LINKgroup

Distance Learning System

Alati za kolaboraciju

Python and programming fundamentals

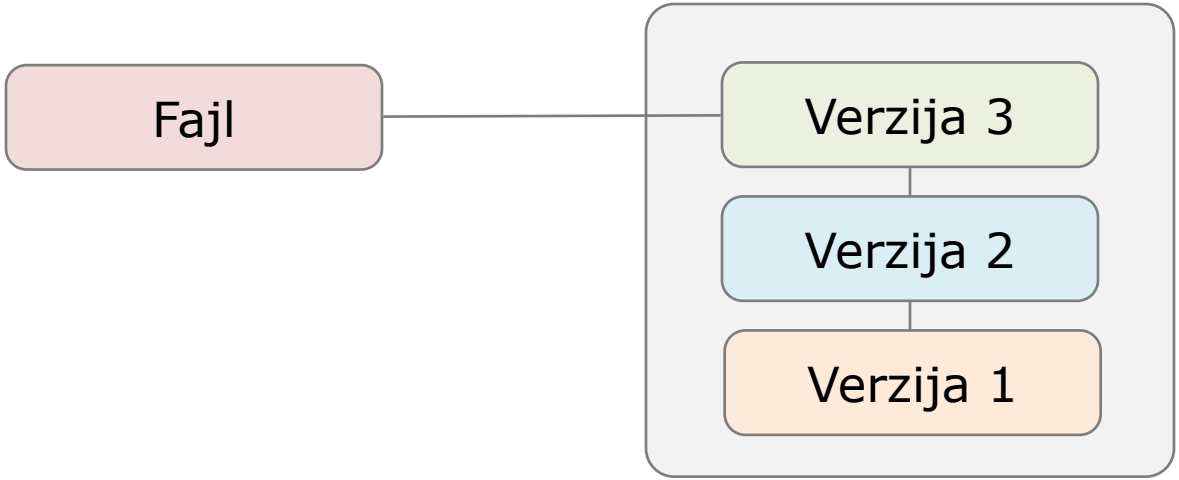
Verzioniranje (version control)

https://git-scm.com/book/sr/v2

- Verzioniranje omogućava čuvanje prethodnih verzija jednog ili grupe fajlova
- Verzioniranje obezbeđuje veću pouzdanost u radu jer je moguće lako povratiti prethodne verzije oštećenih ili obrisanih fajlova
- Verzioniranje se može obavljati **lokalno**, **centralizovano** ili **distribuirano**

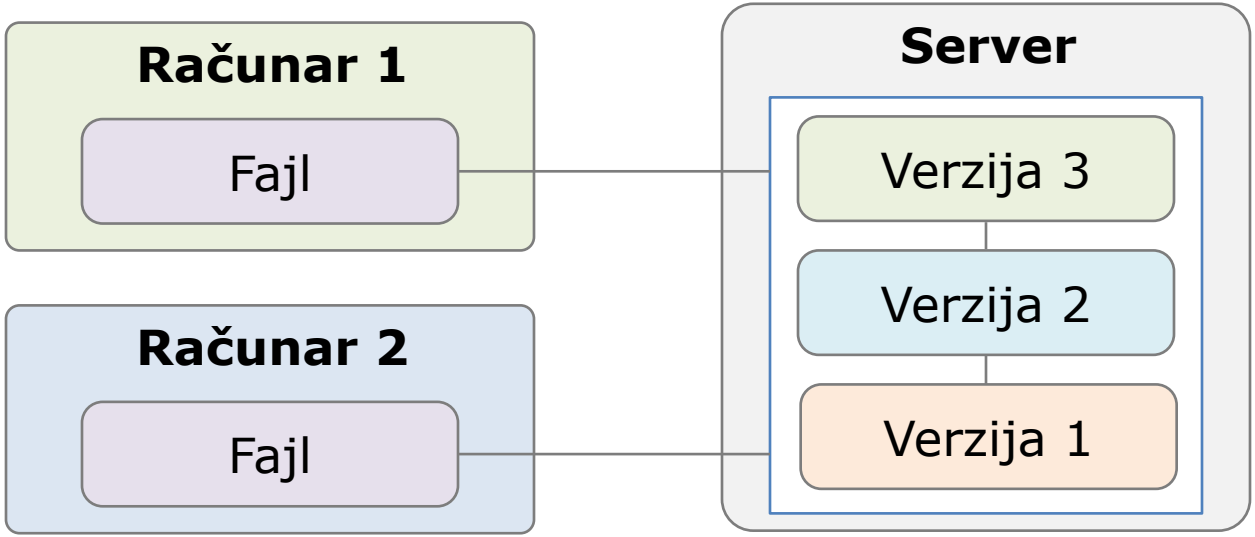
Lokalno verzioniranje

- Verzije fajlova se čuvaju lokalno
- Prilikom gubitka ili oštećenja fajla, prethodna verzija se preuzima iz lokalnog repozitorijuma



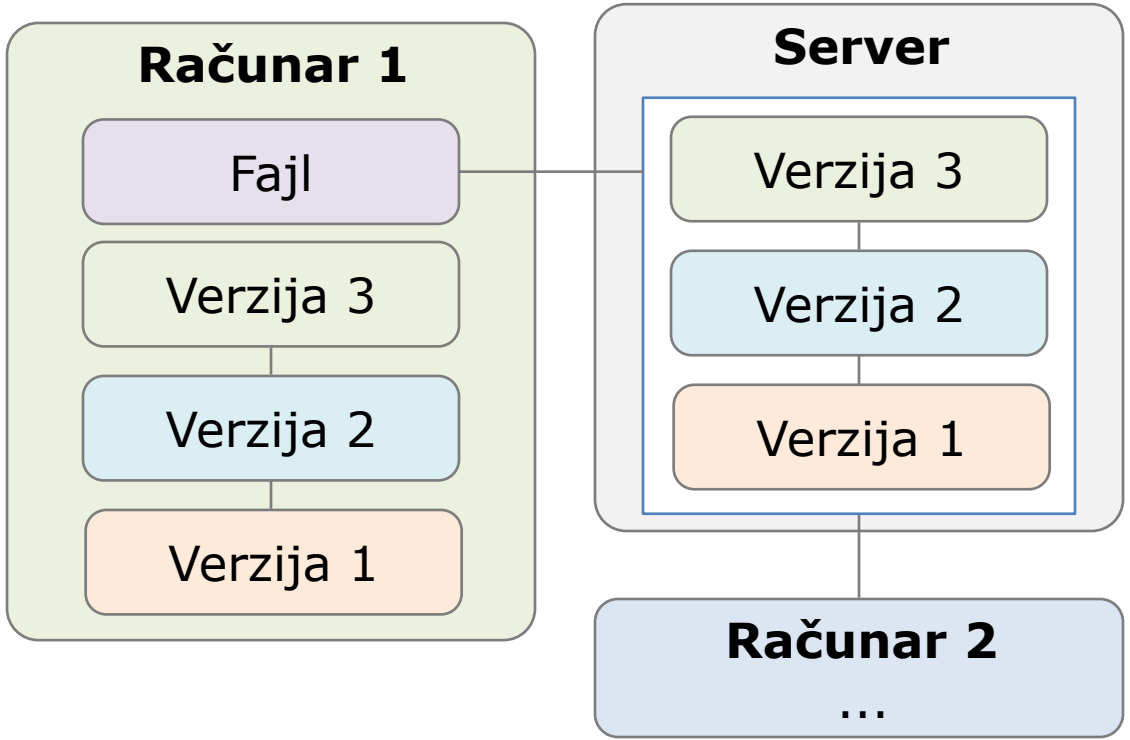
Centralizovano verzioniranje

- Verzije se čuvaju na udaljenom računaru
- Olakšava se kolaboracija
- Ne postoje lokalne verzije



Distribuirano verzioniranje

- Verzije se čuvaju na udaljenom računaru i na lokalnim računarima
- Olakšava se kolaboracija
- Postoje lokalne verzije



