

Git Github

1. Ders
13.07.2023

B149 AWS & DevOps

B146 Cyber Security



Bugün ne yapıyoruz?

- ♦ Git nedir?
- ♦ Git Kurulumu
- ♦ Git Komutları
- ♦ Github Hesap Açılışı
- ♦ Kahoot



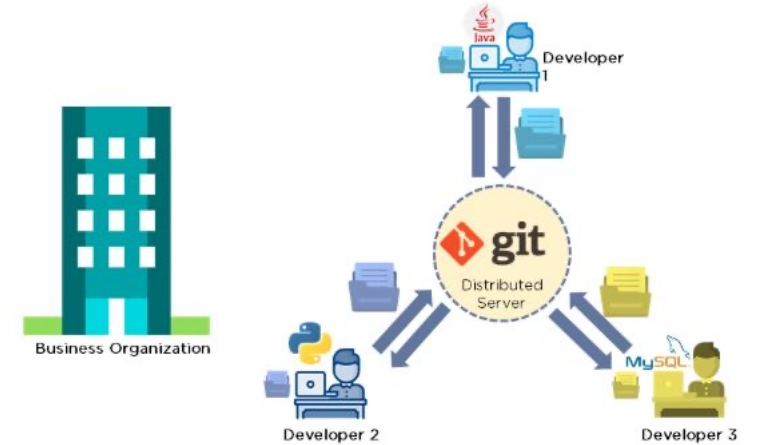
Git nedir?

- ♦ Git, değişiklikleri kaydeden bir "sürüm kontrol sistemi" dir.
- ♦ Git, dağıtık bir yapıya sahiptir. Bu, her kullanıcının kendi yerel kopyasının olduğu anlamına gelir. Bu şekilde, çevrimdışı çalışabilirsiniz ve değişiklikleri yerel olarak kaydedebilirsiniz.

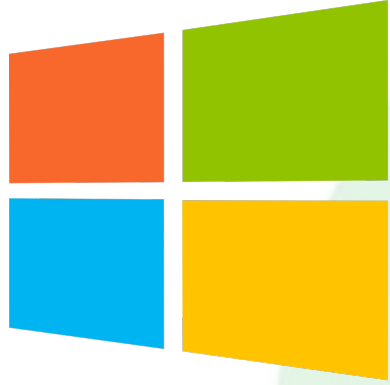


Git nedir?

- ♦ Git, komut satırı üzerinden çalışır.
 - ♦ Temel komutlar ve kısayollar öğrenildikten sonra, projelerinizin sürüm kontrolünü etkili bir şekilde git depolarınız(**repository** || **repo**) olarak yönetebilirsiniz.
- ♦ Git, proje dosyalarınızın farklı sürümlerini ve değişikliklerini dallar (**branch**) yönetebilirsiniz.
 - ♦ Her değişiklik bir "**commit**" olarak adlandırılır ve benzersiz bir kimlik (**hash**) ile tanımlanır.



Git Kurulumu



Kurulum Linki

<https://git-scm.com/>

Terminalde Kurulum Sonrası Kontrol

`git --version` | `git -v`

Git Config

Kullanıcı Adı ve E-Posta Adresi Yapılandırma

- `git config --global user.name "John Doe"`
- `git config --global user.email "john.doe@example.com"`

Kontrol etmek için komutları değersiz olarak yazınız.

Editör Ayarını Yapılandırma (isteği bağlı)

