



BATCH	:	146 - 149
LESSON	:	Git & Github
DATE	:	13.07.2023
SUBJECT	:	Git & Github - 1

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ



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



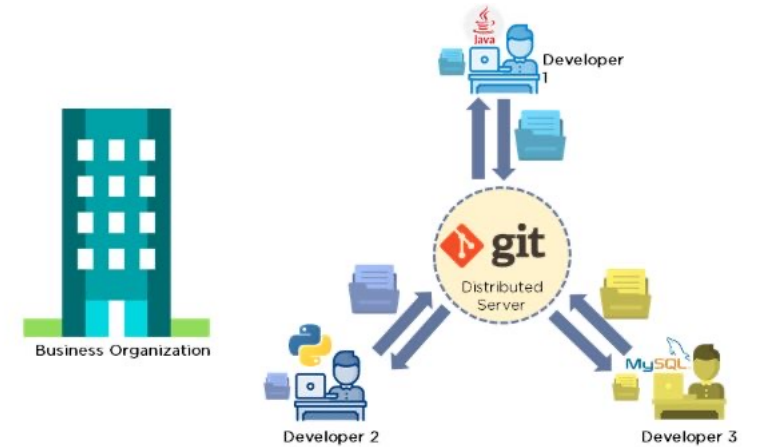
Git nedir?

- ♦ 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.

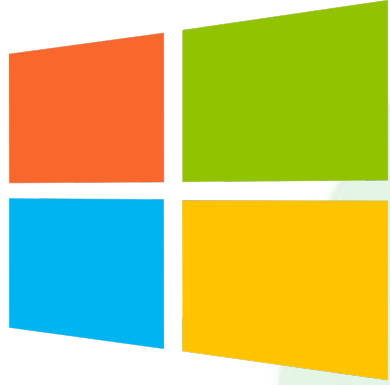


Git nedir?

- ♦ 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

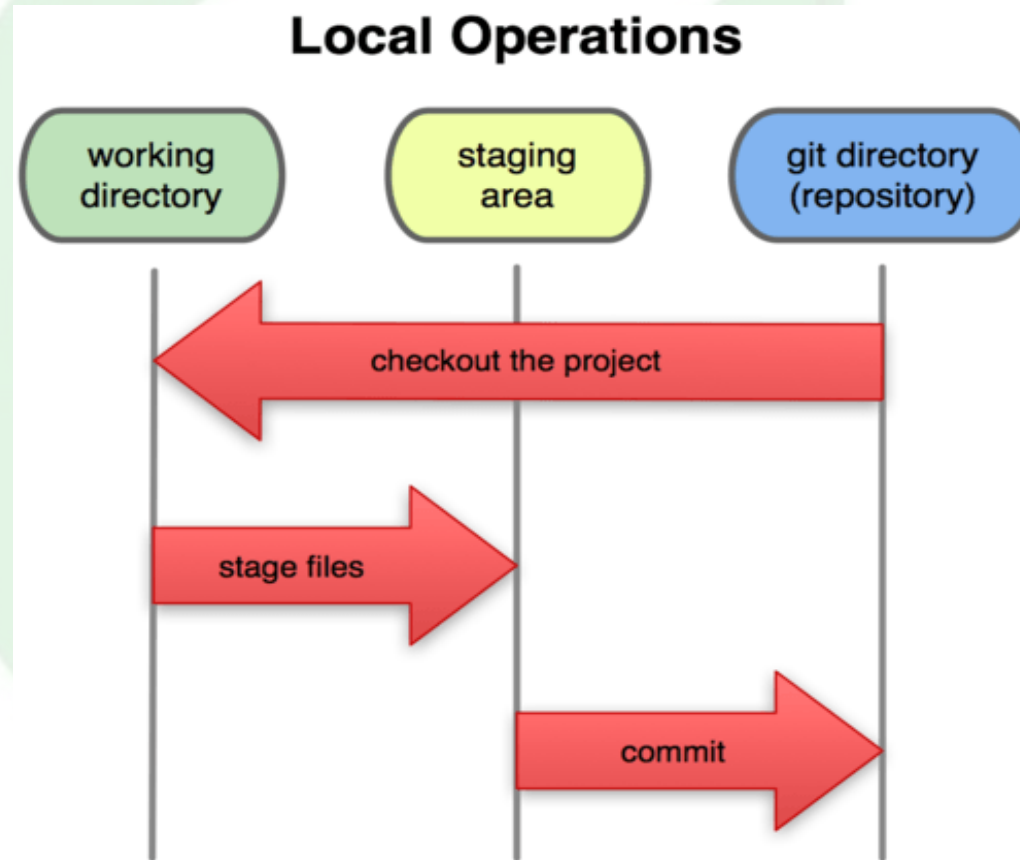
- `git config --global color.ui true`



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.