Alati za verzioniranje

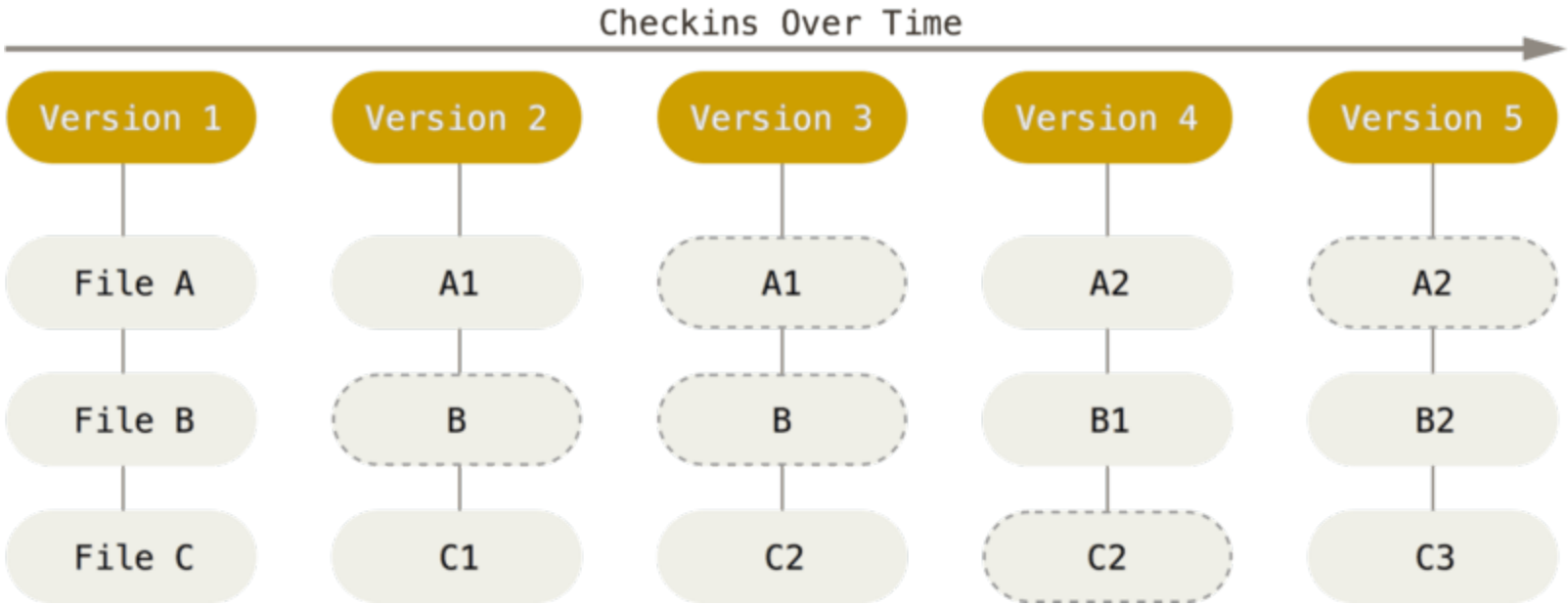
- Da se ne bi radilo ručno, u svrhu verzioniranja se koriste različiti alati
- Među najpoznatijim alatima za verzioniranje su **git** i **svn**, iako postoje i mnogi drugi (cvs, mercurial, visual studio team services)

Git

- Jedan od najpopularnijih alata za verzioniranje danas
- Inicijalno kreiran za Linux okruženje, ali se koristi i na Windows-u
- Brz
- Jednostavan za korišćenje
- Dobro dokumentovan

Čuvanje verzija

- U bazi se čuvaju snapshotovi projekta, pri čemu se nepromenjeni fajlovi referenciraju



Integritet podataka

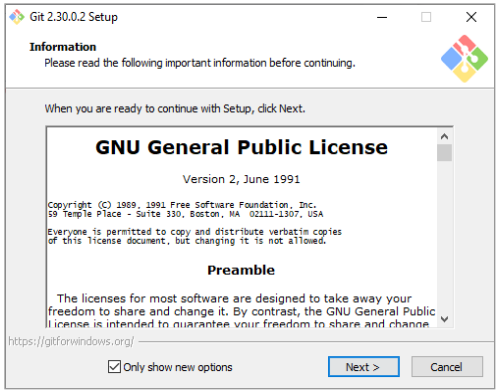
- Za sve fajlove praćene git-om, kreira se **checksum**
- Na osnovu checksum-a prate se fajlovi kontrolisani git sistemom
- Za izračunavanje checksum-a, git koristi algoritam SHA-1
- Na primer, poruka: hello world , provučena kroz SHA-1 funkciju glasi uvek: 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed

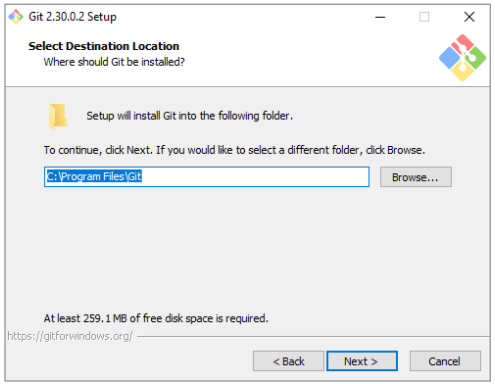
Instalacija git-a

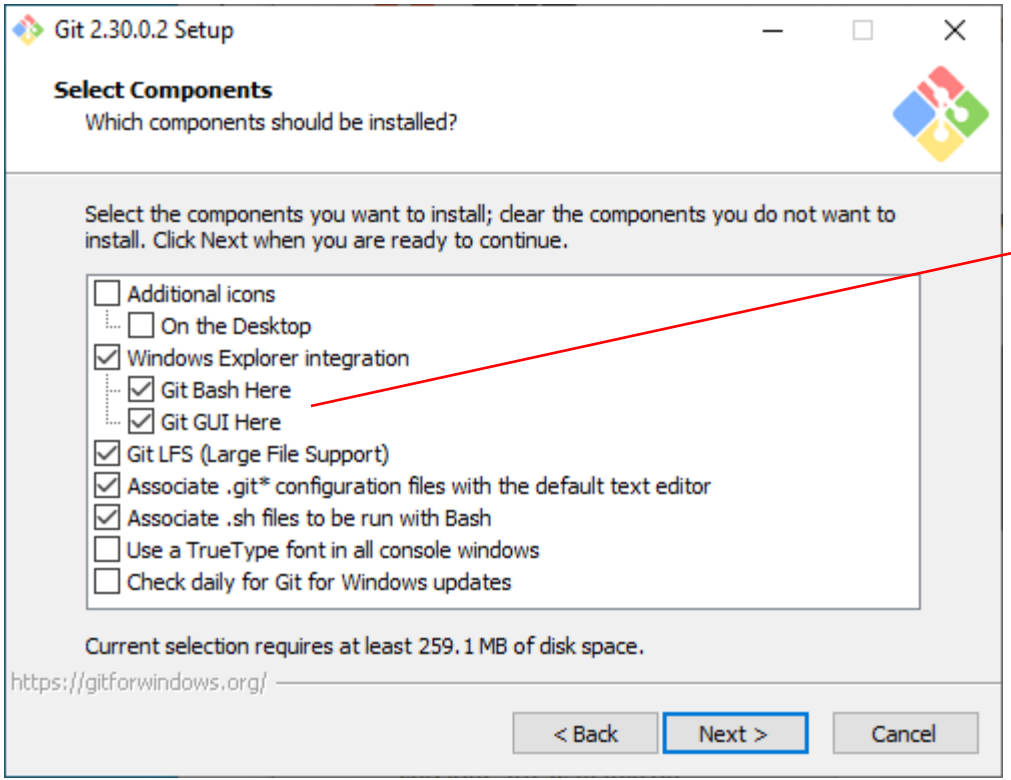
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