- `git config --global core.editor "nano"`

Renkli Çıktıları Etkinleştirme

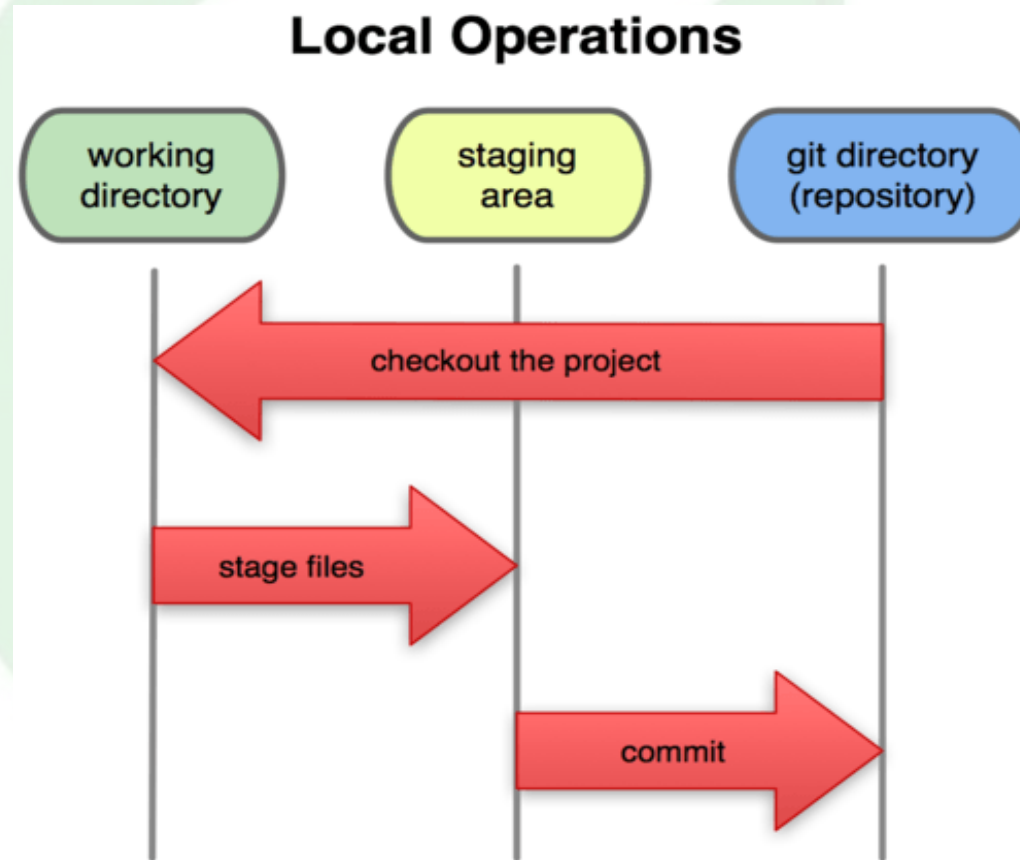
- `git config --global color.ui true`



Git Config

```
git config --global
```

Git Çalışma Yapısı



Git Proje Oluşturma

Dizini Git Deposu Olarak Başlatalım

- `git init`

Kontrol etmek için; terminale "`ls -la`" yazabilirsiniz.

Bir dosya oluşturup, bu dosyayı depoya ekleyelim

- `echo "Merhaba, Git!" > dosyam.txt`

Çalışma alanında olduğunuzu kontrol ediniz.

- `git status`
- `git add dosyam.txt`

İzleme alanında olduğunuzu kontrol ediniz.

