



Git Osnove - Online Radionica 2021

Autor: Aleksandar Pajić

icbtech.io



Aleksandar Pajić

Web Developer & UI/UX Designer

Email: aleksandar.pajic@icbtech.rs



Version Control System

Šta je to Version Control System?

- “Version control” je sistem koji prati promene koje činimo nad jednim ili više fajlova kroz određeni vremenski period, tako da bi imali mogućnost da kasnije izaberemo specifične verzije našeg sistema.
- U “version control” može da se postavi mnogi različiti tipovi fajlova.



**Ali zašto nam je potreban
Version Control?**

Version Control System

Zašto Version Control?

- A Version Control System (VCS) vam dozvoljava da vratite fajlove u njihovo prethodno stanje. Ne samo fajlove, već i čitave projekte, kao i da nadgledate sve promene koje ste vi ili vaše kolege načinili u izabranom periodu.

Zašto bi vraćali fajl u prethodno stanje?

- Uz praćenje koda i njegovo vraćanje na prethodno stanje možemo lakše da vodimo evidenciju ko je kad i šta modifikovao. Samim tim možemo lakše otkloniti problem (bug) ukoliko on postoji.
- Ukoliko se desi da projekat dođe u sledeće stanje:  uvek imamo opciju da povratimo sve na prvobitno stanje sa kojim smo počeli da radimo



Šta je to Git?

Više o Git-u

- Git je version-control sistem koji služi za praćenje promena unutar vaših fajlova, između jedne ili više osoba. Git je rasprostranjen Version Control sistem. Šta to znači? Znači da Git ne zavisi od centralnog servera kako bi skladištio verzije fajlova unutar projekata.
- Umesto toga, svaki korisnik klonira kopiju repozitorijuma (kolekcija fajlova) kako bi imao celu istoriju projekta na njegovom hard disku.

Kako je nastao Git?

Istorija Git-a

- Git je kreiran od strane Linux Torvalds-a, 2005 godine za razvijanje Linux kernels. Velike zasluge za doprinos Git-u idu kernel developerima koji razvijaju Git još od samog nastanka.
- Torvalds je želeo raspoloživ sistem koji bi koristio kao bolju i besplatnu alternativu BitKeeper-u (source-control management (SCM) sistem koji su koristili za održavanje projekata početkom 2002. godine).

Razvijanje Git-a počelo je 2005. godine od strane Linusa, koji je kasnije zadužio Junia Hamanu da aktivno vodi održavanje projekta. Junio Hamano je takođe jedan od programera koji je pružio mnogo Git projektu.

Primer

```
list1 = [1,2,3]  
list1.append(4)  
print(list1)
```

#initialFile

```
list1 = [1,2,3]  
list1.append(4)  
list1.append(5)  
print(list1)
```

#addedALine

```
list1 = [1,2,3]  
list1.pop()  
print(list1)
```

#makingChanges

Prednosti Git-a

Koje su prednosti korišćenja Git-a?

- Git vam takođe pomaže da sinhronizujete vaš kod sa drugom osobom, ili čak sa celim timom.
- Zamislite da radite sa vašim kolegom na istom projektu, i oboje radite na istim fajlovima unutar projekta.
- Git uzima vaše promene i promene vašeg kolege zasebno i spaja ih u jedan “Master” repozitorijum.
- Dakle koristeći Git možete biti sigurno da svi radite na istoj verziji repozitorijuma, što isključuje brigu o tome da li ste poslali fajlove između sebe putem mejla ili USB stick-a.
- Samim tim kolaboracija, tj. zajednički rad postaje vrlo lagan i zanimljiv posao.

Repozitorijum

Šta je repozitorijum (eng. “repository”)?

- Repozitorijum je ništa više sem kolekcije izvornog (eng. “source”) koda.

