Pod Security Policies

Minikube Setup

https://minikube.sigs.k8s.io/docs/tutorials/using_psp/

Install minikube on Mac with Apple Silicon

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-arm64
sudo install minikube-darwin-arm64 /usr/local/bin/minikube
```

Start Minikube with Kubernetes version 1.21.5 running in Docker for Desktop.

```
minikube start --extra-config=apiserver.enable-admission-plugins=PodSecurityPolicy --addons=pod-security-policy
--driver=docker --alsologtostderr --kubernetes-version=v1.21.5
```

```
...
I1014 10:25:53.474818   47355 out.go:177] Done! kubectl is now configured to use "minikube" cluster and
"default" namespace by default
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Apply PSPs

By default, minikube will apply the following pod security policies, cluster roles and bindings:

```
---
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: privileged
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: "*"
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - "*"
  volumes:
  - "*"
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
```

```
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
---
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restricted
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
spec:
  privileged: false
  allowPrivilegeEscalation: false
  requiredDropCapabilities:
    - ALL
  volumes:
    - 'configMap'
    - 'emptyDir'
    - 'projected'
    - 'secret'
    - 'downwardAPI'
    - 'persistentVolumeClaim'
  hostNetwork: false
  hostIPC: false
  hostPID: false
  runAsUser:
    rule: 'MustRunAsNonRoot'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'MustRunAs'
    ranges:
      # Forbid adding the root group.
      - min: 1
        max: 65535
  fsGroup:
    rule: 'MustRunAs'
    ranges:
      # Forbid adding the root group.
      - min: 1
        max: 65535
  readOnlyRootFilesystem: false
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: psp:privileged
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - privileged
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: psp:restricted
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - restricted
---
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: default:restricted
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: psp:restricted
subjects:
- kind: Group
  name: system:authenticated
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: default:privileged
  namespace: kube-system
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: psp:privileged
subjects:
- kind: Group
  name: system:masters
  apiGroup: rbac.authorization.k8s.io
- kind: Group
  name: system:nodes
  apiGroup: rbac.authorization.k8s.io
- kind: Group
  name: system:serviceaccounts:kube-system
  apiGroup: rbac.authorization.k8s.io
```

Test

```
vi dangerous-pod.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dangerous-deploy
  labels:
    app: dangerous-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dangerous-deploy
  template:
    metadata:
      labels:
        app: dangerous-deploy
    spec:
      containers:
        - name: alpine
          image: alpine
          stdin: true
          tty: true
          securityContext:
            privileged: true
      hostPID: true
      hostNetwork: true
```

```
kubectl apply -f dangerous-pod.yaml
```

You should notice that it doesn't deploy. Now test some other installations and see what needs to be done to bypass.

Fluent-bit Example

For fluent-bit, we are going to define a separate pod security policy and bind it to fluent-bit's service account.

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: ncyd-fluent-bit-psp
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: "*"
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
spec:
  privileged: false
  allowPrivilegeEscalation: false
  requiredDropCapabilities:
    - ALL
  volumes:
    - 'hostPath'
    - 'configMap'
    - 'emptyDir'
    - 'projected'
    - 'secret'
    - 'downwardAPI'
    - 'persistentVolumeClaim'

  hostNetwork: false
  hostIPC: false
  hostPID: false
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'MustRunAs'
    ranges:
      # Forbid adding the root group.
      - min: 1
        max: 65535
  fsGroup:
    rule: 'MustRunAs'
    ranges:
      # Forbid adding the root group.
      - min: 1
        max: 65535
  readOnlyRootFilesystem: false

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ncyd-fluent-bit
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - ncyd-fluent-bit-psp
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: ncyd-fluent-bit
subjects:
- kind: ServiceAccount
  name: fluent-bit
  namespace: default
roleRef:
  kind: ClusterRole
  name: ncyd-fluent-bit
  apiGroup: rbac.authorization.k8s.io
```

Apply:

```
kubectl apply -f <psp_file>
```

References

Reference	URL
Using Minikube with Pod Security Policies	https://minikube.sigs.k8s.io/docs/tutorials/using_psp/
How to run a Minikube on Apple Silicon M1	https://medium.com/@seohee.sophie.kwon/how-to-run-a-minikube-on-apple-silicon-m1-8373c248d669