CKAD Ders Dökümanı #06 — ConfigMap

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

1) Konu Anlatımı

ConfigMap, uygulama konfigürasyonunu (hassas olmayan) container image'ından ayırmak için kullanılır. Böylece aynı image farklı ortamlarda (dev/test/prod) farklı ayarlarla çalıştırılabilir.

ConfigMap verisini Pod'a iki ana şekilde verirsiniz: env / envFrom (ortam değişkeni) veya volume mount (dosya olarak).

2) En Sık Karşılaşılan YAML Örnekleri

ConfigMap (key/value)

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  LOG_LEVEL: "info"
  FEATURE_X: "true"
```

Pod'a envFrom ile verme

```
apiVersion: v1
kind: Pod
metadata:
  name: web
spec:
  containers:
  - name: web
    image: nginx:1.25
    envFrom:
    - configMapRef:
        name: app-config
```

Dosya olarak mount etme

```
apiVersion: v1
kind: Pod
metadata:
  name: app
spec:
  volumes:
  - name: cfg
    configMap:
      name: app-config
  containers:
  - name: app
    image: busybox:1.36
    command: ["sh","-c"]
    args: ["ls -la /etc/config; cat /etc/config/LOG_LEVEL; sleep 3600"]
    volumeMounts:
    - name: cfg
```

```
      mountPath: /etc/config
```

3) Sık Kullanılan Alanlar (Kısa Açıklamalar)

Alan	Ne işe yarar?
data	Key/value konfigürasyon içeriği.
binaryData	Binary içerik (base64) – nadir.
envFrom.configMapRef	Tüm key’leri env olarak ekler.
env.valueFrom.configMapKeyRef	Tek bir key’i env olarak ekler.
volumes.configMap	Config’i dosya olarak mount eder.
items	Sadece belirli key’leri seçip dosya adı verebilirsiniz.

4) En Sık Kullanılan Komutlar ve Kullanımları

Aşağıdaki komutlar CKAD pratiklerinde en çok kullanılanlardır.

4.1 Oluşturma (literal / file)

Hızlıca ConfigMap üretir.

```
kubectl create configmap app-config --from-literal=LOG_LEVEL=info --from-literal=FEATURE_X=true
kubectl create configmap app-config --from-file=app.properties
```

4.2 Görüntüleme

Key’leri ve içeriği doğrular.

```
kubectl get cm
kubectl describe cm app-config
kubectl get cm app-config -o yaml
```

4.3 Pod içinde doğrulama

Env veya dosya mount’unu kontrol edersiniz.

```
kubectl exec -it web -- env | grep LOG_LEVEL
kubectl exec -it app -- ls -la /etc/config
kubectl exec -it app -- cat /etc/config/LOG_LEVEL
```

4.4 Güncelleme (yaklaşım)

Genelde YAML ile apply edilir; create --dry-run ile YAML üretebilirsiniz.

```
kubectl create configmap app-config --from-literal=LOG_LEVEL=debug --dry-run=client -o yaml | ku
```

5) Troubleshooting Hızlı Rehber

Env gelmiyor: ConfigMap adı doğru mu? Pod YAML’ında envFrom / keyRef doğru mu?Dosya mount boş: volume + volumeMount isimleri eşleşiyor mu?Uygulama yeni değeri almıyor: Env ile verilmişse Pod restart gerekebilir; mount edilmiş dosyada davranış uygulamaya bağlıdır.

6) CKAD İpuçları

Hızlı YAML üret: kubectl create configmap ... --dry-run=client -o yamlEn çok hata: isim/namespace yanlışlığı. -n ile doğrula.Dosya mount için: volumes + volumeMount eşleşmesini kontrol et.