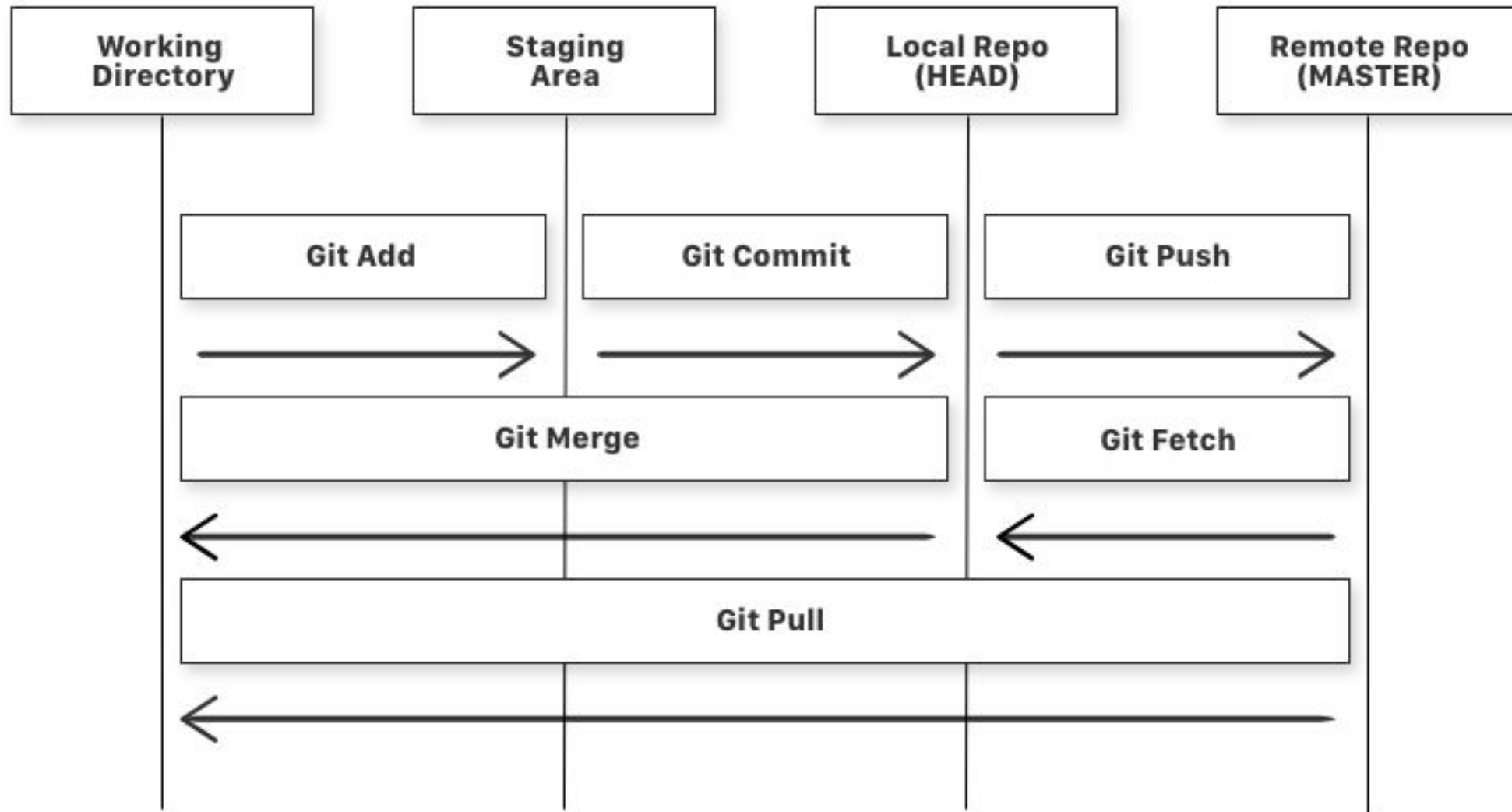
Funkcionalnost Git-a

Git workflow se sastoji od četiri osnovna elementa.

- Radni direktorijum, “Zona” pripreme (“Staging Area”), Lokalni repozitorijum, Remote repozitorijum

Unutar radnog direktorijuma, postoje tri moguća stanja

- Promene mogu biti pripremljene. Što znači da su izmenjeni fajlovi označeni za commit u lokalni repozitorijum, ali i dalje nisu commit-ovani.
- Mogu biti modifikovani, što znači da fajlovi sa izmenama još uvek nisu sačuvani u lokalni repozitorijum
- I takođe mogu biti commit-ovani, tj. pripremljeni, tačnije sve izmene koje ste napravili su bezbedno sačuvane u vaš lokalni repozitorijum..



Osnovne komande

Osnovne komande koje se svakodnevno koriste tokom rada na projektima.

