Jak na Pod Security Policies v Kubernetes

Začínáme

- ověř, že běží minikube, případně nastartuj

```
> minikube status || minikube start
```

- naklonuj repozitář

```
> git clone https://gitlab.com/filip.havlicek/linuxdays_psp-workshop.git
```

- ověř, že cluster je up

```
> kubectl get all --all-namespaces
```

Pokračujeme

https://gitlab.com/filip.havlicek/linuxdays_psp-workshop

Co jsou Pod Security Policies

Opatření na úrovni clusteru umožňující kontrolu nastavení podu.

Objekt `PodSecurityPolicy` definuje jaké podmínky musí pod splňovat aby mohl být spuštěn. Zároveň umožňuje nastavení výchozích hodnot pro některé aspekty, které kontroluje.

Volitelný admission controller (mutating i validating). Policy objekty mohou být vytvořeny bez zapnutí admission pluginu.

Co umožňují kontrolovat

- privilegované kontejnery
- přístup k host namespacům (IPC, PID)
- host networking a host porty
- typy volumů
- konkrétní HostPaths a FlexVolumes
- container uid a gid
- eskalace oprávnění
- read-only root file system
- Linux capabilities
- SELinux context, AppArmor profil, seccomp profil

Proč je používat

- bezpečnost
- povolení nebo odepření přístupu ke konkrétním prostředkům
- defaulting

ale…

- neexistuje jedna dokonalá policy
- různé úrovně policy

Zapnutí Pod Security Policies

- vytvoření `PodSecurityPolicy` objektů
- autorizace (RBAC)
- zapnutí `PodSecurityPolicy` admission controlleru

PodSecurityPolicy

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: privileged
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
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'
```

Autorizace

- vytvoření `PodSecurityPolicy` samo o sobě nedělá nic
- aby bylo možné pod security policy použít, musí k tomu být identita, která pouští pod autorizována - RBAC
- pody jsou obvykle vytvářeny nepřímo - `Deployment`, `ReplicaSet`
- default policy - autorizovat všechny (controller manager nebo autentizovaní uživatelé) k nejrestriktivnější
- k privilegovanějším policy autorizovat konkrétní identity (service accounty)

ClusterRole

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: psp:privileged
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - privileged
```

ClusterRoleBinding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: psp:privileged
roleRef:
  kind: ClusterRole
  name: psp:privileged
  apiGroup: rbac.authorization.k8s.io
subjects:
- kind: ServiceAccount
  name: kube-proxy
  namespace: kube-system
```

Doporučené zdroje

- https://kubernetes.io/docs/concepts/policy/pod-security-policy/
- https://github.com/sysdiglabs/kube-psp-advisor

Otázky?

Děkuji za pozornost!