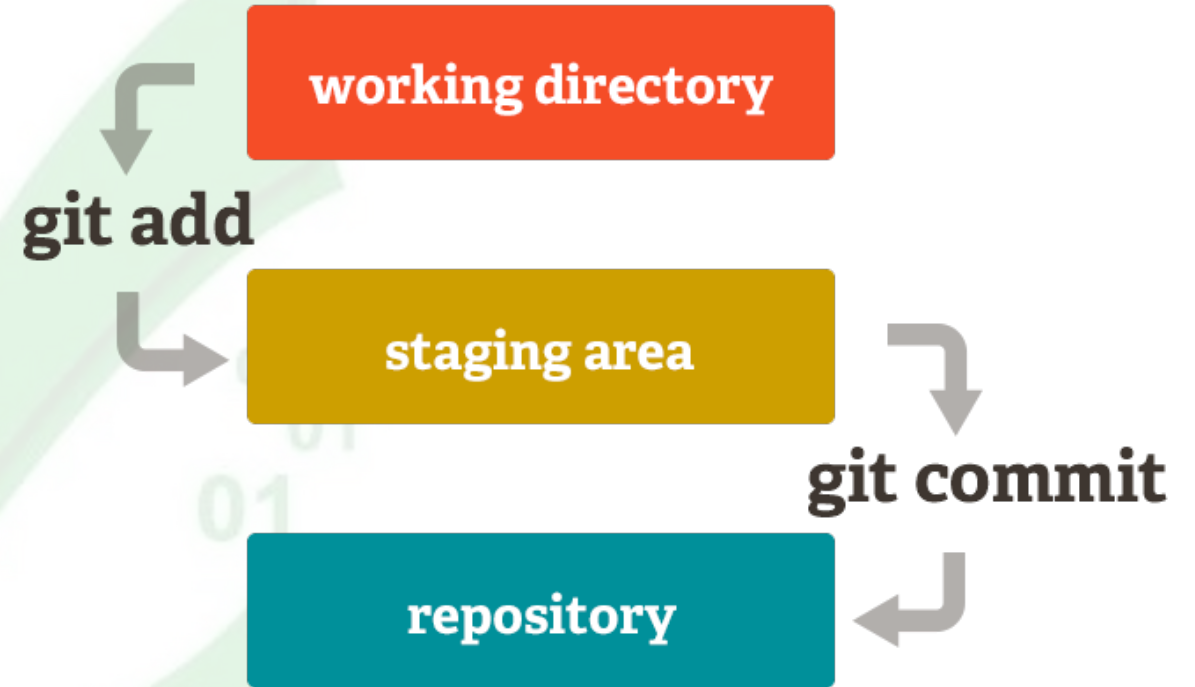
- `git status`

Dosyayı depoya ekledik, şimdi bir "commit" oluşturalım

- `git commit -m "ilkcommit"`

Oluşan Projemizi görüntüleyelim

- `git log`



Git Diff

Çalışma alanındaki tüm değişiklikleri görelim

- `git diff`

İzlemeye eklenmiş (staged) değişiklikleri görelim

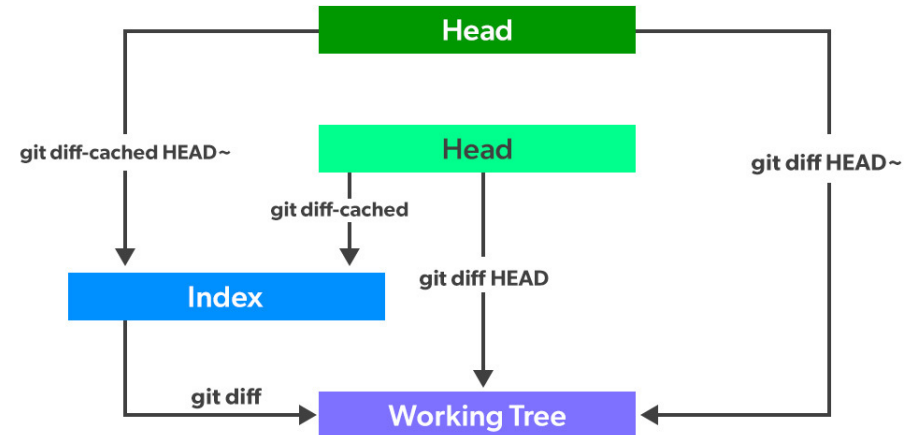
- `git diff --cached`

Belirli bir commit ile çalışma alanındaki değişiklikleri karşılaştıralım

- `git diff <commit_id>`

İki farklı commiti karşılaştıralım

- `git diff <branch_or_commit_1> <branch_or_commit_2>`



Git Checkout

Versiyonlar (Commit) Arası Geçiş Yapalım

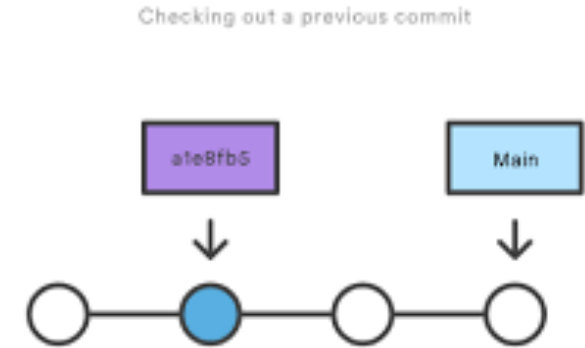
- `git checkout <commit_id>`

Alanlar Arası Geçiş Yapalım

- `git checkout -- <dosya_adi>`

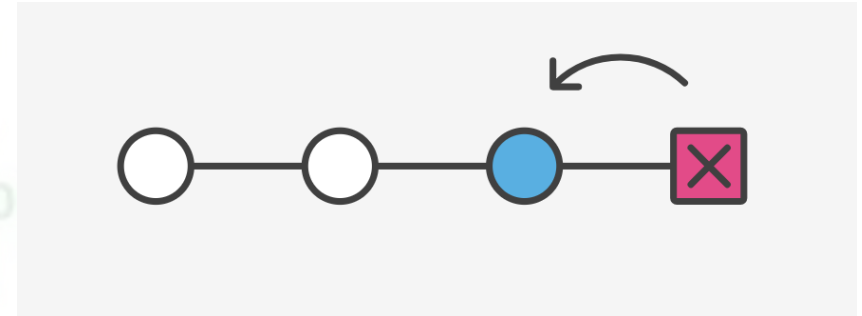
Not: git checkout komutunun 3 görevi vardır.

- Versiyonlar arası geçiş
- Alanlar arası geçiş
- Branch konusunda değinilecek



Git Reset

- ♦ Git reset komutu, geçmiş commitleri geri almak için kullanılır.
 - ♦ **Hard reset:** Geri alınan değişiklikler kalıcı olarak silinir, bu nedenle dikkatli kullanılmalıdır.
 - ♦ `git reset --hard <commit_id>`
 - ♦ **Soft reset:** Geri alınan değişiklikler geçici olarak bekleyen değişiklikler olarak işaretlenir.
 - ♦ `git reset --soft <commit_id>`
 - ♦ `git restore --staged <dosya-adi>`



Git Revert

- ♦ Git revert komutu, belirli bir commit' i geri alır, ancak geçmişi değiştirmez. Yeni bir geri alma commit' i oluşturarak geri alınan değişiklikleri iptal eder. Bu yöntem, geçmişin bozulmasını önler ve geri alınan değişikliklerin takip edilmesini sağlar.
 - ♦ `git revert <commit_id>`
- ♦ Belirli bir commit'i geri alır ve yeni bir geri alma commit'i oluşturur. İptal edilen değişiklikler yeni bir commit olarak geçmişe eklenir.

