

# ARGO CD

Modern yazılım geliştirme dünyasında hız, güvenlik ve otomasyon her zamankinden daha kritik. Ekipler, uygulamalarını daha hızlı ve güvenilir bir şekilde dağıtmak için yeni yollar arıyor. İşte tam da bu noktada GitOps devreye giriyor ve Kubernetes ekosisteminin önde gelen aracı Argo CD, bu felsefeyi gerçeğe dönüştürüyor.

Argo CD, Kubernetes-native bir declarative GitOps aracıdır. Peki bu ne anlama geliyor? Kısaca, uygulamalarınızın ve altyapınızın istenen durumunu Git repolarınızda tanımlarsınız ve Argo CD, Kubernetes kümenizin bu duruma otomatik olarak ulaşmasını sağlar. Artık manuel müdahalelere, karmaşık scriptlere veya belirsiz dağıtım süreçlerine son verebilirsiniz.

## ARGO CD

### KURUMSAL ÖZEL EĞİTİM

Bu konuyu istediğiniz gibi özelleştirebilirsiniz. Kullandığınız teknolojiye özel uygulamalar ile anlatılmasını, projenize uygun içerik haline getirilmesini ...



## ARGO CD EĞİTİMİ

### Neden Argo CD?

- **Tek Doğruluk Kaynağı (Single Source of Truth):** Tüm uygulama ve altyapı konfigürasyonlarınız Git'te versiyonlanır. Bu, değişikliklerin izlenebilir, geri alınabilir ve denetlenebilir olmasını sağlar.
- **Otomatik Senkronizasyon:** Git repodaki değişiklikler algılandığında, Argo CD otomatik olarak hedef Kubernetes kümesini günceller. Bu, dağıtım süreçlerinizi hızlandırır ve insan hatasını minimize eder.
- **Durum Gözlemi ve Otomatik İyileşme (Self-Healing):** Argo CD, Kubernetes kümenizdeki canlı durumu sürekli olarak Git'teki istenen durumla karşılaştırır. Herhangi bir sapma (drift) algılandığında, otomatik olarak düzeltme yaparak sistemlerinizin her zaman istenen durumda kalmasını sağlar.
- **Anlaşılır Kullanıcı Arayüzü (UI) ve CLI:** Sezgisel web arayüzü sayesinde uygulamalarınızın sağlık durumunu, senkronizasyon durumunu ve kaynak ilişkilerini kolayca görselleştirebilirsiniz. Güçlü CLI ise otomasyon ve scripting için tam kontrol sunar.
- **Esnek Konfigürasyon Yönetimi:** Helm, Kustomize veya düz YAML dosyaları gibi popüler Kubernetes konfigürasyon araçlarıyla sorunsuz entegre olur. Mevcut araçlarınızı kullanmaya devam edebilirsiniz.
- **Gelişmiş Özellikler:** ApplicationSet ile birden çok uygulamayı merkezi olarak yönetme, senkronizasyon hook'ları ile dağıtım öncesi/sonrası işlemler yapma, detaylı bildirimler ve kapsamlı RBAC (Rol Tabanlı Erişim Kontrolü) gibi özelliklerle karmaşık senaryoları bile kolayca yönetirsiniz.





## Modül 1: GitOps ve Kubernetes Temelleri - Argo CD ye Giriş

### GitOps Prensipleri:

- GitOps felsefesi ve temel prensipleri.
- Versiyon kontrol sistemini (Git) tek doğruluk kaynağı olarak kullanmanın önemi ve faydaları.
- Declarative (bildirimsel) ve Imperative (emredici) yaklaşımlar arasındaki farklar ve GitOps in Declarative doğası.
- GitOps un CI/CD süreçlerine entegrasyonu ve sağladığı avantajlar (hız, güvenlik, istikrar).

### Kubernetes Temelleri:

- Konteyner düzenleme platformu Kubernetes in temel mimarisi.
- .NET uygulamalarını Kubernetes te çalıştırmak için gerekli temel kavramlar: Pod, Deployment, Service, Namespace.
- Kubernetes kaynaklarını tanımlamak için kullanılan YAML formatı.

