SECURING CONTAINERS IN KUBERNETES DEPLOYMENTS

learnKUBE

Does app need root priviliges ? YES

Does it need all capabilities ? YES

```
apiVersion: v1
kind: Pod
metadata:
  name: node-cap-chown
spec:
  containers:
  - name: node
    image: node:20-slim
    command: ["sleep", "3600"]
    securityContext:
      priviliged: true
```

⚠ Document why it is needed

NO

```
apiVersion: v1
kind: Pod
metadata:
  name: node-nonroot
spec:
  securityContext:
    runAsUser: 1000
    runAsNonRoot: true
  containers:
    - name: node
      image: node:20-slim
      command: ["sleep", "3600"]
```

Force the container to run as a regular user

NO

```
apiVersion: v1
kind: Pod
metadata:
  name: node-cap-chown
spec:
  securityContext:
    runAsUser: 0
    capabilities:
      drop: ["ALL"]
      add: ["CHOWN"]
  containers:
  - name: node
    image: node:20-slim
    command: ["sleep", "3600"]
```

Drop all capabilities and add only needed ones

Doess the app write to the filesystem ? NO

```
apiVersion: v1
kind: Pod
metadata:
  name: node-nonroot
spec:
  securityContext:
    readOnlyRootFilesystem: true
  containers:
    - name: node
      image: node:20-slim
      command: ["sleep", "3600"]
```

Mount the filesystem as read-only

YES

1. Mount writable volumes, keep root filesystem read-only
2. Set fsGroup for volume permissions

```
apiVersion: v1
kind: Pod
metadata:
  name: node-nonroot
spec:
  securityContext:
    fsGroup: 2000
    readOnlyRootFilesystem: true
  containers:
    - name: node
      image: node:20-slim
      command: ["sleep", "3600"]
      volumeMounts:
        - name: logs
          mountPath: /var/log
  volumes:
    - name: logs
      emptyDir: {}
```

```
apiVersion: v1
kind: Pod
metadata:
  name: node-nonroot
spec:
  securityContext:
    allowPrivilegeEscalation: false
  containers:
    - name: node
      image: node:20-slim
      command: ["sleep", "3600"]
```

Disable privilege escalation

OPTIONAL

Apply an AppArmor profile

```
apiVersion: v1
kind: Pod
metadata:
  name: node-nonroot
spec:
  securityContext:
    appArmorProfile:
      type: Localhost
      localhostProfile: example-deny-write
  containers:
    - name: node
      image: node:20-slim
      command: ["sleep", "3600"]
```

```
#include <tunables/global>

profile example-deny-write
flags=(attach_disconnected) {
  #include <abstractions/base>

  file,

  # Deny all file writes.
  deny /** w,
}
```

Apply a seccomp profile

```
apiVersion: v1
kind: Pod
metadata:
  name: node-nonroot
spec:
  securityContext:
    seccompProfile:
      type: RuntimeDefault
  containers:
    - name: node
      image: node:20-slim
      command: ["sleep", "3600"]
```

```
{
 "defaultAction": "SCMP_ACT_ERRNO",
 "architectures": ["SCMP_ARCH_X86_64"],
 "syscalls": [
   {
     "names": [
       "read", "write", "close",
       "exit", "exit_group",
       "brk", "mmap", "munmap"
     ],
     "action": "SCMP_ACT_ALLOW"
   }
 ]
}
```

END