1. `git add` je komanda koja se koristi da dodamo fajl koji je u radnom direktorijumu u zonu pripreme.
2. `git commit` je komanda koja se koristi kako bi se svi pripremljeni fajlovi dodali lokalnom repozitorijumu.
3. `git push` je jedna od najbitnijih komandi, koristi se kako bi sve commit-ovane fajlove unutar lokalnog repozitorijuma poslali u remote repozitorijum, kako bi isti izmenjeni fajlovi bili dostupni svima koji imaju pristup projektu.
4. `git fetch` se koristi kako bi preuzeli fajlove iz remote repozitorijuma u lokalni, ali ne i u radni direktorijum.
5. `git merge` komanda služi kako bi fajlove iz lokalnog repozitorijuma preuzeli u radni repozitorijum.
6. `git pull` is služi kako bi se svi fajlovi preuzeli iz remote repozitorijuma direktno u radni repo, komanda je slična `git fetch` i `git merge` komandama.

Zlatno pravilo

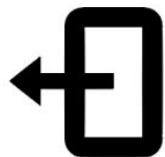
In case of fire



1. `git commit`



2. `git push`



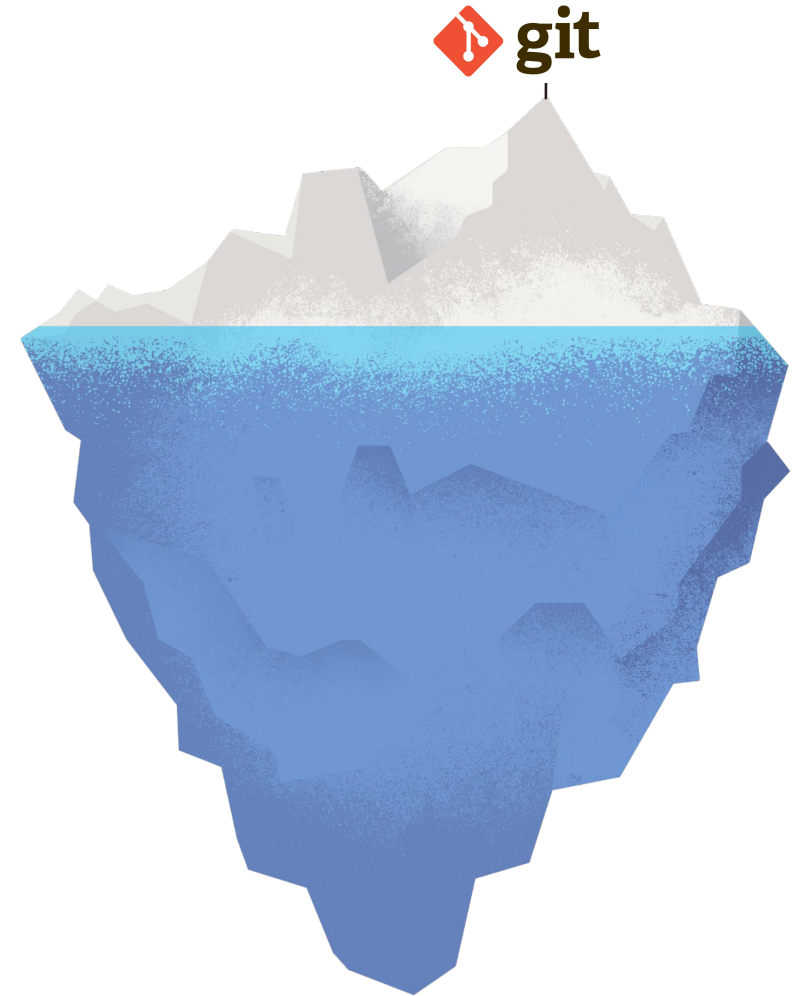
3. leave building

Git cheatsheet

Lista ostalih Git komandi:

Git cheatsheet:

- <https://education.github.com/git-cheat-sheet-education.pdf>
- <http://git-cheatsheet.com/>