### YAML (YAML Ain t Markup Language):

- YAML sözdizimi ve temel yapıları (.NET projeleri için K8s konfigürasyon dosyalarını okuma ve yazma).
- Kubernetes manifest dosyalarında YAML kullanımı: Deployment, Service, ConfigMap, Secret örnekleri.

### Kaynak Kontrol Sistemleri (Git):

- Git in temel komutları ve iş akışları (.NET projelerinin versiyon kontrolüyle entegrasyonu).
- Git branch stratejileri (örneğin, feature branch, develop, main) ve GitOps ile ilişkisi.
- GitHub, GitLab, Azure Repos gibi platformlarda Git repolarının yapısı.

## Modül 2: Argo CD Mimarisi ve Temel Uygulama Yönetimi

### Argo CD Mimarisine Genel Bakış:

- Argo CD sisteminin ana bileşenleri: API Server, Controller, Repo Server.
- Bu bileşenlerin birbiriyle ve Kubernetes API si ile nasıl etkileşimde bulunduğu.
- Argo CD nin Git reposunu nasıl izlediği ve değişiklikleri nasıl algıladığı.

### Argo CD Application Kaynağı:

- Argo CD deki ana kaynak türü olan Application objesi.
- Bir Application kaynağının tanımı: Git repo URL si, hedef Kubernetes kümesi, senkronize edilecek yol (path).
- .NET projeleriniz için bir Application kaynağı YAML dosyasını oluşturma adımları (hands-on).

### Senkronizasyon (Sync) Süreci:

- Argo CD nin Git te tanımlı istenen durumu Kubernetes kümesine uygulama mekanizması.
- Otomatik (Auto-Sync) ve manuel (Manual Sync) senkronizasyon modları.
- Senkronizasyonun tetiklenme şekilleri (webhook, polling, CLI).
- Pratik: Basit bir .NET uygulamasının ilk deployment ı için Argo CD Application kaynağını oluşturma ve senkronize etme (hands-on).

### Argo CD Kullanıcı Arayüzü (UI):

- Argo CD UI na erişim ve arayüzde gezinme.
- Uygulama durumunu (sync status, health status) ve kaynak ilişkilerini görselleştirme.
- UI üzerinden senkronizasyon tetikleme ve yönetme.
- Pratik: Dağıtılan .NET uygulamasının durumunu UI üzerinden izleme (hands-on).

## Modül 3: Senkronizasyon Stratejileri ve Sağlık Kontrolleri

### Sağlık Kontrolleri (Health Checks):

- Argo CD nin dağıtılan Kubernetes kaynaklarının (Deployment, Pod, Service vb.) sağlık durumunu nasıl belirlediği.
- Yaygın kaynak türleri için varsayılan sağlık kontrolleri.
- Bir uygulamanın "Healthy" durumuna geçişini izleme.

### Geri Alma (Rollback):

- Hatalı veya istenmeyen dağıtımları hızlıca önceki stabil sürümlere döndürme yeteneği.
- Argo CD UI ve CLI üzerinden rollback işlemleri.
- Pratik: Uygulama konfigürasyonunda hata yapma ve Argo CD kullanarak geri alma senaryosu (hands-on).NET uygulamanızın farklı ortam konfigürasyonları için Kustomize kullanma).
- Pratik: Aynı .NET uygulamasının farklı ortamlar için Kustomize konfigürasyonlarını hazırlama ve Argo CD ile yönetme (hands-on).

### Gelişmiş Senkronizasyon Seçenekleri:

- Prune: Git te artık bulunmayan kaynakların kümeden otomatik olarak silinmesi.
- Self-Heal: Kümedeki canlı durumun Git ten farklılaştığında (drifting) Argo CD nin otomatik olarak düzeltme yapması.
- Sync Options: --replace=true, --force=true gibi senkronizasyon davranışını etkileyen seçenekler.
- Pratik: Self-Heal özelliğini test etme, kümede manuel değişiklik yapıp Argo CD nin nasıl davrandığını gözlemleme (hands-on).

