# GitLab Kullanım Dökümanı

Bu doküman, tamamen benim tarafımdan hazırlanmıştır ve GitLab’ı sıfırdan ileri seviyeye kadar öğrenmek isteyenler için kapsamlı bir rehber niteliğindedir. İçeriğinde adım adım yönergeler, ipuçları, örnek senaryolar, hata çözüm yöntemleri, güvenlik önlemleri ve performans optimizasyon tavsiyeleri yer almaktadır.

## 1. GitLab Nedir?

GitLab, Git tabanlı versiyon kontrol sistemi üzerine inşa edilmiş, yazılım geliştirme sürecinin planlama, kodlama, test, dağıtım ve izleme aşamalarını tek platformda yönetebileceğiniz bir DevOps çözümüdür. Hem bulut tabanlı (gitlab.com) hem de kendi sunucunuzda barındırabileceğiniz self-hosted versiyonları vardır.

**Öne çıkan özellikler:**

* Kod depolama ve versiyon kontrolü
* Issue (görev/hata) yönetimi
* Merge request ile kod inceleme
* CI/CD otomasyonu
* Güvenlik taramaları ve uyumluluk raporları

## 2. Hesap Oluşturma ve Giriş

1. <https://gitlab.com> adresinden hesap oluşturun.
2. Kullanıcı adı, e-posta ve güçlü bir şifre belirleyin.
3. E-posta adresinizi onaylayın.
4. **User Settings → SSH Keys** bölümünden SSH anahtarınızı ekleyin.

**İpucu:** SSH kullanarak her push/pull işleminde şifre girmek zorunda kalmazsınız.

## 3. Proje Oluşturma

1. **New Project → Create blank project** adımlarını izleyin.
2. Proje adını, açıklamasını girin ve erişim seviyesini (Private, Internal, Public) seçin.
3. İsterseniz README, .gitignore ve lisans ekleyin.
4. **Create Project** ile tamamlayın.

**Senaryo:** Bir ekip yeni bir web uygulaması geliştirmeye başlarken tüm görevleri issue’lar üzerinden tanımlar ve her geliştirici kendi branch’inde çalışır.

## 4. Depoyu Klonlama

HTTPS ile:

git clone https://gitlab.com/kullanici/proje.git

SSH ile:

git clone git@gitlab.com:kullanici/proje.git

**Not:** SSH kullanmak için önce anahtar oluşturmanız gerekir: ssh-keygen -t rsa -b 4096

## 5. Commit ve Push

1. Dosyalarınızı ekleyin:

git add .

1. Commit mesajı girin:

git commit -m "İlk commit"

1. Uzak sunucuya gönderin:

git push origin main

**Commit mesajları** kısa, anlamlı ve değişiklikleri özetler şekilde olmalıdır.

## 6. Branch Yönetimi

* Yeni branch oluşturma:

git checkout -b feature/ozellik

* Branch değiştirme:

git checkout main

* Branch silme:

git branch -d feature/ozellik

**İpucu:** Ana branch’te doğrudan çalışmak yerine her iş için ayrı branch açın.

## 7. Merge Request Açma

1. **Merge Requests → New merge request**.
2. Kaynak (source) ve hedef (target) branch seçin.
3. Açıklama ekleyin, reviewer atayın.
4. Gerekirse issue numarası ekleyerek ilişkilendirin.

Kod inceleme aşamasında "Discussion" sekmesini aktif kullanın.

## 8. Issue Yönetimi

* **Issues → New issue** adımından yeni görev veya hata açın.
* Etiketler (labels), milestone ve assignee atayın.
* Issue Board ile görevleri sürükle-bırak yöntemiyle yönetin.

**Örnek:** "Login sayfası tasarımı" isimli bir issue açıp frontend geliştiricisine atamak.

## 9. CI/CD Kullanımı

.gitlab-ci.yml dosyası ile otomasyon işlemleri tanımlayın.

stages:

- build

- test

- deploy

build\_job:

stage: build

script:

- npm install

- npm run build

test\_job:

stage: test

script:

- npm test

deploy\_job:

stage: deploy

script:

- echo "Deploy işlemi yapılıyor"

Pipeline’lar commit sonrası otomatik tetiklenir.

## 10. Hata Çözüm Önerileri

* **Push hatası**: git pull --rebase ile güncel kodu çekin.
* **Merge çatışması**: Çatışan dosyaları manuel düzenleyin, git add ve git commit ile kaydedin.
* **Pipeline başarısızlığı**: CI loglarını inceleyin, gerekli düzeltmeleri yapın.

## 11. İleri Seviye İpuçları

* Protected branch’ler ile ana branch’i koruyun.
* Merge request onay zorunluluğu koyun.
* Auto DevOps kullanarak test ve deploy süreçlerini hızlandırın.
* Code Quality ve Security Scan raporlarını CI/CD pipeline’ına ekleyin.

## 12. Güvenlik ve Uyumluluk

* 2FA (iki faktörlü kimlik doğrulama) aktif edin.
* Access Token’larınızı güvenli şekilde saklayın.
* Düzenli olarak repository izinlerini gözden geçirin.

## 13. Kaynaklar

* [GitLab Resmi Doküman](https://docs.gitlab.com)
* [GitLab CI/CD Pipeline Rehberi](https://docs.gitlab.com/ee/ci/)
* [Git Komutları](https://git-scm.com/docs)

Bu döküman, hem yeni başlayan hem de ileri düzey kullanıcılar için hazırlanmıştır ve sürekli güncellenecektir.