Git Branch

Yeni Bir Branch Oluşturalım

- `git branch < yeni-branch-adı>`

Mevcut Branch - > Yeni Branch' e Geçelim

- `git checkout < yeni-branch-adı>`

Branch' lerimizi Listeleyelim

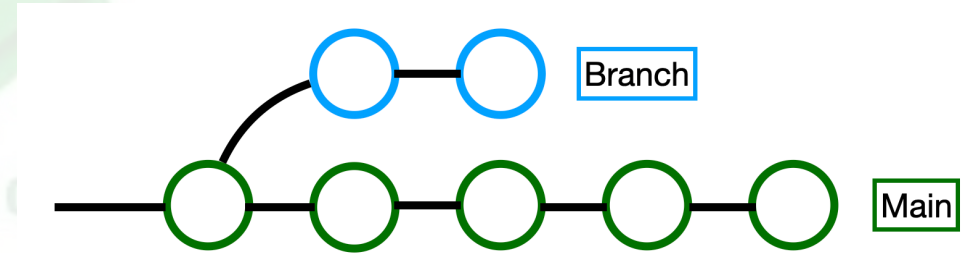
- `git branch`

Yeni Branch' imizi Silelim

- `git branch -d < yeni-branch-adı>`

Eğer Uzak Sunucunuza Bağlı Branch' leri Görmek İstiyorsanız

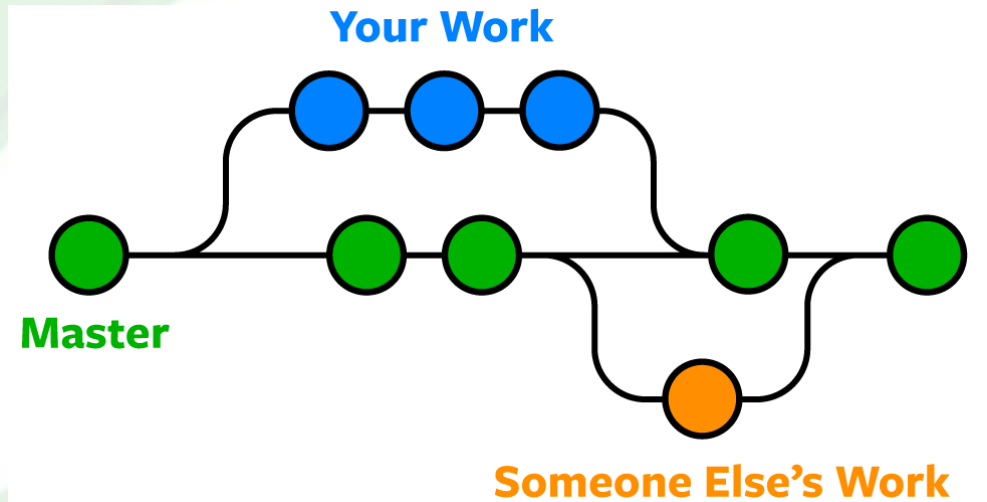
- `git branch -r`
- **Not :** Bu komuta Github konusunda tekrar bakacağız.



Git Merge

Yeni Branch ile Ana Branch' imizi Birleştirelim

- ♦ git checkout main
- ♦ git merge < yeni-branch-adı>



.gitignore Dosyası

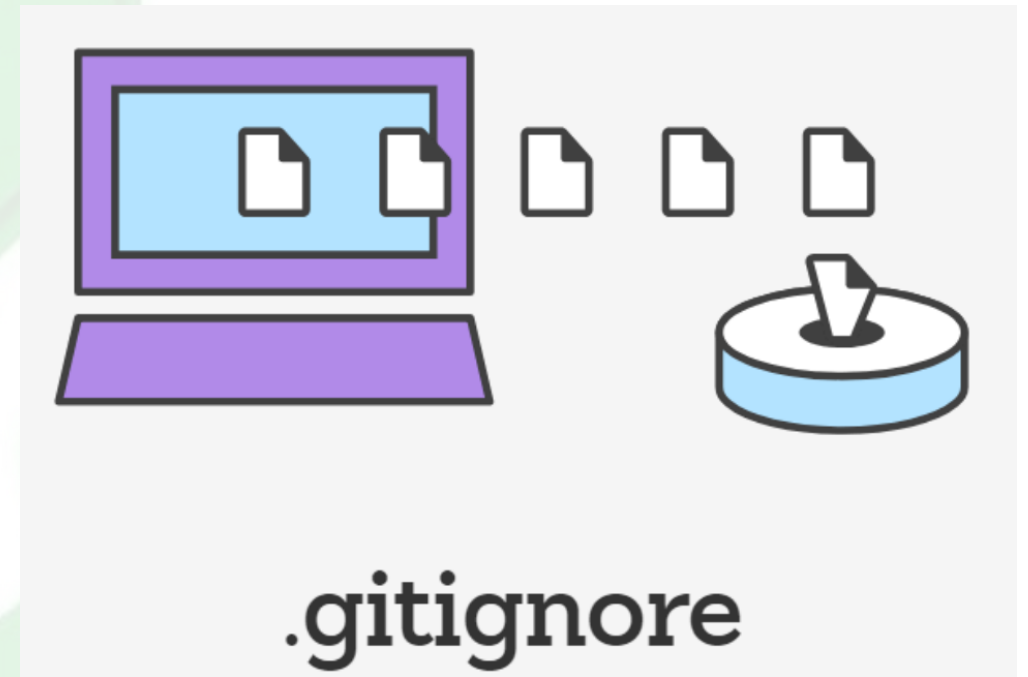
- Git tarafından takip edilmesini istemediğiniz durumlarda kullanılır.

