

Git podržani protokoli i priprema servera

23. siječnja 2018.

Protokoli

Git podřava řetiri protokola

- ▶ Lokalni protokol
- ▶ HTTP protokol
- ▶ SSH(secure shell) protol
- ▶ Git protokol

SSH protokol

- ▶ Standardni protokol za prijenos podataka u slučaju da korisnik konfigurira Git server
- ▶ SSH pristup serverima je obično namješten, a ako nije, vrlo ga je jednostavan konfigurirati
- ▶ Način korištenja: `git clone ssh://[user@]server/project.git`
- ▶ Velik broj ljudi koji koriste Git su upoznati sa SSH protokolom
- ▶ Siguran i kriptiran pristup i transfer podataka
- ▶ Nedostatak je što za pristup podacima korisnika SSH protokola ostali korisnici moraju imati poseban SSH pristup

Lokalni protokol

Koristi se kada je repozitorij podignut na računalu na koji svi članov tima imaju lokalni pristup. Članovi imaju pristup repozitoriju ako su ulogirani na istome računalu ili ako koriste disk koji je *share-an* svim članovima tima.

Naredba za kloniranje lokalnog repozitorija:

```
$ git clone /srv/git/project.git
```

Lokalni protokol

Tablica: Prednosti i nedostaci

Prednosti	Nedostaci
<p>Jednostavno podizanje repozitorija.</p> <p>Lagano je dohvatiti repozitorij od kolege:</p> <pre>git pull /home/john/project</pre>	<p>Teško je pristupiti repozitoriju sa udaljenih lokacija.</p> <p>Konfiguriranje NFS-a(network file system) nije jednostavno</p> <p>Svaki korisnik ima potpunu ovlast u mijenjanju git-ovih datoteka.</p>

HTTP protokol

Do verzije gita 1.6.6. HTTP protokol je bio samo read-only. Odnosno, podaci su se mogli samo preuzeti preko http protokola. Nakon 1.6.6. verzije *pametni HTTP* protkol je predstavljen

Pametni HTTP protokol (eng.Smart HTTP)

Radi slično kao i git i ssh protokol te koristi i kriptiranu verziju HTTP-a. Podosta je bolji od SSH(secured shell protocol), jer se mogu koristiti lozinke i korisnička imena koja su pogodnija za korisnika više nego SSH ključevi.

Smart HTTP je postao najpopularniji način korištenja gita. Za provjeru podataka i enkripciju koristi jedan te isti link (npr.git:// protcol). Pri pushanju repozitorija ili čitanju, pitati će nas lozinku i korisničko ime.

Glupi HTTP protokol (eng.Dumb HTTP)

Prepoznamo ga po tome što server ne reagira na Smart protokol.
Ljepota Dumb protokola je u tome što se lako postavlja.

HTTP protokol

Tablica: Prednosti i nedostaci pametne verzije

Prednosti	Nedostaci
Jedan jedini URL za sve načine povezivanja Korištenje lozinke i korisničkog imena, a ne SSH ključ Možemo postaviti repository-e kao read-only preko HTTPS-a.	Spajanje na neke servere zna se zakomplicirati Ako se koristi preko HTTP-a kompliciranije je push-at nego preko SSH

Git protokol

- ▶ Pruža uslugu sličnu SSH protokolu.
- ▶ Nema dodatnu provjeru autentičnosti.
- ▶ Repozitorij će biti dostupan svima. Zbog toga što je repozitorij javan, nitko ne može pushati na njega.

Git protokol

Tablica: Prednosti i nedostaci git protokola

Prednosti	Nedostaci
Najbrži protokol Pogodan je za hostanje velikih projekata	Nedostatak provjere autentičnosti Teško se konfigurira

Konfiguracija servera

1. Bare repozitorij

Za početak treba inicijalizirati *bare* repozitorij, sa komandom `git init --bare`.

Bare repo. ne sprema slike(snapshotove) trenutnog napredka na nekome projektu, nego sprema povijest projekta.

Takvi repozitoriji su npr. na githubu. Završavaju sa `.git` ekstenzijom.

Onda se izvrši kloniranje `git clone /direktorij`.

2. Uploadanje na server

Nakon što smo napravili bare repozitrij, isti moramo podignuti na server.

Kopiranje na server se izvršava preko SSH protokola komandom `scp -r bare-repo.git korisnik@domena.com:/direktorij`

Svatko tko ima SSH pristup tom serveru može klonirati repo komandom `git clone korisnik2@domena.com:/direktorij/bare-repo.git`

Oni korisnici koji imaju pristup SSH pisanju na server mogu i pushati promjene na repo na serveru.

Opisani koraci su dovoljni kako bi git repo bio postavljen na serveru i kako bi ga kolaboratori mogli koristiti.

SSH pristup

Ako želimo dodatnu kontrolu nad pristupom repozitorija, možemo ju postaviti sa dozvolama pristupa od operativnog sustava.

Kako bi svim članaovima tima dopustili pristup serveru, moramo na serveru dodati svakoga korisnika, što može bit naporno kod većih timova.

U tom slučaju možemo napraviti samo jednog korisnika na serveru i prikupiti sve SSH javne ključeve, od članova tima, koje spremamo u text fajlu sa javnim kljucevima na serveru

Sada svatko može pristupati serveru preko komande `ssh git@server`.

Ova metoda nikako ne mijenja način funkcioniranja gita, odnosno svi developeri mogu koristiti repozitorij normalno kao i sa više korisnika na serveru.

- ▶ <https://git-scm.com/book/en/v2/Git-on-the-Server-The-Protocols>
- ▶ https://en.wikipedia.org/wiki/Secure_Shell
- ▶ <https://git-scm.com/book/en/v2/Git-on-the-Server-Getting-Git-on-a-Server>
- ▶ <http://www.saintsjd.com/2011/01/what-is-a-bare-git-repository/>