

# CKAD Ders Dökümanı #02 — Deployment

Dil: Türkçe • CKAD Kategorisi: Application Deployment (20%) • Odak: Konu anlatımı + en sık kullanılan komutlar

## 1) Konu Anlatımı

Deployment, stateless uygulamaları yönetmek için kullanılan en yaygın Kubernetes controller'ıdır. Arka planda ReplicaSet oluşturarak Pod replikalarını yönetir ve rolling update / rollback gibi rollout özellikleri sağlar.

Sınavda genellikle sizden: image güncelleme, rollout izleme, hatalı versiyondan geri dönme (rollback), replica sayısını değiştirme ve doğru label/selector kurgusunu yapmanız beklenir.

## 2) En Sık Karşılaşılan YAML Örnekleri

### Basit bir Deployment (tek container)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web
  labels:
    app: web
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx
          image: nginx:1.25
          ports:
            - containerPort: 80
```

### RollingUpdate stratejisi (maxSurge/maxUnavailable)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api
spec:
  replicas: 4
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
```

```
    maxUnavailable: 0
  selector:
    matchLabels:
      app: api
  template:
    metadata:
      labels:
        app: api
  spec:
    containers:
      - name: api
        image: myrepo/api:1.0.0
```

### 3) Sık Kullanılan Alanlar (Kısa Açıklamalar)

Alan	Ne işe yarar?
spec.replicas	Çalışacak Pod sayısı (desired replicas).
spec.selector.matchLabels	Deployment'ın yöneteceği Pod'ları seçer; template labels ile uyumlu olmalıdır.
spec.template	Pod şablonu: metadata.labels + spec.containers vb.
spec.strategy	Update stratejisi (RollingUpdate en yaygın).
spec.strategy.rollingUpdate.*	Rollout sırasında ek Pod sayısı / aynı anda down olabilecek Pod sayısı.
spec.revisionHistoryLimit	Rollback için tutulacak eski ReplicaSet sayısı.

## 4) En Sık Kullanılan kubectl Komutları

Aşağıdaki komutlar CKAD'de pratikte en çok kullanılan komutlardır. Her komut için amaç ve örnekler verilmiştir.

### 4.1 Deployment oluşturma (imperative yaklaşım)

Sınavda hızlı başlangıç için önce Pod/Deployment iskeleti üretip YAML'a geçebilirsiniz. Not: 'kubectl create deployment' basittir ama rollingUpdate gibi ayarlar için YAML düzenlemek gerekir.

```
# Deployment oluştur
kubectl create deployment web --image=nginx:1.25
```

```
# YAML üret (oluşturmadan)
kubectl create deployment web --image=nginx:1.25 --dry-run=client -o yaml
```

### 4.2 Ölçekleme (replica)

Replica sayısını hızlıca değiştirir.

```
kubectl scale deployment web --replicas=5
```

### 4.3 Image güncelleme (set image)

Rollout tetiklemek için en sık yöntem.

```
kubectl set image deployment/web nginx=nginx:1.26
```

```
# Birden fazla container varsa:
kubectl set image deployment/web api=myrepo/api:2.0 sidecar=myrepo/sidecar:1.2
```

### 4.4 Rollout izleme

Rollout durumunu ve geçmişini takip eder.

```
kubectl rollout status deployment/web
kubectl rollout history deployment/web
```

### 4.5 Rollback

Hatalı rollout sonrası önceki revision'a döner.

```
kubectl rollout undo deployment/web
```

```
# Belirli revision'a dön
kubectl rollout undo deployment/web --to-revision=2
```

### 4.6 Deployment/ReplicaSet/Pod ilişkisini görme

Hangi ReplicaSet'in aktif olduğunu ve Pod'ları doğrulamak için kullanılır.

```
kubectl get deploy,rs,pod -l app=web
kubectl describe deployment web
```

## 5) Troubleshooting Hızlı Rehber

Rollout takıldı: kubectl rollout status ile bak; ardından kubectl describe + kubectl get rs event'lerini incele.Yeni Pod'lar Ready olmuyor: readiness probe, imagePull hatası, config eksikliği; kubectl logs ve describe pod.Selector/labels uyuşmuyor: Deployment selector ile template labels aynı ailede olmalı; aksi halde Pod yönetimi bozulur.

## 6) CKAD İpuçları

Image değişikliğini en hızlı: `kubectl set imageRollout` kontrolü: `kubectl rollout status`  
Geri dönüş: `kubectl rollout undo` (gerekirse `revision` ile) ilişkileri görmek için:  
`kubectl get deploy,rs,pod -l ...`