.gitignore Dosyasını Oluşturalım

- `cat >> .gitignore`
 - `new.txt`
 - `doc/*`
 - `*.txt`
 - `**/*.txt`

Dosyayı Depoya Ekleyelim

- `git add .`
- `git commit -m "update"`



Git Github

2. Ders

14.07.2023

B149 AWS & DevOps

B146 Cyber Security



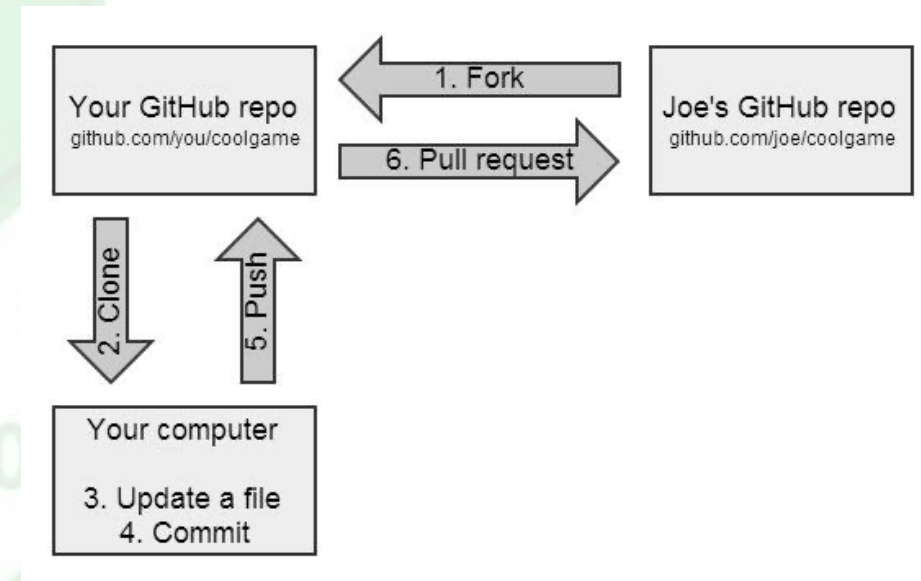
Bugün ne yapıyoruz?

- ♦ Github Özellikleri
- ♦ SSH Key & Token Oluşturma
- ♦ Github Proje Oluşturma
- ♦ Kahoot



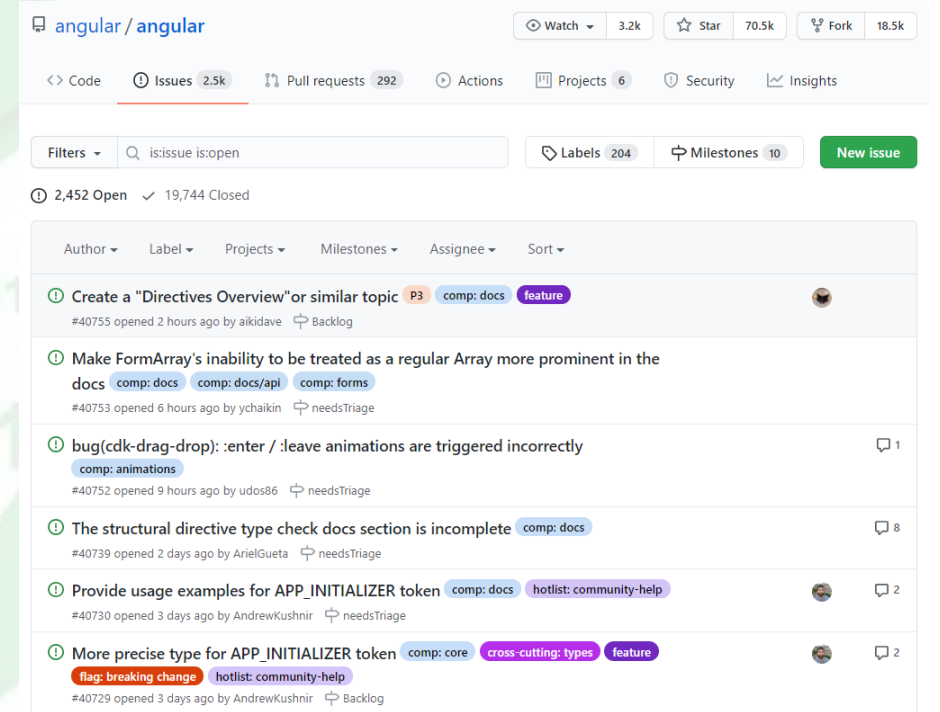
Github Watch | Star | Fork

- ♦ **Watch** : Kullanıcının belirli bir repo veya organizasyonu **takip etmek** için kullandığı bir özelliktir.
- ♦ **Star** : Kullanıcının beğendiği veya ilgi duyduğu bir repo'yu **işaretleme** ve **favorilerine eklemek** için kullandığı bir özelliktir.
- ♦ **Fork** : Başka bir kullanıcının repo' sunu kopyalayarak **kendi GitHub hesabına taşımak** ve bu kopya üzerinde bağımsız bir şekilde çalışmak için kullanılan bir işlemdir.



