |   name: not-trusted
5 |   handler: runsc
```

### 2. Find all the pods/deployment and edit runtimeClassName parameter to not-trusted under spec

```
[desk@cli] $ k edit deploy nginx
```

```
1 | spec:
2 |   runtimeClassName: not-trusted. # Add this
```

Explanation

```
[desk@cli] $vim runtime.yaml
```

```
apiVersion: node.k8s.io/v1
```

```
kind: RuntimeClass
```

```
metadata:
```

```
name: not-trusted
```

```
handler: runsc
```

```
[desk@cli] $k apply -f runtime.yaml
```

```
[desk@cli] $k get pods
```

```
NAME READY STATUS RESTARTS AGE
```

```
nginx-6798fc88e8-chp6r 1/1 Running 0 11m
```

```
nginx-6798fc88e8-fs53n 1/1 Running 0 11m
```

```
nginx-6798fc88e8-ndved 1/1 Running 0 11m
```

```
[desk@cli] $k get deploy
```

```
NAME READY UP-TO-DATE AVAILABLE AGE
```

```
nginx 3/3 11 3 5m
```

```
[desk@cli] $k edit deploy nginx
```



```
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: {}
```

## Answer:

---

A

## Question 10

---

### Question Type: MultipleChoice

---

You can switch the cluster/configuration context using the following command: [desk@cli] \$kubectl config use-context dev A default-deny NetworkPolicy avoid to accidentally expose a Pod in a namespace that doesn't have any other NetworkPolicy defined.

Task: Create a new default-deny NetworkPolicy nameddeny-networkin the namespacestestfor all traffic of type Ingress + Egress

The new NetworkPolicy must deny all Ingress + Egress traffic in the namespacestest.

Apply the newly createddefault-denyNetworkPolicy to all Pods running in namespacestest.

You can find a skeleton manifests file at /home/cert\_masters/network-policy.yaml

## Options:

---

**A-** Explanation:

```
master1 $k get pods -n test --show-labels
```

```
NAME READY STATUS RESTARTS AGE LABELS
```

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

```
testing 1/1 Running 0 17d run=testing
```

```
$vim netpol.yaml
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: NetworkPolicy
```

```
metadata:
```

```
name: deny-network
```

```
namespace: test
```

```
spec:
```

```
podSelector: {}
```

```
policyTypes:
```

```
- Ingress
```

```
- Egress
```

```
master1 $k apply -f netpol.yaml
```

```
Explanation
```

```
controlplane $ k get pods -n test --show-labels
```

```
NAME READY STATUS RESTARTS AGE LABELS
```

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

```
testing 1/1 Running 0 17d run=testing
```

```
master1 $ vim netpol1.yaml
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: NetworkPolicy
```

```
metadata:
```

```
name: deny-network
```

```
namespace: test
```

```
spec:
podSelector: {}
policyTypes:
- Ingress
- Egress
master1 $ k apply -f netpol1.yaml
```

Reference:

<https://kubernetes.io/docs/concepts/services-networking/network-policies/>

Explanation

```
controlplane $ k get pods -n test --show-labels
NAME READY STATUS RESTARTS AGE LABELS
test-pod 1/1 Running 0 34s role=test,run=test-pod
testing 1/1 Running 0 17d run=testing
master1 $ vim netpol1.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: deny-network
namespace: test
spec:
podSelector: {}
policyTypes:
- Ingress
- Egress
master1 $ k apply -f netpol1.yaml
```

Reference:

<https://kubernetes.io/docs/concepts/services-networking/network-policies/>

**Answer:**

---

A



**To Get Premium Files for CKS Visit**

<https://www.p2pexams.com/products/cks>

**For More Free Questions Visit**

<https://www.p2pexams.com/linux-foundation/pdf/cks>