## Modül 4: Konfigürasyon Yönetim Araçları Entegrasyonu

### Helm ile Entegrasyon:

- Kubernetes uygulamalarını paketlemek için popüler bir araç olan Helm in temelleri.
- Argo CD nin Helm Chart ları kullanarak uygulama dağıtımı.
- Value dosyaları ile Helm Chart ları özelleştirme (.NET uygulamanız için Helm Chart oluşturma ve Argo CD ile dağıtma senaryoları).
- Pratik: Basit bir .NET Core uygulamasının Helm Chart ını hazırlama ve Argo CD Application kaynağını bu Chart ı gösterecek şekilde yapılandırma (hands-on).

### Kustomize ile Entegrasyon:

- Uygulama konfigürasyonlarını şablonuz ve overlay (katmanlama) yaklaşımıyla özelleştirme aracı Kustomize.
- Argo CD nin Kustomize izinlerini kullanarak uygulama dağıtımı.
- Development, Staging, Production gibi farklı ortamlar için Kustomize overlay leri oluşturma ve yönetme (.NET uygulamanızın farklı ortam konfigürasyonları için Kustomize kullanma).
- Pratik: Aynı .NET uygulamasının farklı ortamlar için Kustomize konfigürasyonlarını hazırlama ve Argo CD ile yönetme (hands-on).

### Düz YAML Dizileri ile Entegrasyon:

- En temel yöntem: Kubernetes manifest dosyalarını içeren bir Git dizinini doğrudan senkronize etme.
- Argo CD nin bu tür izinleri nasıl işlediği.
- Pratik: Klasik Kubernetes YAML dosyaları ile bir .NET uygulamasını dağıtma (hands-on).



## Modül 5: Argo CD CLI, Projeler ve Temel RBAC

### Argo CD CLI Kullanımı:

- Argo CD komut satırı arayüzünü kurma ve yapılandırma.
- CLI üzerinden Argo CD sunucusuna kimlik doğrulama (argocd login).
- CLI kullanarak uygulama oluşturma, silme, güncelleme, senkronize etme (argocd app create, argocd app sync, argocd app list).
- CLI ile uygulama durumunu sorgulama ve detayları inceleme (argocd app get, argocd app logs).
- Pratik: Önceki modüllerde dağıtılan uygulamaları CLI üzerinden yönetme pratikleri (hands-on).

### Argo CD Projeleri:

- Uygulamaları mantıksal gruplara ayırmak için Argo CD Project kavramı.
- Projelerin kullanımı: Uygulama, hedef küme ve kaynak kısıtlamaları tanımlama.
- Çok kullanıcı veya çok ekipli ortamlarda projelerin önemi.
- Pratik: Farklı ekipler veya .NET uygulamaları için Argo CD projeleri oluşturma (hands-on).

### Rol Tabanlı Erişim Kontrolü (RBAC) Temelleri:

- Argo CD içerisinde farklı kullanıcı ve gruplar için yetkilendirme yapısı.
- Argo CD nin dahili RBAC mekanizması.
- Kullanıcı ve grup tanımlama (SSO entegrasyonlarına genel bakış).
- Proje bazlı RBAC kuralları tanımlama örnekleri (bir kullanıcının sadece belirli bir projede uygulama oluşturmaya izin verme).
- Pratik: Temel RBAC kuralları yapılandırma ve farklı kullanıcılarla test etme (hands-on).

## Modül 6: Gelişmiş Yönetim ve Sorun Giderme

### ApplicationSet Controller:

- Birden çok benzer Argo CD Application kaynağını tek bir kaynaktan (ApplicationSet) yönetme.
- Kullanım senaryoları: Monorepo içindeki birden çok dizin, birden çok küme, Git organizasyonları/kullanıcıları.
- ApplicationSet generator ları (Git, Cluster, List, Matrix vb.).
- Pratik: Monorepo yapısındaki birden çok .NET microservice i tek bir ApplicationSet ile dağıtma (hands-on).