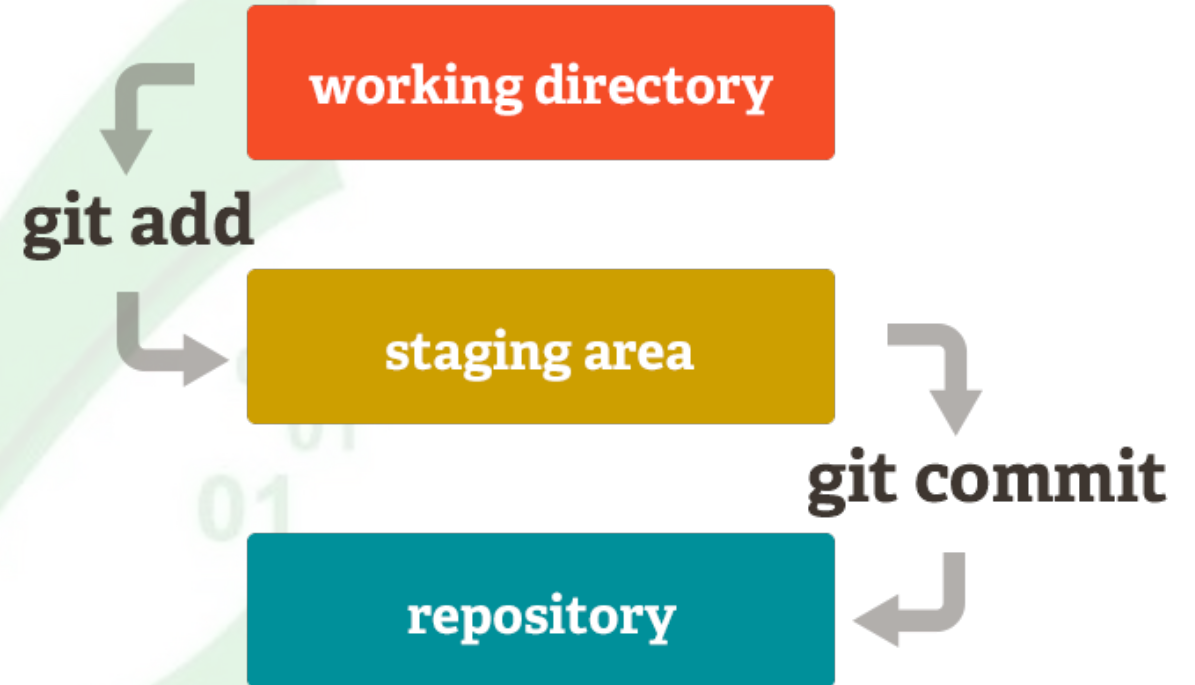
- `git status`

Dosyayı depoya ekledik, şimdi bir "commit" oluşturalım

- `git commit -m "ilkcommit"`

Oluşan Projemizi görüntüleyelim

- `git log`



Git Diff

Çalışma alanındaki tüm değişiklikleri görelim

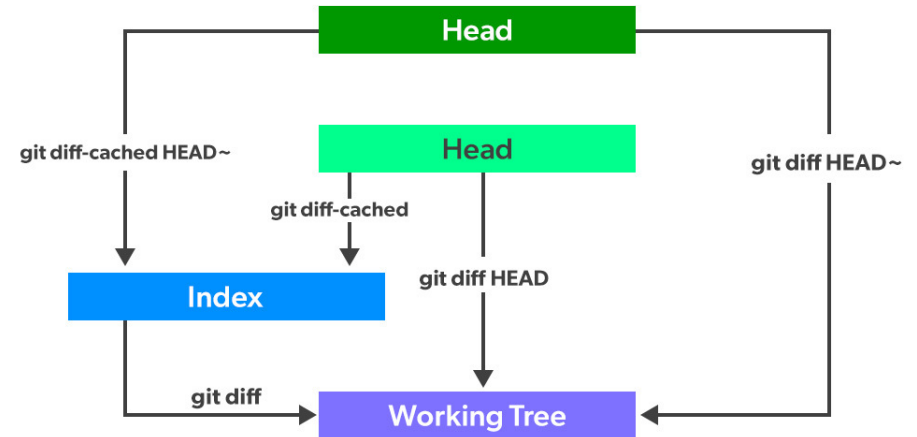
- `git diff`

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

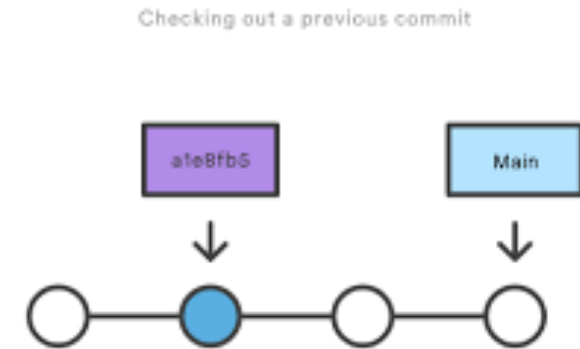
- `git checkout <commit_id> .`

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 Branch

Yeni Bir Branch Oluşturalım

- `git branch < yeni-branch-adı>`

Mevcut Branch - > Yeni Branch' e Geçelim

- `git checkout < yeni-branch-adı>`

Branch' lerimizi Listeleyelim

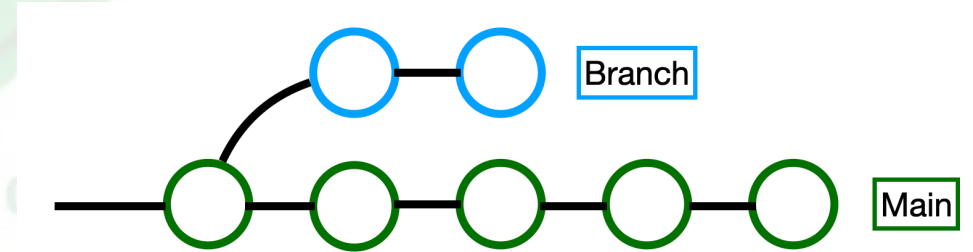
- `git branch`

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>

