

# İleri Seviye CI/CD Eğitimi ve Kubernetes, Docker, OpenShift Deployment



[Download PDF Version](#)

## Eğitim Süresi

- **Format 1**
  - **5 Gün**
  - **Ders Süresi:** 50 dakika
  - **Eğitim Saati:** 10:00 - 17:00
- **Format 2**
  - **12 Gün**
  - **Ders Süresi:** 50 dakika
  - **Eğitim Saati:** 10:00 - 17:00
- Her iki eğitim formatında eğitimler 50 dakika + 10 dakika moladır. 12:00-13:00 saatleri arasında 1 saat yemek arasındaki verilir. Günde toplam 6 saat eğitim verilir. 5 günlük formatta 30 saat eğitim, 12 günlük formatta toplam 72 saat eğitim verilmektedir. 12 saatlik eğitimde katılımcılar kod yazar ve eğitmenle birlikte sorulan sorulara ve taleplere uygun içerikler ve örnekler çalışılır.
- Eğitimler uzaktan eğitim formatında tasarlanmıştır. Her eğitim için teams linkleri gönderilir. Katılımcılar bu linklere girerek eğitimlere katıllırlar. Ayrıca farklı remote çalışma araçları da eğitmen tarafından tüm katılımlara sunulur. Katılımcılar bu araçları kullanarak eğitimlere katılırlar.

- Eğitim içeriğinde github ve codespace kullanılır. Katılımcılar bu platformlar üzerinden örnek projeler oluşturur ve eğitimmenle birlikte eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir. Katılımcılar bu araçlarla eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir.
- Eğitim yapay zeka destekli kendi kendine öğrenme formasyonu ile tasarlanmıştır. Katılımcılar eğitim boyunca kendi kendine öğrenme formasyonu ile eğitimlere katılır. Bu eğitim formatı sayesinde tüm katılımcılar gelecek tüm yaşamlarında kendilerini güncellemeye devam edebilecekler ve her türlü sorunun karşısında çözüm bulabilecekleri yeteneklere sahip olacaklardır.

## İleri Seviye CI/CD Eğitimi

Yazılım geliştirme süreçlerinde verimliliği artırmak ve güvenliği sağlamak için kritik öneme sahip **CI/CD** (Sürekli Entegrasyon/Sürekli Dağıtım), günümüzde vazgeçilmez bir araçtır. Bu ileri seviye eğitim, **Docker**, **Kubernetes** ve **OpenShift** gibi container teknolojilerini kullanarak CI/CD süreçlerini daha etkili ve güvenli hale getirmeyi hedefleyen yazılım geliştiriciler, DevOps mühendisleri ve sistem yöneticileri için kapsamlı bir rehber sunmaktadır.

Katılımcılar, popüler CI/CD araçlarıyla süreçleri optimize ederken, yüksek güvenlikli, ölçeklenebilir ve sürdürülebilir yazılım dağıtım süreçlerini öğrenir. Ayrıca gerçek dünya vaka çalışmalarıyla, teoriyi практиge dökme imkanı bulurlar.

Bu eğitim, yazılım geliştirme ve dağıtım süreçlerini daha hızlı, güvenli ve verimli hale getirmek isteyen profesyoneller için ideal bir fırsatır. **CI/CD ve container teknolojilerinde yetkinlik kazanmak için bu eğitime katılın!**

## Eğitim Hedefi

Bu ileri seviye eğitim, katılımcılara **CI/CD** süreçlerini daha etkin bir şekilde yönetmek ve otomatikleştirmek için gereken derinlemesine bilgi ve becerileri kazandırmayı hedeflemektedir. Katılımcılar, **Docker**, **Kubernetes** ve **OpenShift** gibi modern container teknolojilerini kullanarak güvenli, ölçeklenebilir ve sürdürülebilir yazılım dağıtım süreçlerini nasıl tasarlayacaklarını öğrenirler.