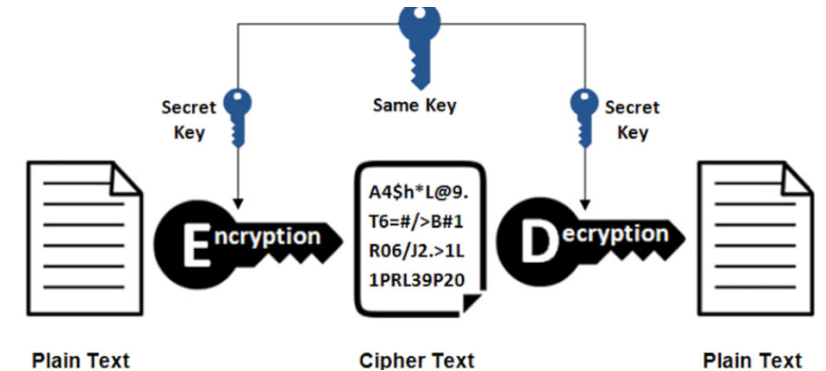
Github Issues

- Bir projede karşılaşılan sorunları, hataları veya önerileri takip etmek ve yönetmek için kullanılan bir özelliktir.
- Kullanıcılar, projenin GitHub sayfasında issues bölümünden yeni bir issue açabilir, mevcut issue' ları takip edebilir, yorumlar ekleyebilir ve issue' ları kapatılana kadar ilerleyişini izleyebilir.
- Bu, proje ekibi ve katkıda bulunanlar arasında iletişimi kolaylaştırarak projenin geliştirilmesine katkıda bulunur.



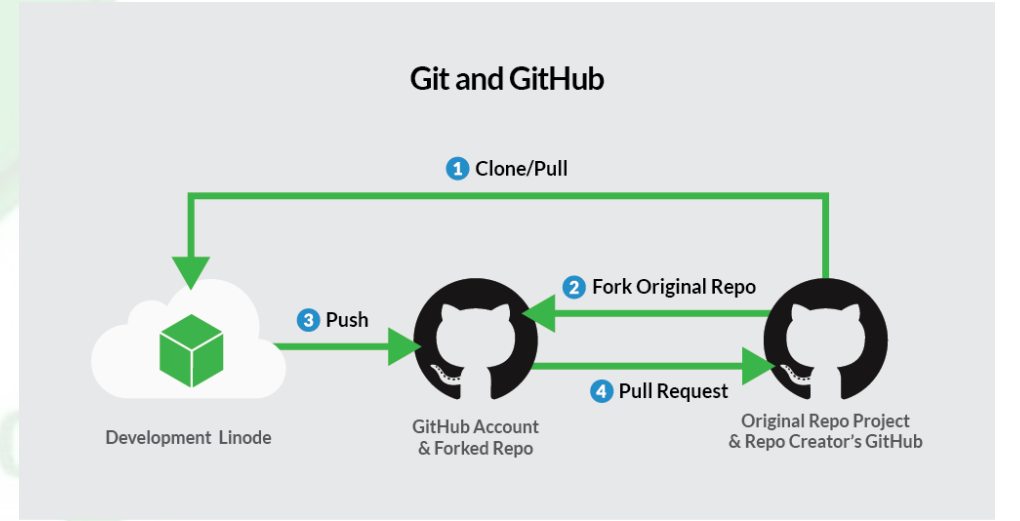
Github SSH Key & Token Oluşturma

- ♦ GitHub' da SSH anahtar çifti oluşturarak, SSH protokolünü kullanarak GitHub' a güvenli bir şekilde erişebilirsiniz.
 - ♦ **ssh-keygen**
- ♦ Örnek Token Kullanımı; komut satırında bir depoyu klonlamak için aşağıdaki komutu girersiniz git clone. Daha sonra kullanıcı adınızı ve şifrenizi girmeniz istenecektir. Parolanız istendiğinde, parola yerine kişisel erişim belirtecini girin.
 - ♦ **git clone <https://github.com/USERNAME/REPO.git>**
 - ♦ Username: **YOUR_USERNAME**
 - ♦ Password: **YOUR_PERSONAL_ACCESS_TOKEN**



