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.passapply.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

CORRECT TEXT

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.

A. See explanation below.

B. PlaceHolder

Correct Answer: A

```
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=restri
ct-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 wi
th --dry-run=client.
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  creationTimestamp: null
  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:~$ vim /home/candidate/KSMV00102/cluster-role
cluster-role-binding.yaml  cluster-role.yaml
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role
cluster-role-binding.yaml  cluster-role.yaml
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role-binding.yaml
```

```
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
```

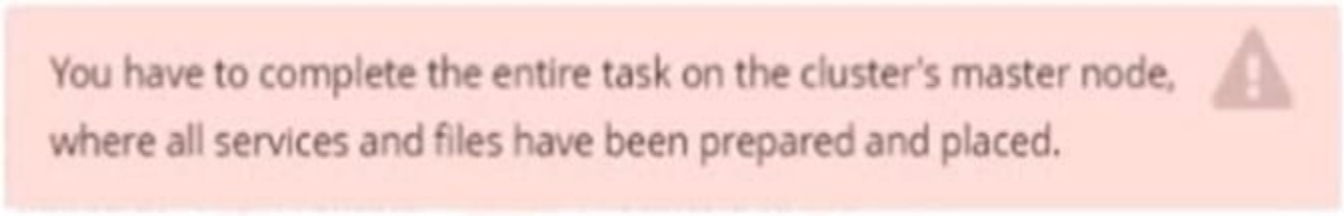
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restrict-access-bind
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restrict-access-role
subjects:
- kind: ServiceAccount
  name: psp-restrict-sa
  namespace: staging

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:~$
```

QUESTION 2

CORRECT TEXT

A container image scanner is set up on the cluster, but it\\'s not yet fully integrated into the cluster s configuration. When complete, the container image scanner shall scan for and reject the use of vulnerable images.



Given an incomplete configuration in directory /etc/kubernetes/epconfig and a functional container image scanner with HTTPS endpoint https://wakanda.local:8081 /image_policy:

1.

Enable the necessary plugins to create an image policy

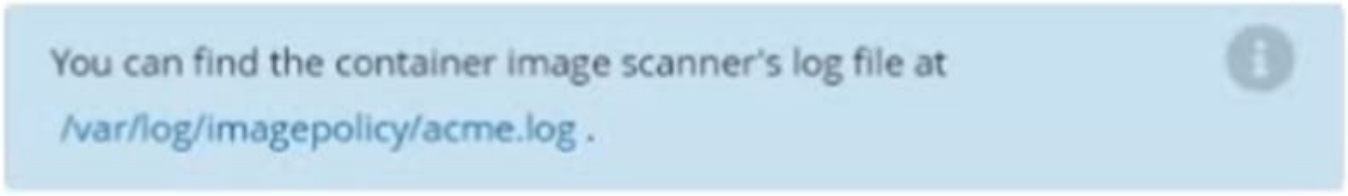
2.

Validate the control configuration and change it to an implicit deny

3.

Edit the configuration to point to the provided HTTPS endpoint correctly

Finally, test if the configuration is working by trying to deploy the vulnerable resource /root/KSSC00202/vulnerable-resource.yml.



A. See the explanation below

B. PlaceHolder

Correct Answer: A

QUESTION 3

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 named deny-network in the namespace test for all traffic of type Ingress + Egress

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

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

You can find a skeleton manifests file at /home/cert_masters/network-policy.yaml

A. See the explanation below

B. PlaceHolder

Correct Answer: A

master1 $ k get pods -n test --show-labels uk.co.certification.simulator.questionpool.PList@132b47c0 $ vim netpol.yaml uk.co.certification.simulator.questionpool.PList@132b4af0 master1 $ k apply -f netpol.yaml

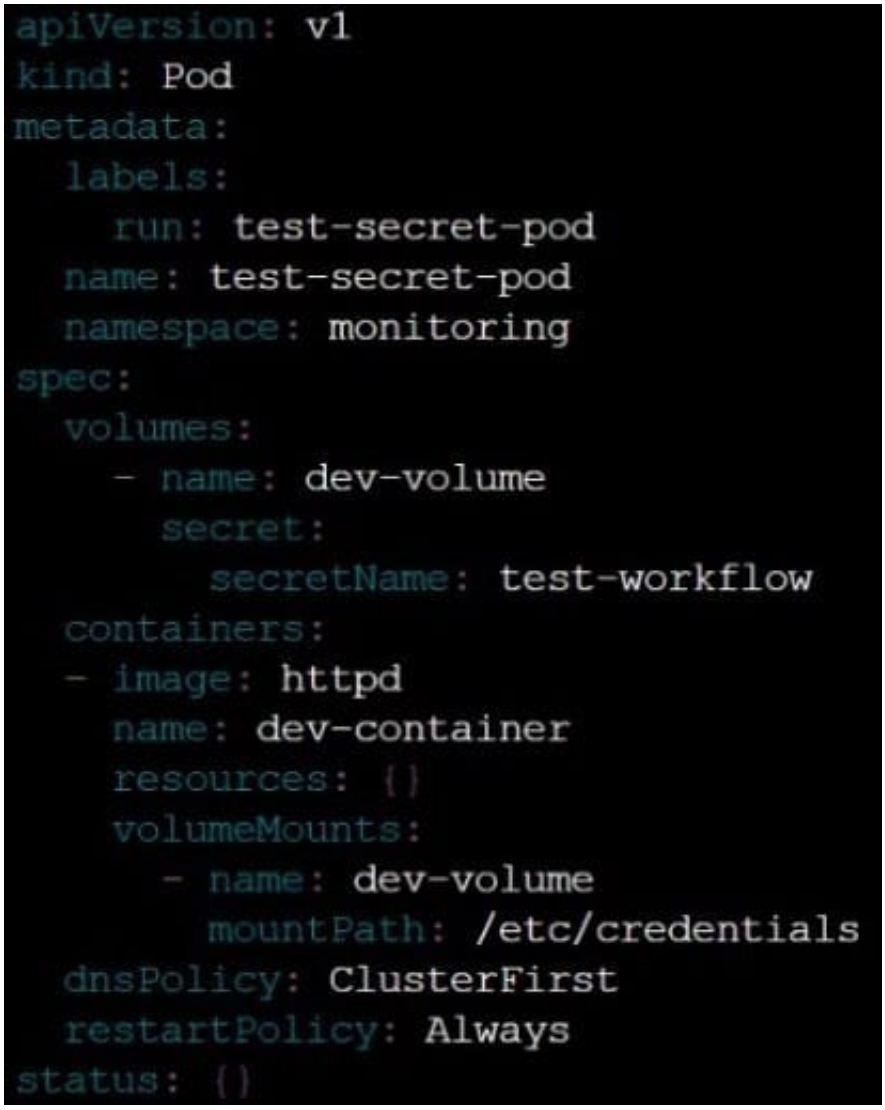
controlplane $ k get pods -n test --show-labels NAME READY STATUS RESTARTS AGE LABELS test-pod 1/1 Running 0 34s role=test,run=test-pod testing 1/1 Running 0 17d run=testing master1 $ vim netpol1.yaml apiVersion: networking.k8s.io/v1 kind: NetworkPolicy metadata: name: deny-network namespace: test spec: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

-Ingress

-Egress

QUESTION 4