### Bildirimler (Notifications):

- Argo CD olayları (senkronizasyon başarısı/hatası, sağlık durumu değişiklikleri vb.) hakkında bildirim yapılandırması.
- Bildirim hedefleri: E-posta, Slack, Microsoft Teams, Webhooks.
- Bildirim şablonları ve yapılandırma detayları.
- Pratik: Argo CD bildirimlerini bir Slack kanalına gönderecek şekilde yapılandırma (hands-on).

### Temel Senkronizasyon Sorun Giderme:

- Sık karşılaşılan senkronizasyon hataları ve nedenleri.
- Argo CD loglarını inceleme ve hata mesajlarını anlama.
- argocd app sync ve argocd app diff gibi komutlarla sorun tespiti.
- Canlı durum (live state) ve istenen durum (desired state) arasındaki farkları anlama.
- Pratik: Kasıtlı olarak senkronizasyon hatası yaratma ve bu hataları giderme (hands-on).

### Kaynak Kayması (Drifting) Tespiti:

- Kubernetes kümesindeki canlı durumun Git te tanımlı durumdan nasıl farklılaşabileceği (manuel değişiklikler vb.).
- Argo CD nin bu kaymaları nasıl tespit ettiği ve UI da nasıl raporladığı.
- Drift durumlarını düzeltme (sync veya manual correction).

## Modül 7: CI/CD Entegrasyonu ve İleri Konulara Bakış

### CI/CD Süreçleriyle Argo CD Entegrasyonu:

- Geleneksel CI (Sürekli Entegrasyon) pipeline ları ile GitOps/Argo CD nin birleşimi.
- CI pipeline ı tamamlandıktan sonra Argo CD senkronizasyonunu tetikleme yöntemleri:
  - Image Updater kullanımı (Container imaj etiketlerindeki değişikliği takip etme).
  - Argo CD API veya CLI kullanarak senkronizasyon tetikleme (CI pipeline son adımında).
  - Git commit/push sonrası webhook ile otomatik tetikleme.
- .NET projelerinin build ve test aşamalarının ardından Argo CD ile dağıtımın nasıl otomatikleştirileceği senaryoları.
- Pratik: Bir CI simülasyonu yaparak Argo CD senkronizasyonunu otomatik tetikleme (hands-on senaryo).

### Kaynak Hookları (Resource Hooks):

- Senkronizasyon sürecinin belirli aşamalarında (PreSync, Sync, PostSync, Skip, Ignore) özel işlemler çalıştırma.
- Hook türleri (Job, Pod, vb.) ve kullanım alanları (veritabanı migrasyonları, kurulum scriptleri vb.).
- Hook yaşam döngüsü ve hata yönetimi.
- Pratik: Bir .NET uygulamasının veritabanı migrasyonu için PreSync Job hook u tanımlama (hands-on).

### Argo CD Yüksek Erişilebilirlik (HA) ve Ölçeklendirme (Genel Bakış):

- Üretim ortamları için Argo CD bileşenlerinin yüksek erişilebilir dağıtımı (kavramsal).
- Artan iş yüküne göre Repo Server ve Application Controller ölçeklendirme stratejileri (kavramsal).

### Güvenlik Hususları (Genel Bakış):

- Argo CD kurulumunun güvenliği.
- Secret yönetimi entegrasyonları (Kubernetes Secrets, Vault vb.).
- Güvenli konfigürasyon desenleri.

### Özel Sağlık Kontrolleri ve Kaynaklar (Genel Bakış):

- Varsayılan sağlık kontrollerinin yeterli olmadığı durumlar.
- Custom Resource Definitions (CRD) ve Argo CD nin bunları yönetme yeteneği (kavramsal).



## EĞİTİM SÜRESİ

- 3 Gün
- Ders Süresi: 50 dakika
- Eğitim Saati: 10:00 - 17:00

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. 3 günlük formatta 18 saat eğitim verilmektedir.