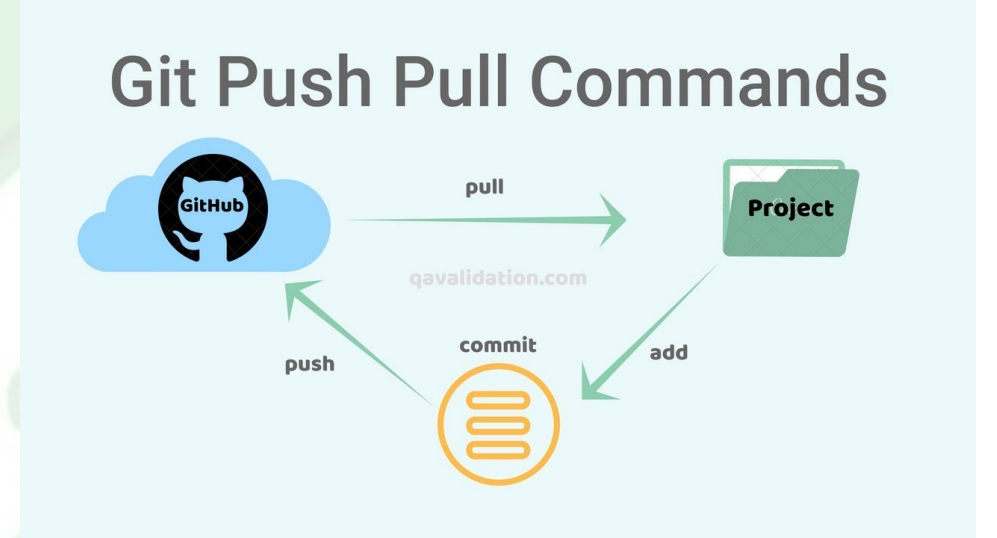
Github Clone

- GitHub' dan **git clone** komutunu kullanarak bir repo'yu kopyalamak, o repo' nun tam bir kopyasını lokal bilgisayarınıza indirmenizi sağlar.
 - GitHub' da tarayıcınızı açın ve repo' nun sayfasına gidin.
 - Sayfanın sağ üst köşesinde yeşil bir "Code" düğmesi göreceksiniz. Üzerine tıklayın.
 - Açılan menüden HTTPS veya SSH seçeneklerinden birini seçin. Genellikle başlangıç için HTTPS seçeneği daha uygun olabilir. URL'yi kopyalamak için "Copy" düğmesini tıklayın.
 - Terminali veya Git Bash' i açın ve gitmek istediğiniz klasörü açın.
 - **git clone <git_url>**



Github Pull

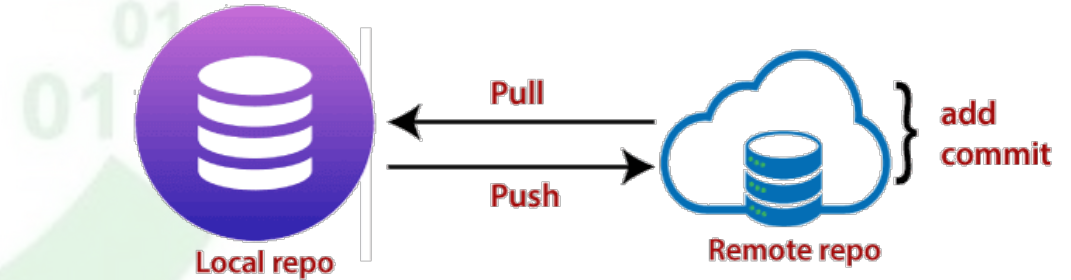
- GitHub' da **git pull** komutunu kullanarak, lokalde bulunan bir repo ile GitHub' daki uzak repo arasındaki deęişiklikleri senkronize edebilirsiniz. Bu işlem, GitHub' daki güncellemeleri lokaldeki çalışma kopyanıza entegre etmenizi sağlar.
 - **git pull**
- Bu komut, lokaldeki çalışma kopyanızı GitHub' daki uzak repo ile senkronize eder. Eğer uzak repo ile local repo arasında farklılıklar varsa, git pull komutu bu farklılıkları birleştirir veya günceller.



Github Push

- Lokaldeki değişiklikleri GitHub veya başka bir uzak repo üzerine yükleyebilirsiniz. Bu işlem, yerelde yaptığınız commit' leri uzak repo ile paylaşmanızı sağlar.

- **git push**

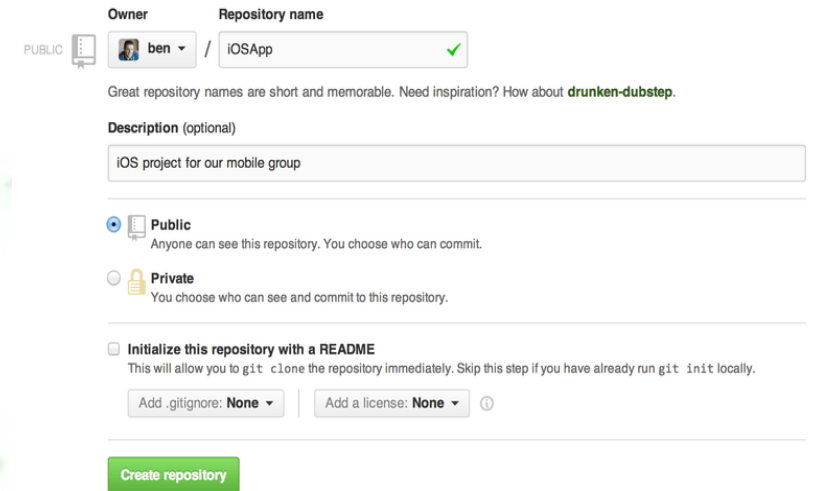


Github Proje Oluşturma



- ♦ Github web sitesinden hesabınızı oluşturduktan sonra ilk Repository oluşturarak local oturumunuz ile entegre edebilirsiniz.