### Eğitimdeki ana hedefler şunlardır

1. **CI/CD Süreçlerini Derinlemesine Anlamak:** Katılımcılar, CI/CD süreçlerinin temellerinden ileri düzey uygulamalarına kadar geniş bir bilgi birikimi elde ederler. Popüler CI/CD araçlarını (Jenkins, GitLab CI, CircleCI vb.) etkin bir şekilde kullanarak yazılım geliştirme ve dağıtım süreçlerini optimize ederler.
2. **Container Teknolojileri ile Verimliliği Artırmak:** Docker, Kubernetes ve OpenShift ile container'ların yönetimini, orkestrasyonunu ve dağıtmını öğrenerek uygulama geliştirme süreçlerini daha güvenli ve verimli hale getirirler.
3. **Yüksek Güvenlikli ve Ölçeklenebilir Dağıtımlar Tasarlamak:** Güvenlik ve ölçeklenebilirlik odaklı dağıtım süreçleri geliştirerek, büyük ölçekli sistemlerde yüksek erişilebilirlik sağlarlar.
4. **Gerçek Dünya Uygulamalarını Pratikte Uygulamak:** Katılımcılar, gerçek dünya vaka çalışmalarıyla öğrendikleri teoriyi практиge dökme fırsatı bulurlar.

# Eğitim İçeriği

## DevOps Kültürü ve CI/CD Temelleri

- **DevOps Kültürüne Derinlemesine İncelenmesi:** İşbirliği, otomasyon, sürekli iyileştirme ve geri bildirim döngülerinin yazılım geliştirmedeki önemi.
- **CI/CD'nin Yazılım Geliştirme Yaşam Döngüsündeki Yeri:** Yazılım geliştirme süreçlerinin hızlandırılması, sürekli test ve sürekli dağıtımın yazılım kalitesine etkisi.
- **CI/CD Araçları ve Entegrasyonları:** Jenkins, GitLab CI, CircleCI gibi araçların detaylı kullanımı ve entegrasyonu. Ayrıca, GitOps yaklaşımları ve nasıl otomatikleştirileceği.

## Jenkins ile İleri Düzey CI/CD Süreç Yönetimi

- **Jenkins Pipeline Derinlemesine İncelemesi:** Declarative ve Scripted pipeline yapılarını detaylı inceleme ve gerçek dünya uygulamalarına uygun pipeline tasarımları.
- **Jenkins ile Dinamik Test Süreçleri:** Jenkins üzerinde test otomasyonu ve entegrasyonu; test öncesi ve sonrası scriptler ile otomatik testlerin yönetilmesi.
- **Jenkins ile Çoklu Dağıtım Stratejileri:** Jenkins'in Kubernetes ve Docker ile entegrasyonu, mikro servisler için çoklu dağıtım süreçleri.

## Docker ve Kubernetes ile Container Orkestrasyonu ve CI/CD Entegrasyonu

- **Docker ile Konteynerleştirme Stratejileri:** Mikro servisler için container oluşturma, Docker imajlarının optimize edilmesi ve güvenlik kontrolleri (Image Scanning).
- **Docker ve Kubernetes Entegrasyonu:** Docker container'larının Kubernetes ortamında yönetilmesi, Pod ve Deployment kullanımı ile uygulama dağıtıımı.
- **CI/CD Süreçlerinde Kubernetes Kullanımı:** Kubernetes üzerinde otomatik ölçeklendirme, yüksek erişilebilirlik (HA) ve hata tolere edebilirlik. CI/CD süreçlerinde Kubernetes'te rollback, rollout ve canlı izleme işlemleri.

## Kubernetes İleri Düzey Kullanımı ve Senaryolar

- **Kubernetes İle Yüksek Erişilebilirlik ve Hata Yönetimi:** Kubernetes'in ReplicaSets, StatefulSets, Persistent Storage ve ConfigMaps ile yüksek erişilebilirliğe sahip uygulama dağıtımları.
- **Kubernetes ve Helm ile Konfigürasyon Yönetimi:** Helm chart'ları ile Kubernetes uygulama dağıtımları ve sürüm yönetimi. Helm'i CI/CD süreçlerine entegre etme.
- **Kubernetes'te CI/CD Otomasyonu:** Kubernetes'in CI/CD süreçlerinde kullanımı için araçlar, örneğin ArgoCD, Spinnaker, Jenkins X. Gerçek dünya senaryolarında bu araçlar ile süreçlerin otomatikleştirilmesi.

## OpenShift ile Enterprise Seviye Dağıtım ve Yönetim

- **OpenShift'in Özellikleri ve DevOps'a Entegrasyonu:** OpenShift'in Kubernetes ile olan derin entegrasyonu, CI/CD süreçlerinde OpenShift'in avantajları.
- **OpenShift Dağıtımları:** OpenShift üzerinde mikro servis tabanlı uygulamaların dağıtıımı, yönetimi ve izlenmesi.
- **OpenShift İle Güvenlik ve Performans Yönetimi:** OpenShift üzerinde güvenlik, erişim kontrolü, izinler ve izleme süreçlerinin DevOps kültürü ile entegrasyonu.

