**Git** je sistem kontrole verzija za praćenje promena u računarskim fajlovima i za koordinisan rad više ljudi na tim fajlovima. Primarno se koristiza verzionisanje koda pri razvoju softvera,ali se može koristiti za praćenje promena u bilo kojim fajlovima. – Razvio ga je Linus Torvalds, 2005. godine za potrebe razvoja Linuks jezgra – Git je otvoren softver

**Git konfiguracija**

Prvi put kada upalite git na svom računaru moraćete da uradite početnu konfiguraciju – Postoji nekoliko glavnih stvari koje morate da konfigurišete: • Vaše detalje(email,ime) • GIT editor • Podrazumevano ime grane(branch)

Sve ovo možemo promeniti korišćenjem git config komande

Git config – global.user za username

Git se uglavnom koristi preko komandne linije.

**Ls**

Ova komanda vam omogućava navodjenje sadržaja foldera

**Cd**

Omogucava vam da se krećete kroz sistem fajlova

**Pwd**

Pokazaće vam direktorijum u kom se trenutno nalazite

**Rm**

Omogućava brisanje fajlova i foldera

Kada se naprave promene one treba da se urežu.To se radi pomoću commit(git commit)

**Git log** Pomoću ove komande možete videti sve zabeležene promene(ko je upisao i kada)

**Git revert** Ako želimo da poništimo poslednje upisivanje i da upišemo nešto novo koristićemo:

**Git reset** ponistavanje poslednjeg urezivanja

**Git add** za promenu I dodavanje fajlova koje želimo

**Git commit** urezivanje novih promena

**Git push** Kada poništiš upisivanje,urežeš novo commit komandom ,onda je potrebno da to novo premestiš iz lokalnog projekta u lokaciju na GitHub-u

**Git pull** Pull je suprotna komanda od push. – Pomoću ove komande možete prebaciti sve sa lokacije na GitHub na vaš lokalan projekat

**Branches**

Grane su pokazivači na upisivanja. – Korišćenjem više grana bi mogli da odvojimo funkcije ,i da ih razvijamo na posebnoj grani ,a onda na kraju bi spojili grane u glavnu.

**Merge**

Kada završite sa promenama,onda možete spojiti grane sa glavnom granom i postaviti to na vebsajt

**Ssh**

Da bi mogli da prebacujete promene sa vašeg racunara na lokaciju i obrnuto možete koristiti: – HTTPS-što zahteva vaše korisničko ime i lozinku svaki put kada pokušate da unesete svoje promene – SSH ključeve-možete da generišete svoje ssh ključeve i dodate svoj javni ključ na GitHub-u.Na taj način se ne bi tražilo korisničko ime i lozinka svaki put kada unosite promene

Za pravljenje novog para ključeva koristi se: ssh-keygen

Ovim se generišu dva fajla: • 1 privatan ključ i 1 javni ključ • 1 javni ključ koji otpremiš(upload) na GitHub