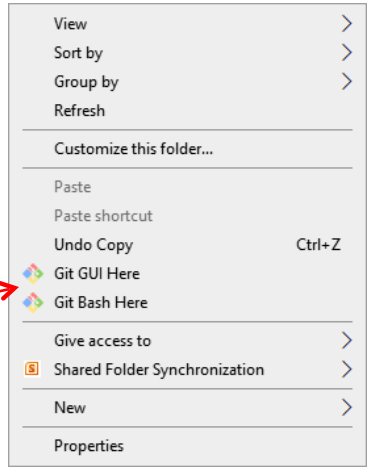
- Git za windows je moguće preuzeti sa adrese **https://git-scm.com**

Instalacija git-a

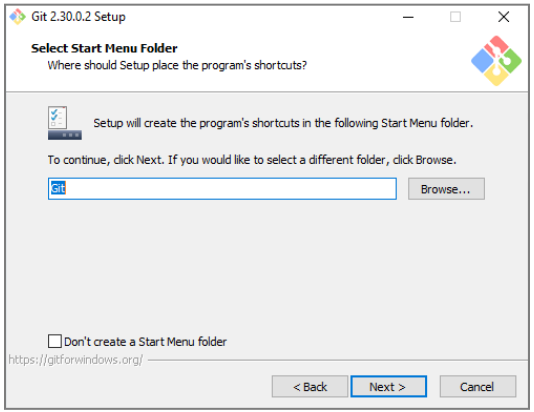


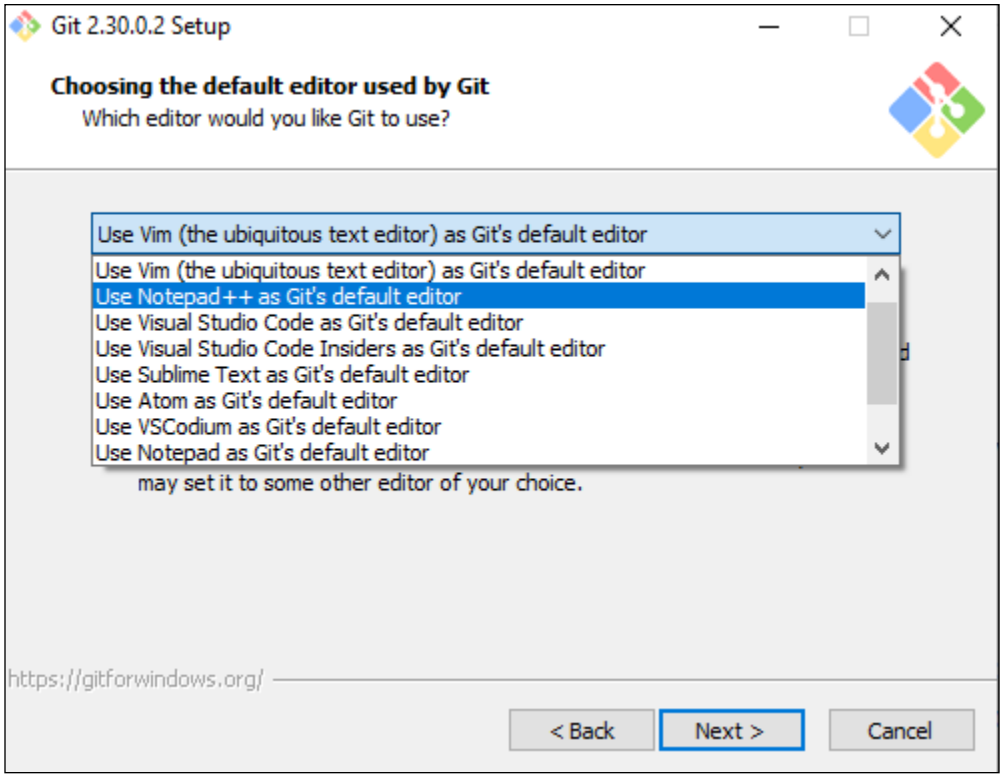




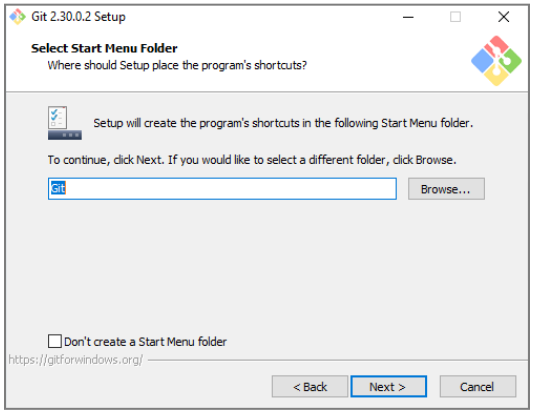


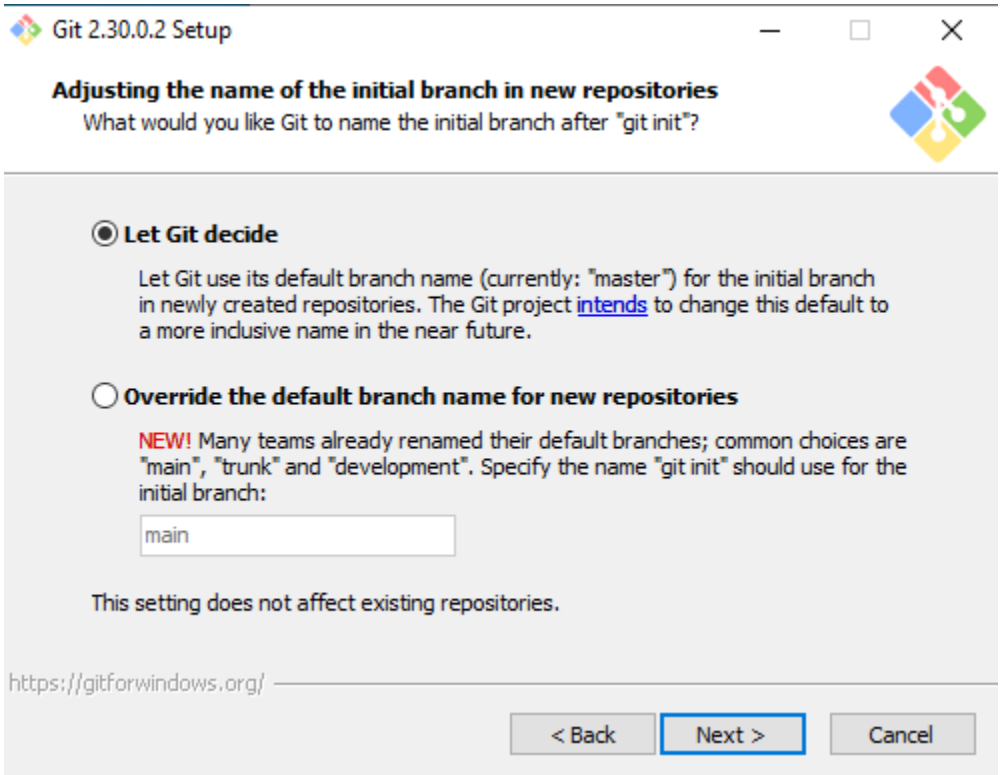
Instalacija git-a



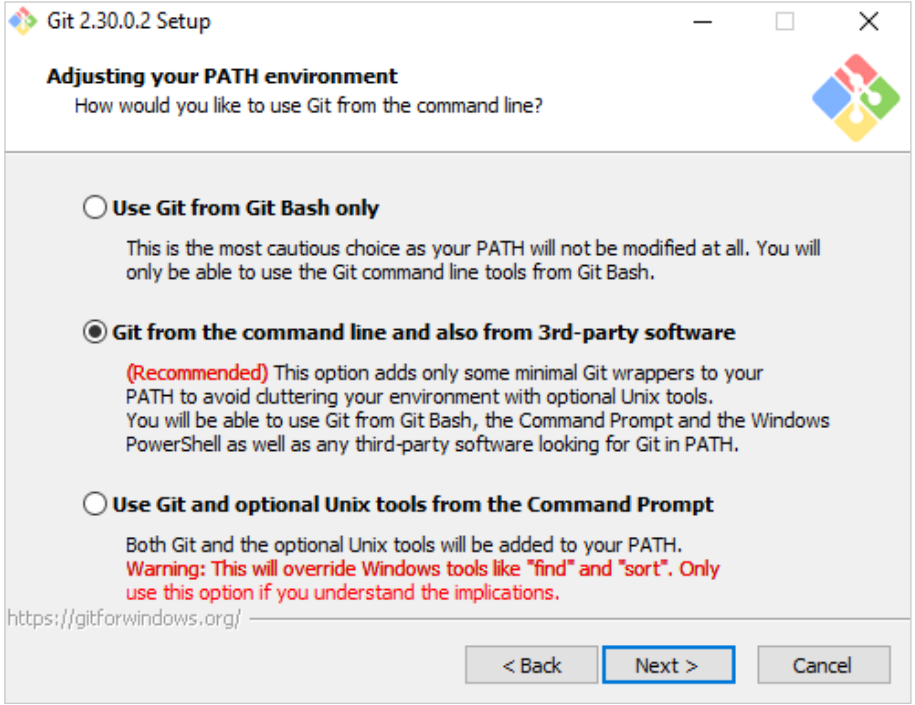


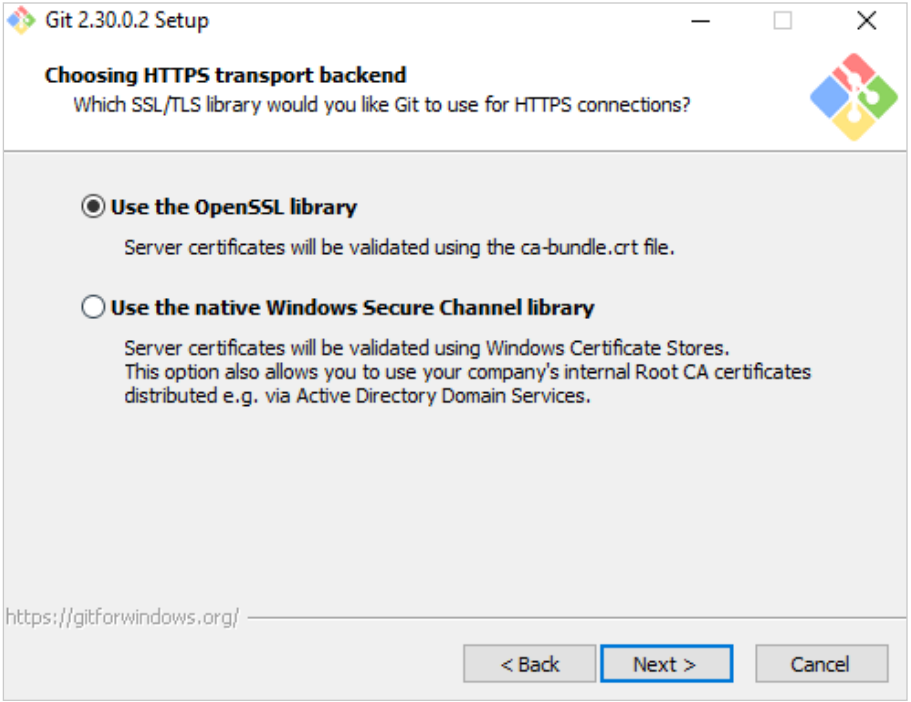
Instalacija git-a



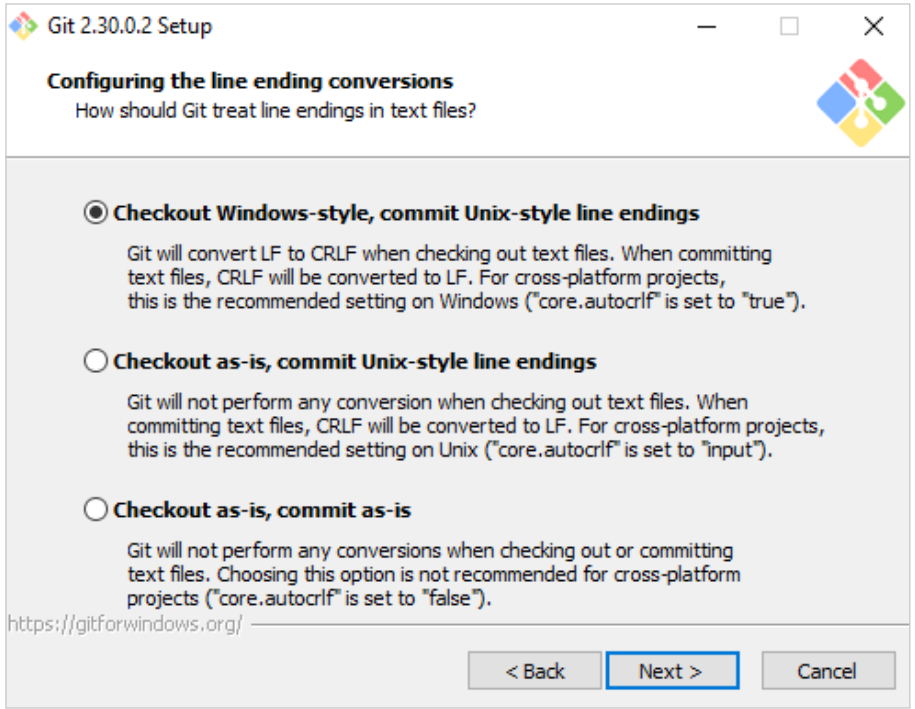


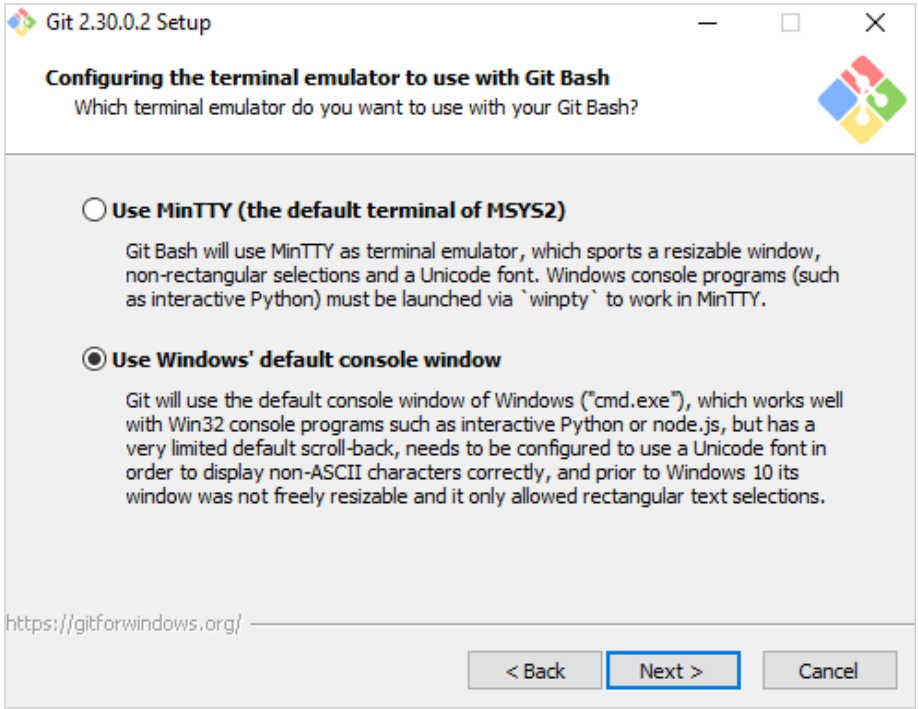
Instalacija git-a



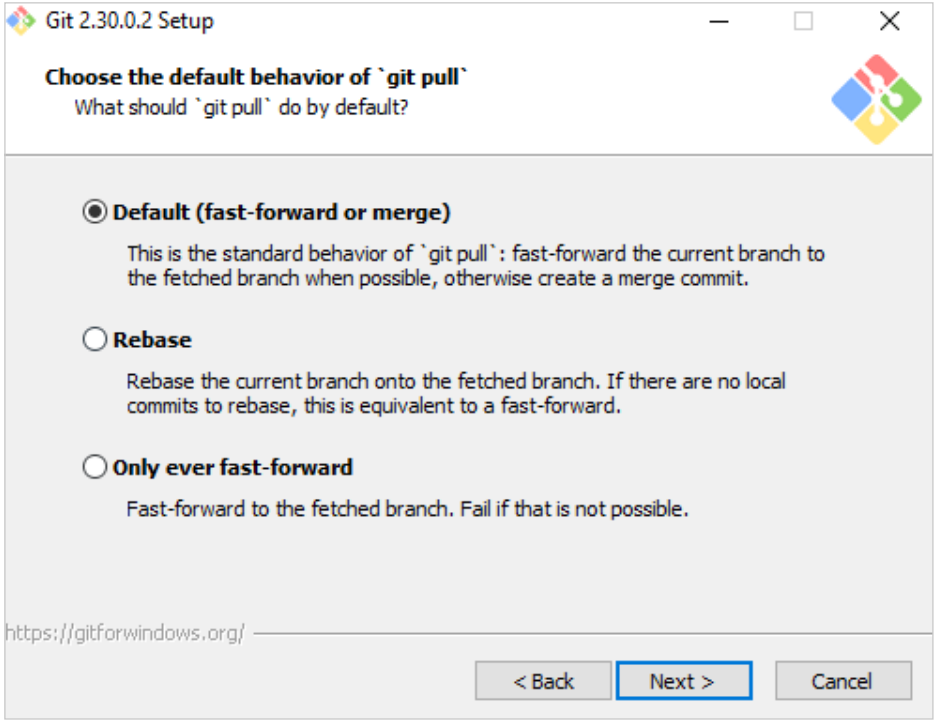


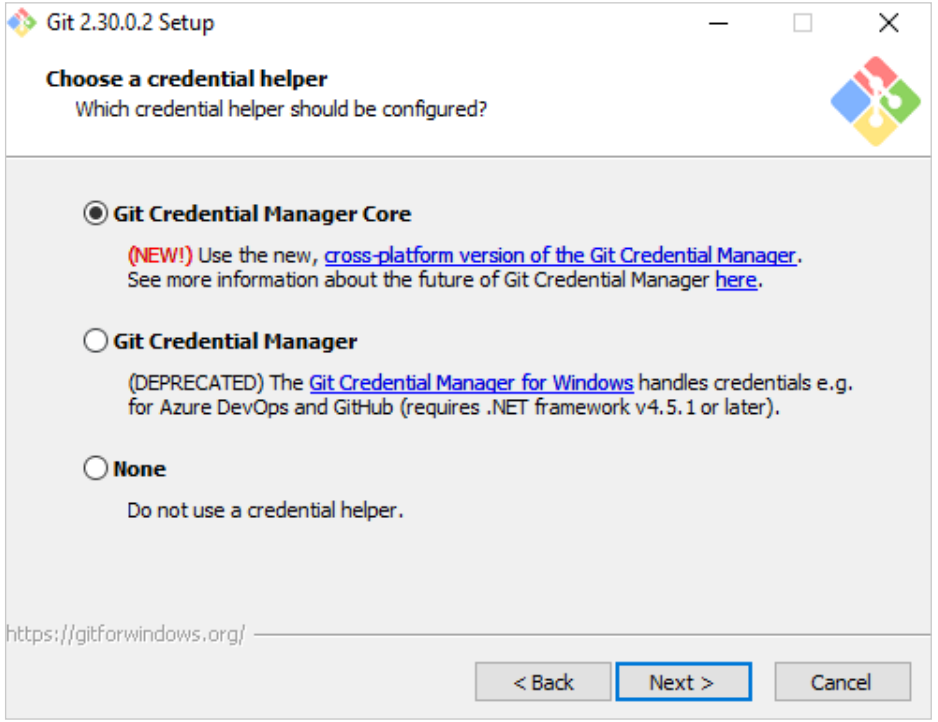
Instalacija git-a



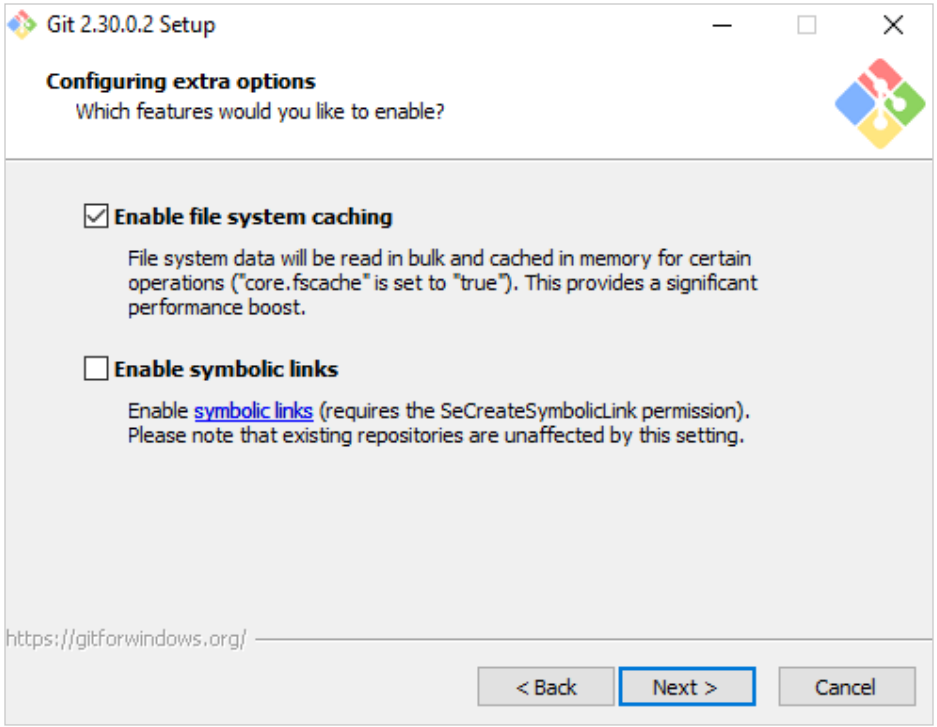


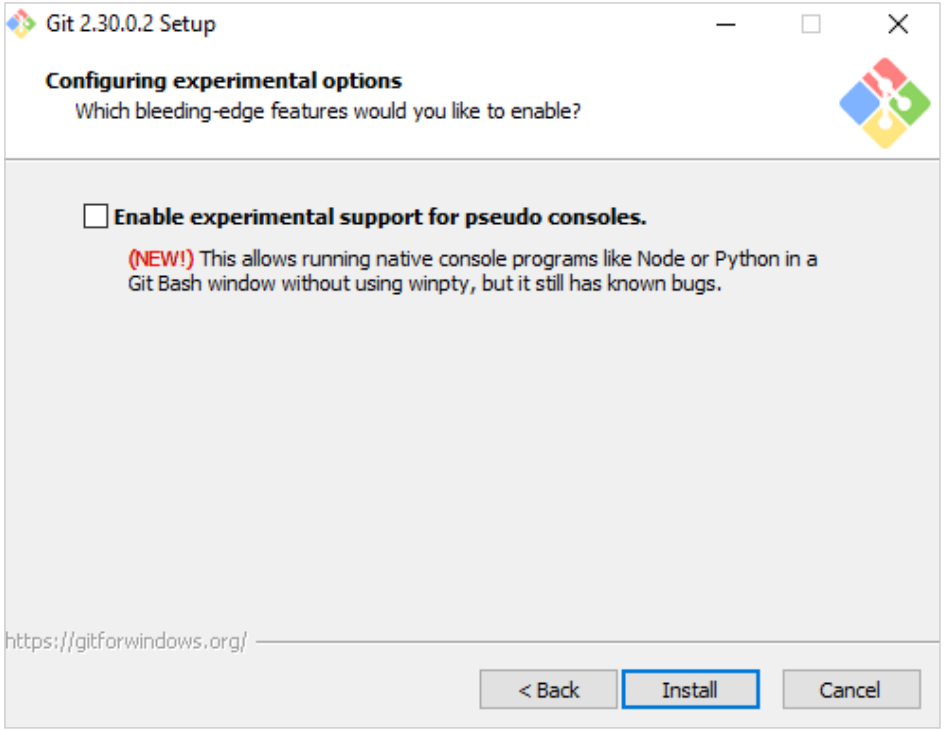
Instalacija git-a





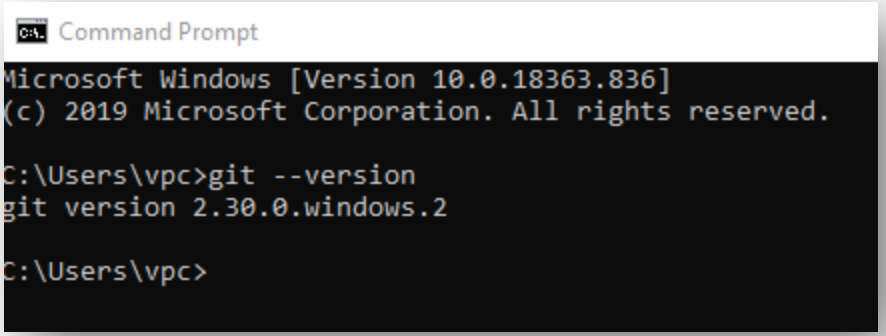
Instalacija git-a





Instalacija git-a

- Za proveru verzije alata git kuca se komanda: git --version



- Komandni prompt treba otvoriti tek nakon kompletirane instalacije

Git upotreba

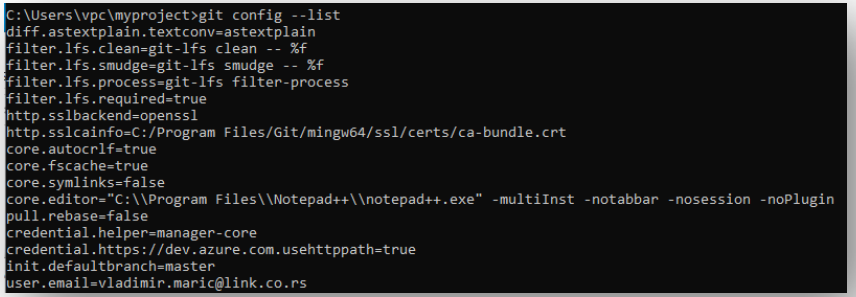