Eğitimler uzaktan eğitim formatında tasarlanmıştır. Her eğitim için teams linkleri gönderilir. Katılımcılar bu linklere girerek eğitimlere katılırlar. 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 eğitmenle 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 katılırlar. Bu eğitim formatı sayesinde tüm katılımcılar gelecek tüm yaşamlarında kendilerini güncellemeye devam edebilecekler ve her türlü sorunun karşısında çözüm bulabilecekleri yeteneklere sahip olacaklardır.

## ARGO CD



Argo CD, CI/CD pipeline'larınızın son halkasını oluşturarak sürekli entegrasyonu (CI) sürekli dağıtım (CD) ile birleştirir. Kodunuz Git'e commit edildiğinde başlayan otomasyon zinciri, uygulamanızın üretimde canlıya alınmasıyla sorunsuz bir şekilde devam eder. Bu sayede ekipleriniz daha hızlı iterasyon yapabilir, yeni özellikleri daha çabuk müşterilere ulaştırabilir ve operasyonel yükü azaltabilir.

## KATILIMCILARDAN BEKLENTİLERİMİZ

- **Temel Kubernetes Bilgisi:** Eğitime başlamadan önce Kubernetes'in temel kavramlarına (Pod, Deployment, Service, Namespace) aşina olmanız, konuları daha hızlı kavramanıza yardımcı olacaktır.
- **Temel Git Kullanımı:** Git versiyon kontrol sistemi ve temel komutları (commit, push, pull, branch) hakkında bilgi sahibi olmanız, GitOps prensiplerini anlamanızı kolaylaştıracaktır.
- **YAML Okuma ve Anlama:** Kubernetes kaynaklarını tanımlamada yaygın olarak kullanılan YAML dosyalarını okuma ve temel yapısını anlama becerisine sahip olmanız önemlidir.
- **Temel .NET C# Geliştirme Deneyimi:** Eğitim, .NET C# Geliştirme Uzmanları profiline yönelik olduğundan, temel seviyede .NET C# uygulama geliştirme ve projelerin derlenmesi süreçlerine hakim olmanız, örnek senaryoları daha iyi anlamanızı sağlayacaktır.
- **Aktif Katılım ve Uygulama Pratiği:** Eğitim süresince yer verilecek hands-on (uygulamalı) çalışmalara aktif olarak katılmanız ve verilen senaryoları bizzat uygulamanız, teorik bilginin pratik beceriye dönüşmesi için kritik öneme sahiptir. Eğitmenimiz, her katılımcının ilerlemesini yakından takip edecek ve destekleyecektir.

## EĞİTİM YÖNTEMİ

- **Teorik Bilgi:** Eğitimin temelini oluşturan konularda (örneğin, Argo CD'nin mimarisi, GitOps prensipleri ve temel bileşenleri) kapsamlı ve güncel bilgi sunulacaktır. Her modül, katılımcıların konuyu derinlemesine anlamasını sağlayacak şekilde teorik altyapıyı güçlü bir şekilde işleyecektir.
- **Uygulamalı Örnekler:** Katılımcılar, eğitim süresince gerçek hayattan alınan senaryolar üzerinden pratik uygulamalar geliştireceklerdir. .NET projelerine özel hands-on çalışmalarla, teorik bilgiyi doğrudan pratiğe dökme ve Argo CD ile deployment yönetimi becerilerini pekiştirme fırsatı bulacaklardır.
- **Proje Tabanlı Öğrenme:** Eğitim boyunca edinilen bilgiler, katılımcıların proaktif bir Argo CD projesi üzerinde çalışarak pekiştirilecektir. Bu, gerçek dünya problemlerine çözüm üretme yeteneğini geliştirecek ve öğrenilenleri kalıcı hale getirecektir.