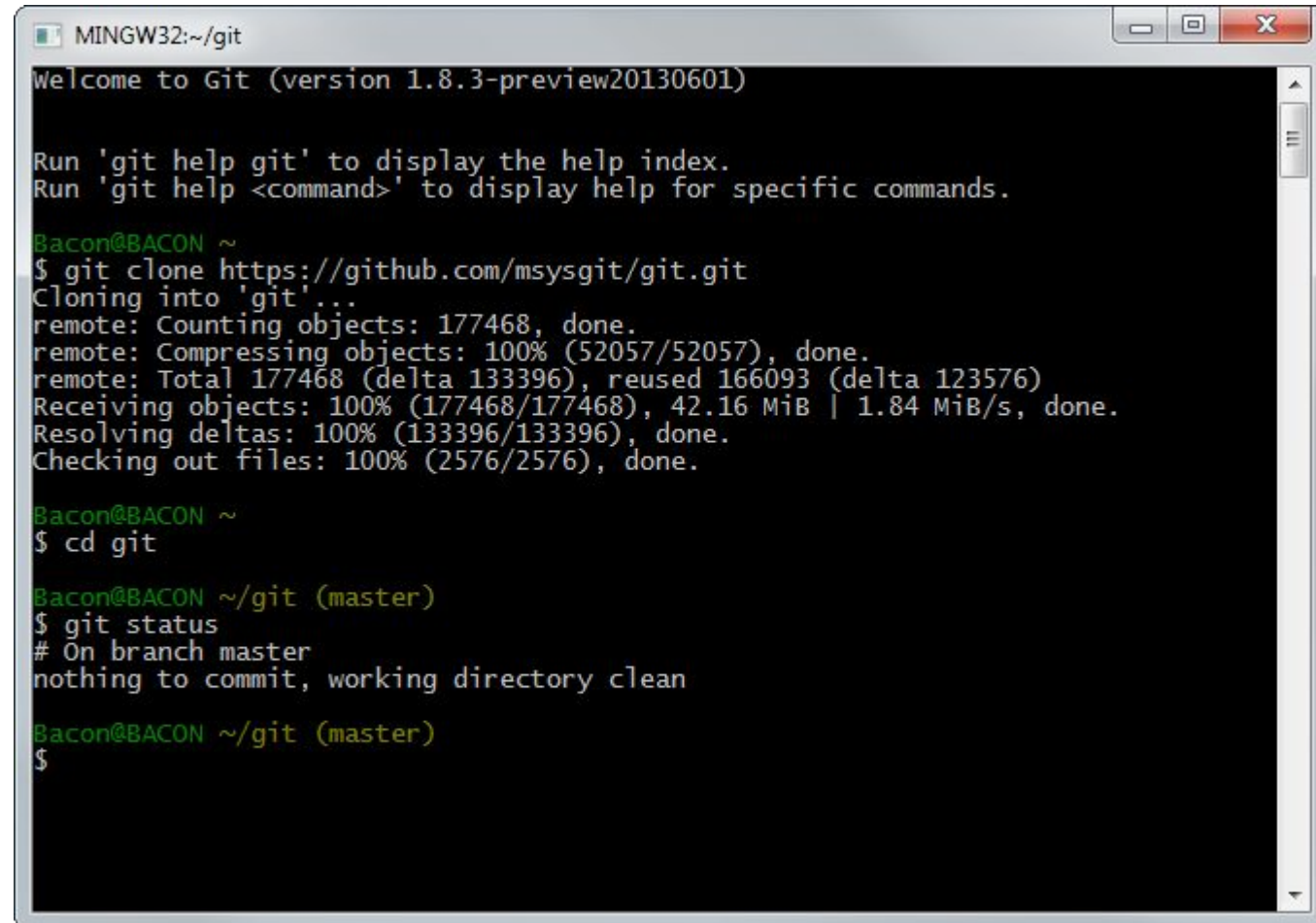
Git alatke

Postoje dva različita načina na koji programeri koriste Git:

Git CLI: CLI predstavlja Command Line Interface. U prevedenom značenju, otvorite terminal, ukucate komandu i sa tim kažete Git-u šta treba da radi. Ovo je podrazumevani interface koji dobijete nakon instalacije Git-a.

Git GUI: GUI je Graphical User Interface. Git GUIs su programi koji nude interaktivno Git iskustvo. Neke od stvari koje GUI nudi je vizualizacija trenutnog stanja vašeg repozitorijuma, istoriju commit-ova i branch-eva. Umesto kucanja komandi u terminal, možete kliknuti na označena polja i meni liste.