- Program git sadrži mnoštvo dodatnih alata ali je njegova osnovna komanda: **git**
- Komande se koriste u kombinaciji sa komandom git:
- Primer:

git help git
git help tutorial
git add .
git mergetool

Git postavke

- Git je potrebno podesiti kako bi ispravno funkcionisao
- Većina postavki obavlja se komandom **git config**
- Konfiguracija se postavlja globalno i lokalno, pri čemu svako lokalno podešavanje prepisuje istoimeno globalno podešavanje
- Podešavanja se nalaze u fajlu **.gitconfig** u okviru korisničkog direktorijuma ili u fajlu **config**, lokalnog direktorijuma **.git**
- Prikaz podešavanja vrši se komandom:

git config --list



Git postavke

- Da bi git ispravno funkcionisao, neophodno je takođe postaviti email i korisničko ime korisnika

```
[user]
        name = Dovla
        email = vladimir.maric@link.co.rs
```

Globalno

```
git config --global user.email "vladimir.maric@link.co.rs"
```

Lokalno

```
git config user.email "vladimir.maric@link.co.rs"
```

Inicijalizacija projekta

- Da bi neki projekat (direktorijum) bio praćen od strane git-a, potrebno je da bude inicijalizovan
- Inicijalizacija podrazumeva kreiranje poddirektorijuma **.git** u okviru direktorijuma projekta
- Inicijalizovani git projekat, naziva se **repozitorijum**
- Inicijalizacija se vrši na jedan od dva načina:

Kreiranjem repozitorijuma u okviru radnog direktorijuma

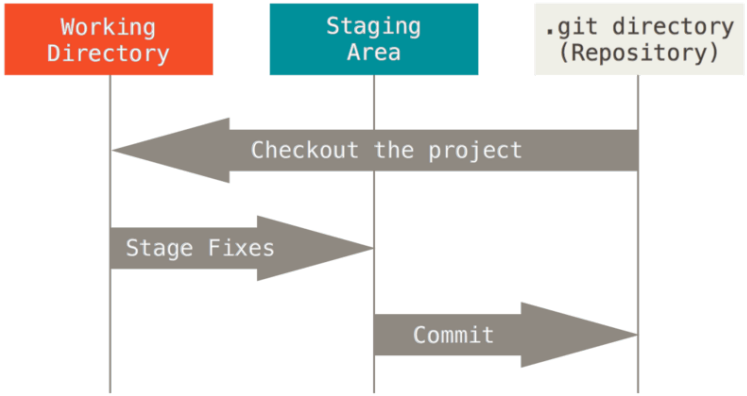
```
C:\Users\vpc\myproject>git init
Initialized empty Git repository in C:/Users/vpc/myproject/.git/
```

Kloniranjem postojećeg repozitorijuma (naredba git clone)

```
C:\Users\vpc>git clone https://vmaric@bitbucket.org/vmaric/qa_test_repo.git
Cloning into 'qa_test_repo'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), 580 bytes | 96.00 KiB/s, done.

C:\Users\vpc>
```

Statusi fajlova

- Fajlovi u okviru repozitorijuma mogu biti u jednom od tri statusa:
- Izmenjeni (**modified**)
  Fajl je dodat ili izmenjen ali još uvek nije potvrđen
- Postavaljeni na scenu (**staged**)
  Fajl je registrovan za praćenje
- Potvrđeni (**committed**)