Local ile Github Repo İlişkilendirme

- ♦ `echo "# a" >> README.md`
- ♦ `git init`
- ♦ `git add README.md`
- ♦ `git commit -m "first commit"`
- ♦ `git branch -M main`
- ♦ `git remote add origin https://github.com/<username>/<repo_name>.git`
- ♦ `git push -u origin main`



Owner **Repository name**

PUBLIC   / iOSApp ✓

Great repository names are short and memorable. Need inspiration? How about [drunken-dubstep](#).

Description (optional)

iOS project for our mobile group

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Git Github

3. Ders

15.07.2023

B149 AWS & DevOps

B146 Cyber Security

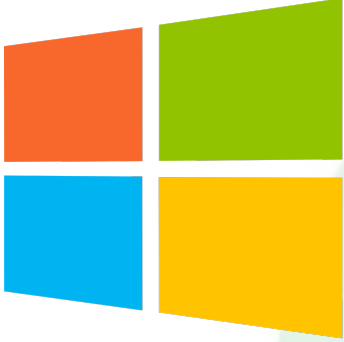


Bugün ne yapıyoruz?

- ♦ Github Desktop Kurulum
- ♦ Github Readme.md Oluşturma
- ♦ Tüm Ders Depoları Oluşturma
- ♦ Yararlı Linkler
- ♦ GitLab ve Bitbucket



Github Desktop Kurulumu



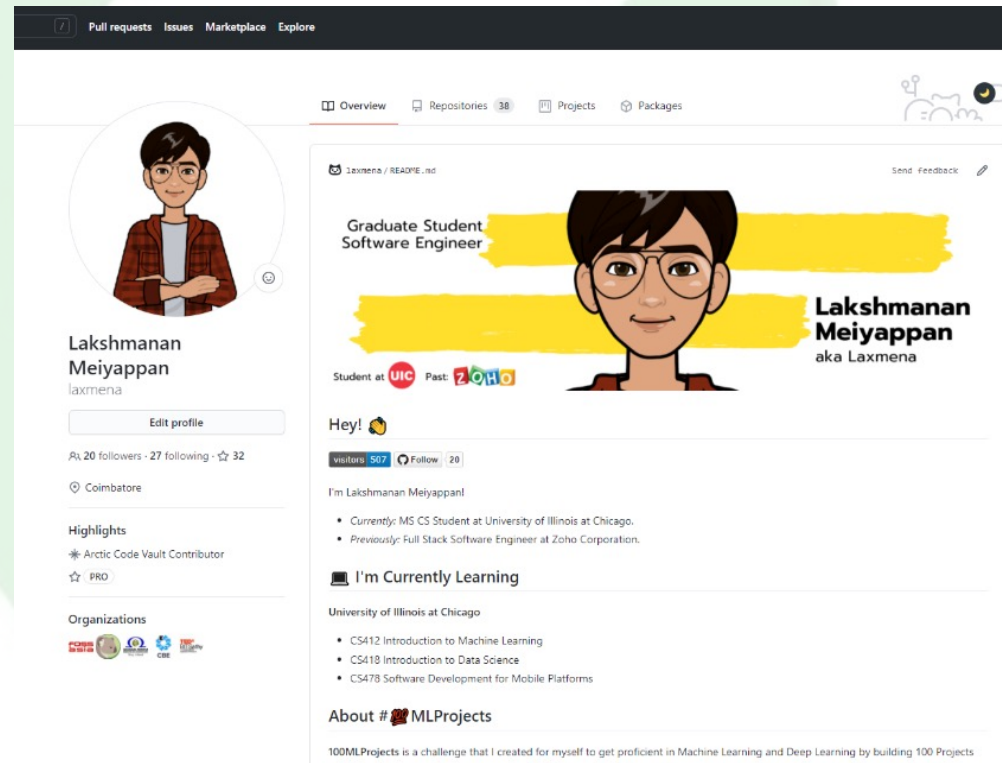
Kurulum Linki

<https://desktop.github.com/>



Github Readme.md Dosyası Oluşturma

Readme Generator



Github Yararlı Linkler

- ♦ [Github.com](#) -> [Github.dev](#)
- ♦ Arama çubuğuna
[awesome-cybersecurity](#) | [awesome-devops](#)

Other Version Control Systems



GitLab

ATLASSIAN



Bitbucket