## Gerçek Dünya Senaryoları ve Vaka Çalışmaları

- **Özelleştirilmiş Senaryolar ile CI/CD Tasarımı ve Uygulama:**

- **Senaryo 1:** Mikro servis mimarisi ile çoklu container uygulamaları için CI/CD pipeline tasarımları. Jenkins ve GitLab CI kullanarak birden fazla servisin entegre olduğu dağıtım süreçleri.
- **Senaryo 2:** Kubernetes üzerinde ölçeklendirilmiş ve yüksek erişilebilirliğe sahip bir uygulamanın CI/CD entegrasyonu. Pod'lar, ReplicaSets, ve StatefulSets ile gerçek zamanlı dağıtım yönetimi.
- **Senaryo 3:** Docker ve Kubernetes ile arka planda çalışan hizmetler için otomatik test süreçleri ve dağıtım sonrası validasyon adımları.

- **DevOps Kültürüne Takım ve Süreçlere Entegrasyonu:**

- Sürekli iyileştirme ile DevOps ekibi işbirliği ve etkileşiminin optimize edilmesi. Ekip içi geri bildirim döngülerinin CI/CD süreçlerine etkisi.
- **Vaka Çalışması:** Bir organizasyonun DevOps ve CI/CD süreçlerini uygulayarak yazılım teslimat süreçlerinin nasıl hızlandırılabileceği. Gerçek organizasyonel değişiklikler ile CI/CD ve DevOps süreçlerinin entegrasyonu.

## CI/CD ve DevOps Süreçlerinde Güvenlik (DevSecOps)

- **DevSecOps ile Güvenlik Otomasyonu:** Güvenlik testlerinin CI/CD pipeline'larına entegre edilmesi. GitLab CI, Jenkins veya CircleCI üzerinde güvenlik testlerinin otomatikleştirilmesi.
- **Kapsayıcılar İçin Güvenlik Stratejileri:** Docker container'ları için güvenlik taramaları, imaj güvenliği, şifreleme ve kimlik doğrulama. Kubernetes ve OpenShift üzerinde container güvenliği.
- **Güvenlik İzleme ve Olay Müdahale:** Gerçek zamanlı güvenlik izleme araçları kullanılarak container, Kubernetes ve OpenShift ortamlarının güvenliğinin sağlanması.

## Eğitim Sonrası Kazanımlar:

- **DevOps Kültürüne Tam Hakimiyet:** Yazılım geliştirme süreçlerinde sürekli entegrasyon, sürekli dağıtım ve otomasyonun DevOps ile entegrasyonu.
- **Container Yönetimi ve Orkestrasyonu:** Docker, Kubernetes ve OpenShift gibi teknolojilerle uygulama yönetimi ve yüksek erişilebilirlik sağlama.
- **Güvenli ve Verimli CI/CD Süreçleri:** CI/CD pipeline'larının güvenlik entegrasyonları, otomasyon süreçleri ve güvenlik testleri ile birlikte hızlandırılması.
- **Gerçek Dünya Senaryolarında Deneyim:** Uygulama geliştirme ve dağıtım süreçlerinin gerçek dünya senaryoları ile entegre edilmesi ve pratikte uygulanabilir çözümler üretilmesi.

## Eğitim Yöntemi

- **Teorik Bilgi:** Güncel bilgiler ve konseptlerin anlatımı.
- **Uygulamalı Örnekler:** Gerçek senaryolarla pratik uygulamalar.
- **Etkileşimli Tartışmalar:** Katılımcıların fikirlerini paylaşabileceği oturumlar.

## Hedef Kitle

- Yazılım geliştirme ekipleri
- DevOps mühendisleri
- Sistem ve ağ yöneticileri
- Proje yöneticileri ve teknik liderler
- CI/CD süreçlerine ilgi duyan tüm profesyoneller

## Katılımcılardan Beklentilerimiz

- Katılımcıların temel programlama ve sistem yönetimi bilgisine sahip olmaları.
- CI/CD ve konteyner teknolojilerine ilgi duymaları.
- Takım çalışmasına yatkın olmaları ve aktif katılım göstermeye istekli olmaları.

[Eğitim Materyalleri \(Eğitmenlere Özel\)](#)