  Izmene izvršene nad fajlom su sačuvane u repozitorijumu
- Status fajlova u repozitorijumu moguće je videti komandom: **git status**




```
C:\Users\vpc\myproject>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        myfirstfile.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Dodavanje fajlova

- Da bi fajlovi u inicijalizovanom direktorijumu bili praćeni, moraju biti dodati
- Dodavanje fajlova se vrši komandom **git add**

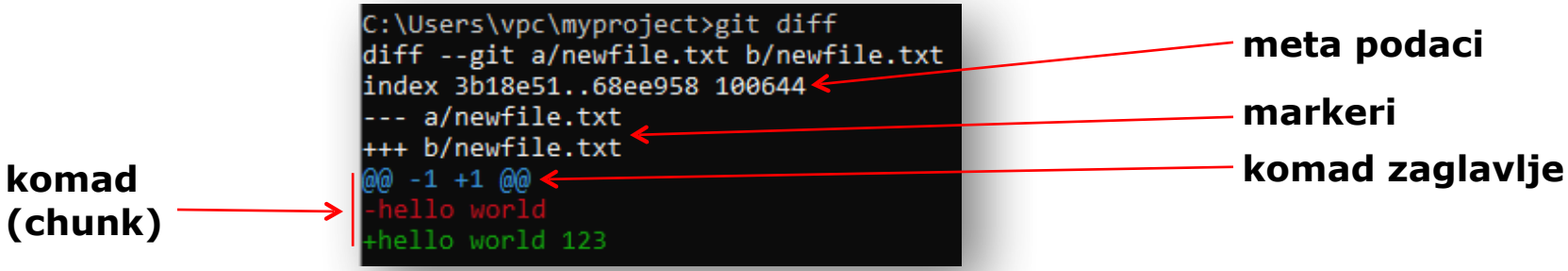
```
C:\Users\vpc\myproject>git add myfirstfile.txt

C:\Users\vpc\myproject>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   myfirstfile.txt
```

- Prestanak praćenja fajlova vrši se komandom **git rm**

```
C:\Users\vpc\myproject>git rm --cached myfirstfile.txt
rm 'myfirstfile.txt'
```

Pregled izmena u fajlovima

- Komanda **git diff** prikazuje izmene po fajlovima
- Prikazuju se izmene u fajlovima koji nisu dodati u praćenje



- Za izmene nad fajlovima koji su dodati u praćenje komanda se koristi sa zastavicom **staged**

