

CKAD Ders Dökümanı #08 — Security Contexts

Dil: Türkçe • CKAD Kategorisi: Application Environment, Configuration and Security (25%) •
Standart: Konu anlatımı + komutlar + örnekler

1) Konu Anlatımı

SecurityContext, Pod veya container seviyesinde güvenlikle ilgili çalışma parametrelerini tanımlar (kullanıcı/grup, yetkiler, dosya sistemi izinleri, capabilities vb.).

CKAD'de sık görevler: runAsUser, fsGroup, readOnlyRootFilesystem ve capabilities ekleme/çıkarma.

2) En Sık Karşılaşılan YAML Örnekleri

Pod seviyesinde runAsUser/fsGroup

```
apiVersion: v1
kind: Pod
metadata:
  name: sc-pod
spec:
  securityContext:
    runAsUser: 1000
    fsGroup: 2000
  containers:
    - name: app
      image: busybox:1.36
      command: ["sh", "-c"]
      args: ["id; sleep 3600"]
```

Container seviyesinde readOnlyRootFilesystem

```
apiVersion: v1
kind: Pod
metadata:
  name: ro-root
spec:
  containers:
    - name: app
      image: busybox:1.36
      securityContext:
        readOnlyRootFilesystem: true
      command: ["sh", "-c"]
      args: ["touch /tmp/x || true; sleep 3600"]
```

Capabilities (add/drop)

```
securityContext:
  capabilities:
    add: ["NET_BIND_SERVICE"]
    drop: ["ALL"]
```

3) Sık Kullanılan Alanlar (Kısa Açıklamalar)

Alan	Ne işe yarar?
spec.securityContext	Pod seviyesinde güvenlik ayarları (tüm container'lara etki edebilir).
containers[].securityContext	Container seviyesinde ayarlar.
runAsUser / runAsGroup	Container süreçlerinin çalışacağı UID/GID.
fsGroup	Volume dosyalarına grup bazlı erişim için.
readOnlyRootFilesystem	Root FS'i salt okunur yapar.
allowPrivilegeEscalation	Privilege escalation kontrolü.
capabilities	Linux capabilities ekleme/çıkarma.

4) En Sık Kullanılan Komutlar ve Kullanımları

Aşağıdaki komutlar CKAD pratiklerinde en çok kullanılanlardır.

4.1 Pod'u inceleme

SecurityContext'in uygulandığını görürsünüz.

```
kubectl get pod sc-pod -o yaml | sed -n '/securityContext:/,/containers:/p'  
kubectl exec -it sc-pod -- id
```

4.2 Hızlı debug

İzin problemlerinde describe + logs.

```
kubectl describe pod ro-root  
kubectl logs ro-root
```

5) Troubleshooting Hızlı Rehber

Permission denied: runAsUser/fsGroup doğru mu? Uygulamanın yazması gereken dizin için emptyDir/PVC + doğru fsGroup gerekebilir. Bind port 80 hatası: non-root çalışıyorsanız 1024 altı portlar için NET_BIND_SERVICE capability gereklidir. readOnlyRootFilesystem: uygulama root filesystem'e yazıyorsa /tmp için emptyDir mount edin.

6) CKAD İpuçları

En sık: runAsUser + fsGroup + readOnlyRootFilesystem Sorun çözmede: describe event'ler + logsPort 80 gerekiyorsa: capability düşün (veya 8080'e geç).