## HEDEF KİTLE

- **.NET C# Geliştirme Uzmanları:** Kubernetes ortamlarında .NET uygulamalarının dağıtım ve yönetimini GitOps prensipleriyle otomatikleştirmek isteyen tüm geliştiriciler.
- **DevOps ve SRE Mühendisleri:** Dağıtım süreçlerini standartlaştırmak, otomatikleştirmek ve operasyonel verimliliği artırmak için GitOps araçlarını etkin kullanmak isteyen mühendisler.
- **Yazılım Mimarları:** Ölçeklenebilir, güvenilir ve sürdürülebilir dağıtım pipeline'ları tasarlayarak modern mimarileri GitOps ile güçlendirmeyi hedefleyen mimarlar.
- **Sistem Yöneticileri ve Operasyon Uzmanları:** Kubernetes kümeleri üzerindeki uygulama yaşam döngüsünü Argo CD ile daha etkin yönetmek ve sorun giderme yeteneklerini geliştirmek isteyen profesyoneller.

"

*Kurumsal size özel eğitimler hazırlıyoruz. Her eğitim yeni bir heyecan.*

## Vebende A.Ş.

Kurumsal Terzi Usulü Butik Eğitimler.

Size özel hazırlanan seminer, danışmanlık, eğitim ve hizmetlerimizle yüksek verim elde edin. Paranızı boşa harcamayın. Zaman çok değerli.

## Her Eğitim Yeni Bir Heyecan

2000 yılından günümüze devam eden eğitim heyecanı. Uluslararası tecrübe, proje geliştirme deneyimleri, danışmanlıklar ve arge mühendislik deneyimlerimizi ülkemize sunmak için yeni bir konsept tasarladık. Sizi dinliyor, takımınıza uygun özel içerikler ile hazırlanmış eğitimler hazırlıyoruz. Her eğitim özel bir çalışma, içerik üretimi, uygulama örnekleri hazırlıkları, sunumlar hazırlamayı gerektiriyor. Aynı eğitimi yönetim kadrosuna farklı, teknik ve mimar ekiplerinize farklı içerikler ile hazırlıyoruz. Her eğitim yeni bir macera ve heyecan.



## Bizimle İletişime Geçin

[iletisim@vebende.com.tr](mailto:iletisim@vebende.com.tr)

[www.vebende.com.tr](http://www.vebende.com.tr)

İzmir - Türkiye

## Kurumsal Eğitimler

[www.vebende.com.tr](http://www.vebende.com.tr)







## İletişime Geçin



Whatsapp: 0542 5505704  
Tel: 0532 512 7811



iletisim@vebende.com.tr



www.vebende.com.tr



Adalet Mah. Manas Bulvarı  
No:39 İç Kapı NO: 3107 Bayraklı  
İZMİR



Sürekli Güncellenen Eğitimlerimizi Sitemizden  
Takip Edin.

“

Kurumsal terzi usulu size özel  
eğitimler ve uluslararası deneyime  
sahip eğitmenler ile çalışmanın keyfi

”



## Size Neler Sunuyoruz?

- Kitaptan okunan eğitimler sunmuyoruz. Sizin için özel hazırlanan içerikler ve uygulamalı eğitimler hazırlıyoruz.
- Her eğitim için sunumlar, materyaller, örnek senaryolar size özel hazırlıyoruz.
- Her eğitim için katılımcılara özel github repoları hazırlanıyor. Eğitimden sonra da katılımcılar hayat boyu kaynaklara ulaşmaya devam edebilsinler diye, **“katılımcılar ve eğitmenle birlikte yapılan tüm çalışmaları”** github repolarında saklıyoruz.



## Misyonumuz

1

Ülkemizde dünyanın en güncel teknolojilerini kazandırmak. En güncel teknolojilerine hakim takımlarına katkı sağlamak.

2

Alışılmışın dışında en güncel teknolojileri deneyimli eğitmenler ve uygulamalar ile sunmak.

3

Siber güvenlik konusunda hassas çalışmalar sunarak, çağımızın büyük kabusu siber tehditler konusunda deneyimli bilgili takımların oluşmasına katkı sağlamak.