```
C:\Users\vpc\myproject>git diff --staged
diff --git a/newfile.txt b/newfile.txt
new file mode 100644
index 0000000..3b18e51
--- /dev/null
+++ b/newfile.txt
@@ -0,0 +1 @@
+hello world
```

Potvrđivanje izmena fajlova

- Izmene napravljene nad fajlovima koje dodamo u praćenje, potvrđujemo naredbom: **git commit**
- Potvrđivanje mora biti praćeno porukom koja obično sadrži kratak opis potvrđenih izmena

```
C:\Users\vpc\myproject>git commit -m "first commit"
[master (root-commit) cc7b8be] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 myfirstfile.txt
```

- Ukoliko se komanda startuje bez zastavice **m**, otvara se podrazumevani tekst editor u koji unosimo poruku:

```
My first commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   modified:   newfile.txt
#
```

Preskakanje staging faze

- Moguće je automatsko priključivanje fajlova praćenju i potvrđivanje, pomoću zastavice –a nad komandom git commit

```
C:\Users\vpc\myproject>git commit -a -m "commit with stage"
[master da2dbae] commit with stage
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Pregled istorije potvrđivanja

- Komanda **git log** prikazuje istoriju svih dosadašnjih potvrđivanja
- Od vrha na dole listaju se najnovija potvrđivanja

```
C:\Users\vpc\myproject>git log
commit 92d66f965233d636d8385b61ff52b9c72d99eb6a (HEAD -> master)
Author: unknown <vladimir.maric@link.co.rs>
Date:   Tue Feb 9 04:40:44 2021 -0800

    ...

commit 3fd261a7840f2bb39333e50321002dfb1719c5e7
Author: unknown <vladimir.maric@link.co.rs>
Date:   Tue Feb 9 04:18:52 2021 -0800

    rm

commit da2dbaecfb65dd0b59bc9785ef60cf35ea8d56db
Author: unknown <vladimir.maric@link.co.rs>
Date:   Tue Feb 9 04:15:12 2021 -0800

    commit with stage

commit db24b7a47b0928c8a9a874fe80606592c088e946
Author: unknown <vladimir.maric@link.co.rs>
Date:   Tue Feb 9 04:10:25 2021 -0800

    My first commit

commit 38e26d71f46562e3175efcfaf488378216abd6a3
Author: unknown <vladimir.maric@link.co.rs>
Date:   Tue Feb 9 04:08:07 2021 -0800

    My first commit
```

Izmena postojećeg commit-a

- Dodavanje izmena u postojeći commit

```
C:\Users\vpc\myproject>git commit -m "some commit"
[master bf7cadd] some commit
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\vpc\myproject>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        mf1.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\vpc\myproject>git add .

C:\Users\vpc\myproject>git commit --amend
[master 9547625] some commit
 Date: Wed Feb 10 03:37:17 2021 -0800
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 mf1.txt
```

Izmena postojećeg commit-a

- Uklanjanje fajla sa stage-a

```
C:\Users\vpc\myproject>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   mf1.txt
        modified:   myfile.txt


C:\Users\vpc\myproject>git restore --staged myfile.txt

C:\Users\vpc\myproject>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   mf1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   myfile.txt
```

Vreaćanje fajla u inicijalni status

```
C:\Users\vpc\myproject>echo Hello > myfile.txt

C:\Users\vpc\myproject>git add .

C:\Users\vpc\myproject>echo World >> myfile.txt

C:\Users\vpc\myproject>git checkout -- myfile.txt

C:\Users\vpc\myproject>type myfile.txt
Hello

C:\Users\vpc\myproject>
```

Vraćanje na prethodni commit

- Komanda **git reset** omogućava kretanje kroz commit-ove
- Koristi se u kombinaciji sa putanjom ili hash-om commit-a
- Koristi se u jednoj od tri varijante:
-

git reset --soft

Stanje se postavlja na fazu posle stage-a a pre commit-a

git reset --mixed

Stanje se setuje na fazu pre stage-a

git reset --hard

Stanje se setuje na fazu nakon prethodnog commit-a

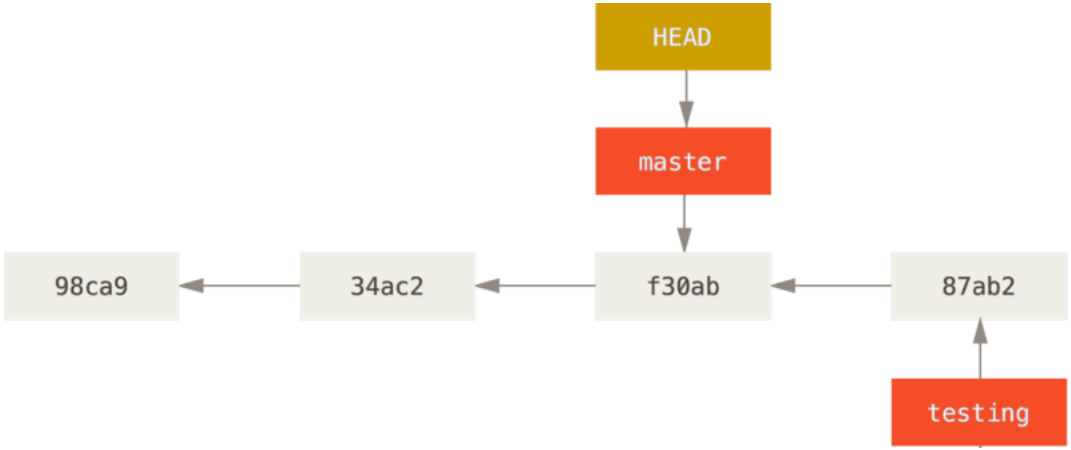
Vraćanje na prethodni commit

```
C:\Users\vpc\myproject>echo First Line > myfile.txt

C:\Users\vpc\myproject>git add myfile.txt

C:\Users\vpc\myproject>git commit -m "First Commit"
[master (root-commit) 5fe1274] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 myfile.txt

C:\Users\vpc\myproject>echo Second Line >> myfile.txt

C:\Users\vpc\myproject>git commit -a -m "Second Commit"
[master 21cd26b] Second Commit
 1 file changed, 1 insertion(+)
```

```
C:\Users\vpc\myproject>git commit -a -m "Third Commit"
[master 9cb55e8] Third Commit
 1 file changed, 1 insertion(+)

C:\Users\vpc\myproject>git reset --hard HEAD~
HEAD is now at 21cd26b Second Commit

C:\Users\vpc\myproject>type myfile.txt
First Line
Second Line

C:\Users\vpc\myproject>
```

Grananje

- Grananjem se omogućava kreiranje različitih razvojnih linija istog radnog direktorijuma
- Jedna grana je zapravo referenca na određeni commit
- Podrazumevana grana u Git-u je master
- Moguće je kreirati i neograničen broj dodatnih grana



Naredbe grananja

- Naredba **git branch** kreira novu granu u repozitorijumu (parametar komande je naziv nove grane)

```
C:\Users\vpc\myproject>git branch test_branch

C:\Users\vpc\myproject>
```

- Kreiranje nove grane ne aktivira tu granu automatski

```
C:\Users\vpc\myproject>git status
On branch master
nothing to commit, working tree clean
```

- Lista postojećih grana može se videti takođe pomoću komande **git branch** (bez parametara)

```
C:\Users\vpc\myproject>git branch
* master
  test_branch
```

- Prebacivanje sa grane na granu, obavlja se komandom **git checkout**

```
C:\Users\vpc\myproject>git checkout test_branch
Switched to branch 'test_branch'

C:\Users\vpc\myproject>git status
On branch test_branch
nothing to commit, working tree clean

C:\Users\vpc\myproject>
```

Spajanje grana - merge

- Grane se u bilo kom trenutku mogu spojiti
- Za spajanje grana koriste se komande **git merge** i **git rebase**

```
C:\Users\vpc\myproject>git init
Initialized empty Git repository in C:/Users/vpc/myproject/.git/

C:\Users\vpc\myproject>echo Hello > myfile.txt

C:\Users\vpc\myproject>git add .

C:\Users\vpc\myproject>git commit -m "first commit"
[master (root-commit) 078e9b5] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 myfile.txt

C:\Users\vpc\myproject>echo World >> myfile.txt

C:\Users\vpc\myproject>git commit -a -m "Second commit"
[master c70c20e] Second commit
 1 file changed, 1 insertion(+)

C:\Users\vpc\myproject>git checkout -b testing
Switched to a new branch 'testing'

C:\Users\vpc\myproject>echo Hello again > yourfile.txt

C:\Users\vpc\myproject>git add .

C:\Users\vpc\myproject>git commit -m "first commit on test branch"
[testing eef8a86] first commit on test branch
 1 file changed, 1 insertion(+)
 create mode 100644 yourfile.txt
```