```
candidate@cli:~$ kubectl config use-context KSMV00201
Switched to context "KSMV00201".
candidate@cli:~$ kubectl get secret -n monitoring
NAME                  TYPE                                  DATA   AGE
db1-test              Opaque                                2      6h23m
default-token-cqqf6   kubernetes.io/service-account-token   3      6h23m
candidate@cli:~$ kubectl get secret/db1-test -n monitoring
NAME       TYPE     DATA   AGE
db1-test   Opaque   2      6h23m
candidate@cli:~$ kubectl get secret/db1-test -n monitoring -o yaml
apiVersion: v1
data:
  password: QVU3dHhlbXFOTHZt
  username: cHJvZHVjdGlvbi0x
kind: Secret
metadata:
  creationTimestamp: "2022-05-20T08:37:33Z"
  name: db1-test
  namespace: monitoring
  resourceVersion: "2588"
  uid: 659bd4ac-e0ba-4d9f-b411-816f2aedf7e6
type: Opaque
candidate@cli:~$ echo "cHJvZHVjdGlvbi0x" | base64 -d
production-1candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ echo "cHJvZHVjdGlvbi0x" | base64 -d  > /home/candidate/username.txt
candidate@cli:~$ cat /home/candidate/username.txt
production-1candidate@cli:~$
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ echo "QVU3dHhlbXFOTHZt" | base64 -d
AU7txumqNLvmcandidate@cli:~$ echo "QVU3dHhlbXFOTHZt" | base64 -d > /home/candidate/password.
txt
candidate@cli:~$ cat /home/candidate/password.txt
AU7txumqNLvmcandidate@cli:~$
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl create secret generic test-workflow --from-literal=username=dev-dat
abase --from-literal=password=aV7HR7nU3JLx -n monitoring
secret/test-workflow created
candidate@cli:~$
candidate@cli:~$
candidate@cli:~$ kubectl  -n monitoring run test-secret-pod --image=httpd --dry-run=client -
o yaml > test-secret-pod.yaml
candidate@cli:~$ vim test-secret-pod.yaml
```



