

Podržani protokoli za Git i priprema servera

M. Čalogović, F. Kočić, D. Mirošević

Sveučilište u Rijeci - Tehnički fakultet

27. siječnja 2019.



Protokoli

O protokolima za Git

Lokalni protokol

HTTP protokoli

Smart HTTP

Dumb HTTP

SSH protokol

Git protokol

Git na serveru

Postavljanje Git-a na server

SSH postava servera

Git Daemon

Smart HTTP

Postava Smart HTTP-a

GitLab

Literatura

Protokoli za Git

Četiri vrste:

- ▶ Lokalni protokol
- ▶ HTTP protokoli:
 - ▶ Smart HTTP
 - ▶ Dumb HTTP
- ▶ SSH protokol
- ▶ Git protokol

Lokalni protokol

- ▶ Najjednostavniji protokol korišten na istom uređaju
- ▶ Koristan ako svi članovi imaju pristup zajedničkom disku ili (manje vjerojatno) koriste isto računalo
- ▶ Kloniranje repozitorija na lokalnom uređaju se jednostavno napravi naredbom:

```
$ git clone /put/do/repositorija/projekt.git
```

ili

```
$ git clone file:///put/do/repositorija/projekt.git
```

Lokalni protokol

- ▶ Također se lokalni repozitorij može dodati postojećem projektu:
\$ git remote add lokalni_projekt
/put/do/repozitorija/projekt.git
- ▶ Ovako se mogu primati i slati datoteke na repozitorij putem imena "lokalni_projekt" kao da ga koristimo preko mreže

HTTP protokoli

- ▶ Jednostavni za korisnika
- ▶ Funkcionira slično SSH i Git protokolima
- ▶ Dvije vrste:
 - ▶ Smart HTTP (noviji)
 - ▶ Dumb HTTP (stariji)
- ▶ Naredba za kloniranje:
\$ git clone https://primjer.hr/projekt.git

Smart HTTP

- ▶ Koristi HTTPS portove i razne HTTP mehanizme za autentifikaciju
- ▶ Lakše za koristiti od SSH protokola, koristi korisničko ime i lozinku za prijavu umjesto SSH ključeva
- ▶ Zbog Smart HTTP protokola, URL koji koristimo za gledati repozitorij preko web preglednika je isti kojim možemo klonirati repozitorij

Dumb HTTP

- ▶ Stariji od Smart HTTP-a, danas se koristi kao pričuvena metoda spajanja na repozitorij, ako Smart HTTP ne uspije
- ▶ Izuzetno jednostavan za koristiti
- ▶ Svatko tko ima pristup web serveru na kojem se nalazi repozitorij, može taj isti repozitorij klonirati
- ▶ Ovaj protokol je najbolje koristiti samo za read-only verziju repozitorija

SSH protokol

- ▶ Jako česta metoda za spajanje na repozitorij
- ▶ Jednostavan za postaviti i koristiti
- ▶ Koristi ključeve za autentifikaciju, no zbog toga se SSH ne može koristiti za anonimne, open-source projekte
- ▶ Naredba za kloniranje:
\$ git clone ssh://[korisnik@]server/projekt.git
ili
\$ git clone [korisnik@]server:projekt.git

Git protokol

- ▶ Poseban protokol koji dolazi sa samim Gitom
- ▶ Sluša na svom zasebnom portu (9418)
- ▶ Funkcionira slično SSH protokolu, no ne traži nikakvu autentifikaciju
- ▶ Bilo tko može pristupiti i klonirati Git repozitorij
- ▶ Da bi se repozitorij mogao posluživati preko Git protokola, repozitorij treba imati datoteku *git-daemon-export-ok* u glavnom direktoriju



Postavljanje Git-a na server

- ▶ Koraci:
 - ▶ izvoženje postojećeg repozitorija u novi prazan repozitorij u kojemu nema radnog direktorija
 - ▶ Primjer: `$ git clone --bare moj-projekt moj-projekt.git`
 - ▶ stavljanje praznog repozitorija u server
 - ▶ Primjer: `$ scp -r moj-projekt.git korisnik@git.primjer.com:/srv/git`
 - ▶ Svi korisnici koji imaju pristup serveru mogu klonirati naš repozitorij

SSH Public Key

- ▶ SSH Public Key - način autentifikacije mnogih Git servera
- ▶ Svaki korisnik koji radi na nekom projektu na serveru mora imati svoj 'javni ključ'
- ▶ Korisnici mogu naći svoj 'javni ključ' u svojem `/.ssh` direktoriju
- ▶ Ako nemaju 'javni ključ' moraju ga generirati u Git-u izvođenjem "ssh.keygen" programa

SSH postava servera

- ▶ Korisnik treba imati git korisnički račun i .ssh direktorij
- ▶ Korisnik prebacuje javne ključeve u ovlaštene ključeve za njega
- ▶ Može se napraviti prazan repoitorij sa naredbom git init i opcijom –bare koji inicijalizira repozitorij bez radnog direktorija
- ▶ Ovom se metodom može brzo napraviti read/write Git server za nekoliko developera

Git Daemon

- ▶ Česti izbor za brz i neautoriziran pristup Git podacima
- ▶ Sve što se radi ovim protokolom je javno
- ▶ Veoma jednostavan za namjestiti ovom naredbom
\$ git daemon -reuseaddr -base-path=/srv/git/ /srv/git/

Smart HTTP

- ▶ SSH protokol nam omogućuje autentificirani pristup, dok nam Git protokol daje slobodan pristup
- ▶ HTTP protokolom možemo oboje istovremeno
- ▶ Pomoću Smart HTTP-a možemo postaviti server tako da provjerava može li klijent komunicirati preko HTTP-a, odnosno ako ne može, da komunicira s njim na starije metode
- ▶ Također će nam ovo omogućiti pristup repozitoriju preko web preglednika

Postava Smart HTTP-a

- ▶ Za postavu HTTP protokola, koristimo Apache za CGI server (više o postavi na poveznici)
- ▶ Uz Apache možemo izabrati hoće li provjeravati jesu li korisnici autentificirani te može li im posluživati repozitorij
- ▶ Autentificirani korisnici imaju read/write pristup, dok neautentificirani imaju read-only pristupS
- ▶ Autentificiranim korisnicima će lozinke biti zapisane u *.htpasswd* datoteci za provjeru
- ▶ Također je bitno postaviti SSL enkripciju radi sigurnosti

GitLab server

- ▶ GitLab nudi postavu modernijeg servera s više pogodnosti
- ▶ Prednost je što dolazi s preglednim sustavom za održavanje repozitorija kojem se pristupa preko web preglednika
- ▶ Postavljanje GitLab servera može biti kompliciranije, pošto koristi i baze podataka
- ▶ Više o postavi na poveznici

Literatura



S. S. B. Straub, *Pro Git (2nd edition)*.
2014.