Git CLI Primer - Git Bash



```
MINGW32:~/git
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

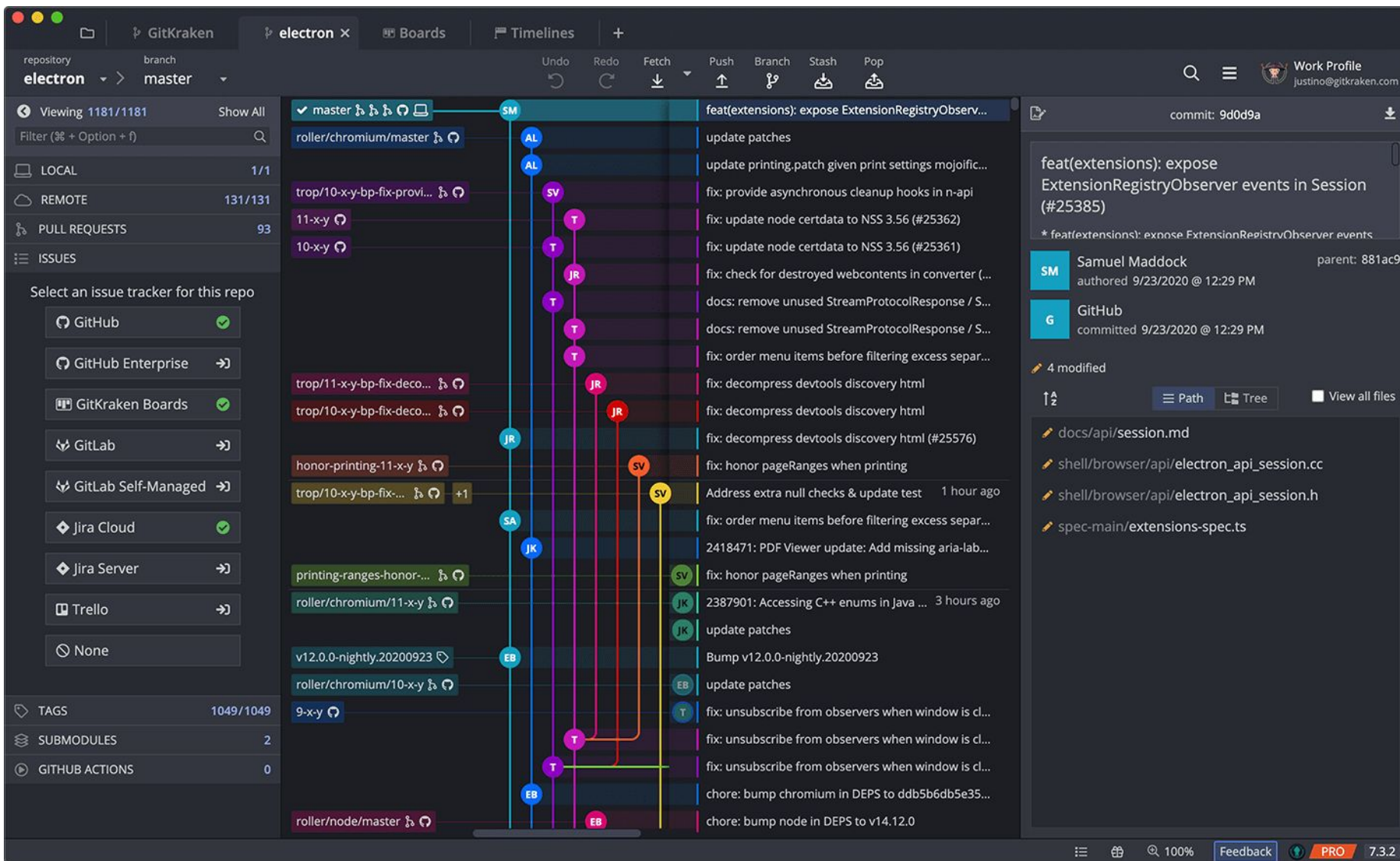
Bacon@BACON ~/git (master)
$
```

GUI Opcije





GitKraken





git



GitHub

Git vs Github

Da li je Git isto kao i GitHub?

- Git i GitHub su dve različite stvari.
- Git je open-source version control alatka kreirana 2005 godine od strane programera na Linux operativnom sistemu.
- GitHub je kompanija osnovana 2008 godine, koja razvija alatku čija je svrha integracija sa Git-om.
- Nije vam potreban GitHub kako bi koristili Git, ali ne možete koristiti GitHub bez Git-a.
- Postoje mnoge alternative GitHub-a, poput GitLab-a, BitBucket-a itd.
- Sve ove platforme se mogu nazvati "remote", ne morate da ih koristite ali će vam definitivno olakšati deljenje koda i lakšu saradnju sa ostalima.