```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: test-secret-pod
  name: test-secret-pod
  namespace: monitoring
spec:
  volumes:
    - name: dev-volume
      secret:
        secretName: test-workflow
  containers:
  - image: httpd
    name: dev-container
    resources: {}
    volumeMounts:
      - name: dev-volume
        mountPath: /etc/credentials
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

```
candidate@cli:~$ kubectl  -n monitoring run test-secret-pod --image=httpd --dry-run=client -
o yaml > test-secret-pod.yaml
candidate@cli:~$ vim test-secret-pod.yaml
candidate@cli:~$ cat test-secret-pod.yaml
```

```
  labels:
    run: test-secret-pod
  name: test-secret-pod
  namespace: monitoring
spec:
  volumes:
    - name: dev-volume
      secret:
        secretName: test-workflow
  containers:
  - image: httpd
    name: dev-container
    resources: {}
    volumeMounts:
      - name: dev-volume
        mountPath: /etc/credentials
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
candidate@cli:~$ kubectl create -f test-secret-pod.yaml
pod/test-secret-pod created
candidate@cli:~$ kubectl get pods -n monitoring
NAME              READY   STATUS    RESTARTS   AGE
test-secret-pod   1/1     Running   0          9s
candidate@cli:~$
```

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 named not-trusted using the prepared runtime handler names runsc.

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

A. See the explanation below

B. PlaceHolder

Correct Answer: A

1.

apiVersion: node.k8s.io/v1

2.

kind: RuntimeClass

3.

metadata:

4.

name: not-trusted

5.

handler: runsc [desk@cli] $ k apply -f runtime.yaml[desk@cli] $ k get pods

1.

NAME READY STATUS RESTARTS AGE

2.

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

3.

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

4.

nginx-6798fc88e8-ndved 1/1 Running 0 11m [desk@cli] $ k get deploy

1.

NAME READY UP-TO-DATE AVAILABLE AGE

2.

nginx 3/3 11 3 5m [desk@cli] $ k edit deploy nginx

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

```
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: not-trusted
handler: runsc
```

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

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

```
spec:
  runtimeClassName: not-trusted. # Add this
```

[desk@cli] $vim runtime.yaml

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

QUESTION 5

```
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:~$
```

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

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

Context:

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

Task:

1.

Create a new PodSecurityPolcy named deny-policy, which prevents the creation of privileged Pods.

2.

Create a new ClusterRole name deny-access-role, which uses the newly created PodSecurityPolicy deny-policy.

3.

Create a new ServiceAccount named psd-denial-sa in the existing namespace development.

Finally, create a new ClusterRoleBindind named restrict-access-bind, which binds the newly created ClusterRole deny-access-role to the newly created ServiceAccount psp-denial-sa

A. See the explanation below

B. PlaceHolder

Correct Answer: A

Create psp to disallow privileged container uk.co.certification.simulator.questionpool.PList@11600d40 k create sa psp-denial-sa -n development uk.co.certification.simulator.questionpool.PList@11601040 namespace: development Explanationmaster1 $ vim psp.yaml apiVersion: policy/v1beta1 kind: PodSecurityPolicy metadata: name: deny-policy spec: privileged: false # Don\\'t allow privileged pods! seLinux: rule: RunAsAny supplementalGroups: rule: RunAsAny runAsUser: rule: RunAsAny fsGroup: rule: RunAsAny volumes:

-\\'*\\'

master1 $ vim cr1.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

name: deny-access-role

rules:

-apiGroups: [\\'policy\\']

resources: [\\'podsecuritypolicies\\']

verbs: [\\'use\\']

resourceNames:

-"deny-policy"

master1 $ k create sa psp-denial-sa -n developmentmaster1 $ vim cb1.yaml apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: restrict-access-bing

roleRef:

kind: ClusterRole

name: deny-access-role

apiGroup: rbac.authorization.k8s.io

subjects:

Authorize specific service accounts:

-kind: ServiceAccount

name: psp-denial-sa

namespace: development