TL;DR

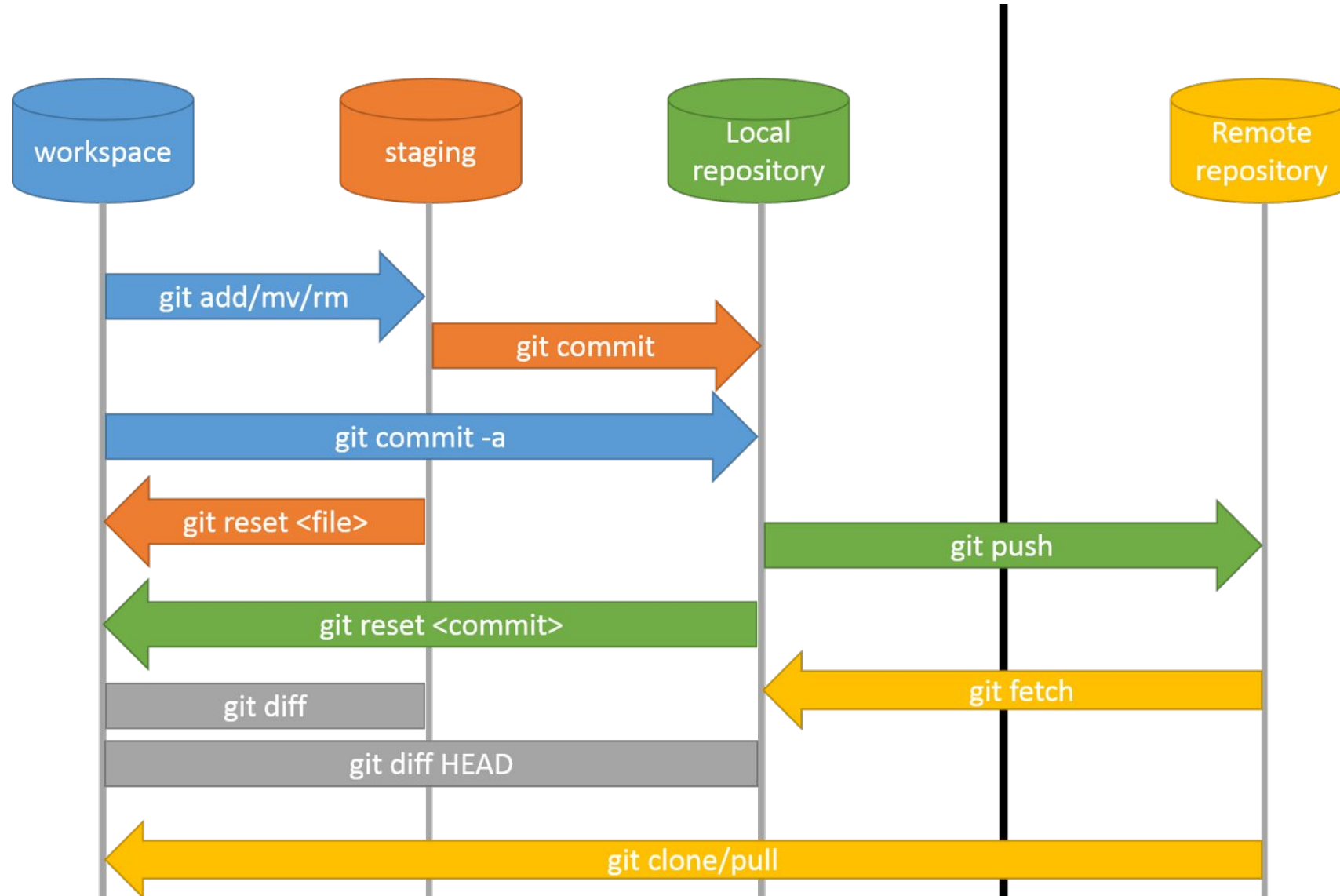
Git je version control sistem, GitHub je platforma za deljenje vašeg koda.

Šta je GitHub?

Uvod u GitHub

- GitHub je profitabilna kompanija koja nudi cloud-based skladištenje za Git repozitorijume.
- GitHub-ov interface je vrlo user-friendly, što omogućava i početnicima u programiranju da lako savladaju i koriste sve mogućnosti koje nudi Git.
- Bez GitHub-a, korišćenje Git-a generalno zahteva više tehničkog znanja. GitHub je toliko fleksibilan, da je redovna pojava da ljudi koriste GitHub za vođenje i drugih tipova projekata poput recimo - pisanja knjiga.
- Svako može da se registruje i da host-uje javni repozitorijum besplatno, što čini GitHub vrlo popularnim kao jednog od open-source projekata.
- Kao kompanija, glavni izvor zarade za GitHub je prodavanje skladišta za privatne repozitorijume, kao i jasno definisane planove za različite korporativne tipove kompanija.

Workflow



Hvala na pažnji

Git Osnove - Online Radionica 2021