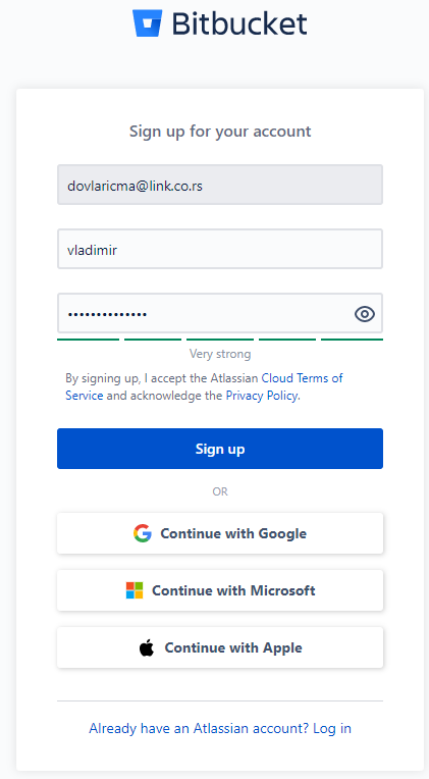
```
C:\Users\vpc\myproject>git checkout master
Switched to branch 'master'

C:\Users\vpc\myproject>git merge testing
Updating c70c20e..eef8a86
Fast-forward
 yourfile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 yourfile.txt

C:\Users\vpc\myproject>dir
 Volume in drive C has no label.
 Volume Serial Number is 3EB1-9BB3

 Directory of C:\Users\vpc\myproject

02/22/2021  10:55 PM    <DIR>          .
02/22/2021  10:55 PM    <DIR>          ..
02/22/2021  10:53 PM                16 myfile.txt
02/22/2021  10:55 PM                14 yourfile.txt
               2 File(s)             30 bytes
               2 Dir(s)  14,619,041,792 bytes free
```

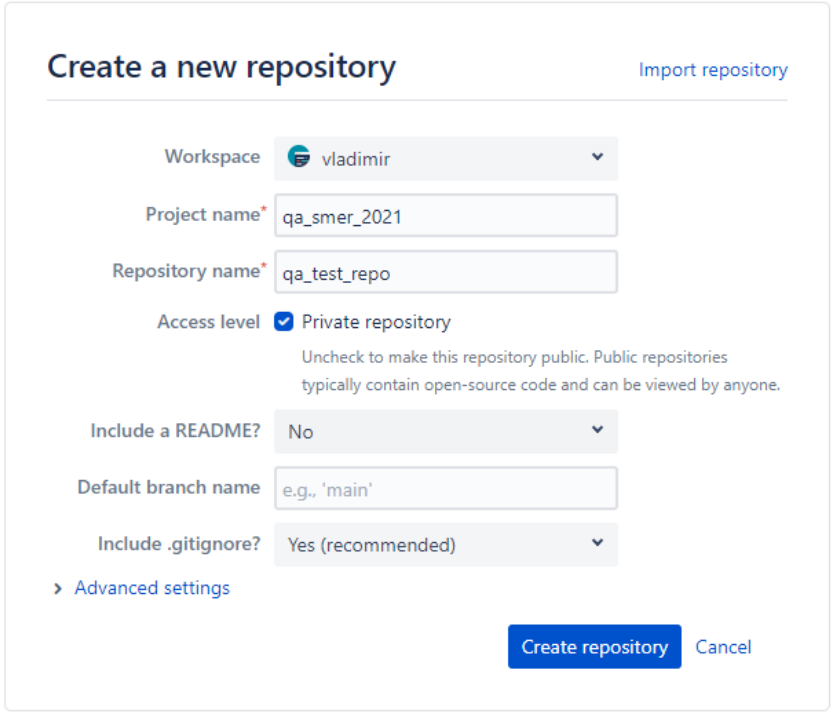
Rukovanje udaljenim repozitorijumom

- Kada se koristi za kolaboraciju, git podrazumeva upotrebu udaljenog repozitorijuma
- Udaljeni repozitorijum se nalazi na udaljenom računaru
- Udaljeni repozitorijumi mogu biti **samo hostovani** ili se mogu koristiti **javni servisi**
- Poznatiji javni git servisi su:

  Github
  Bitbucket
  SourceForge
  AWS CodeCommit

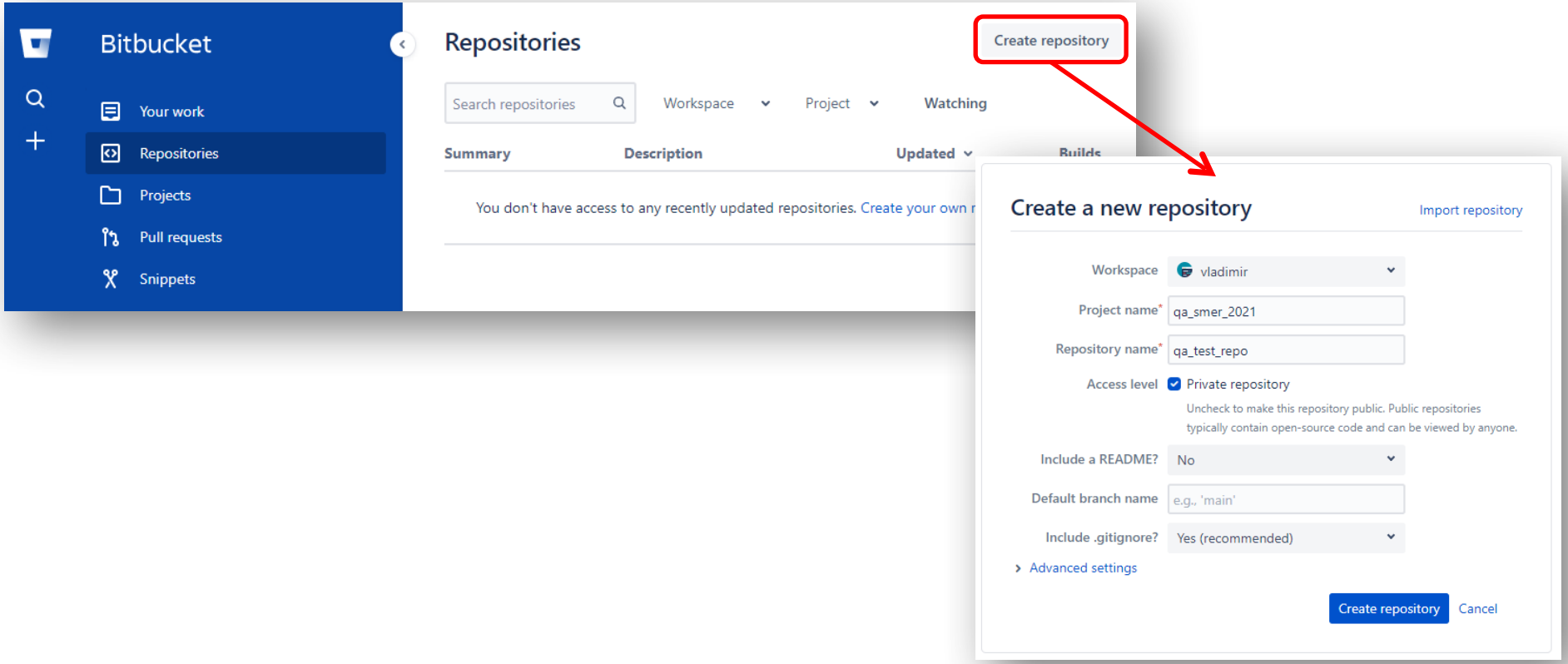
Rukovanje Bitbucket platformom



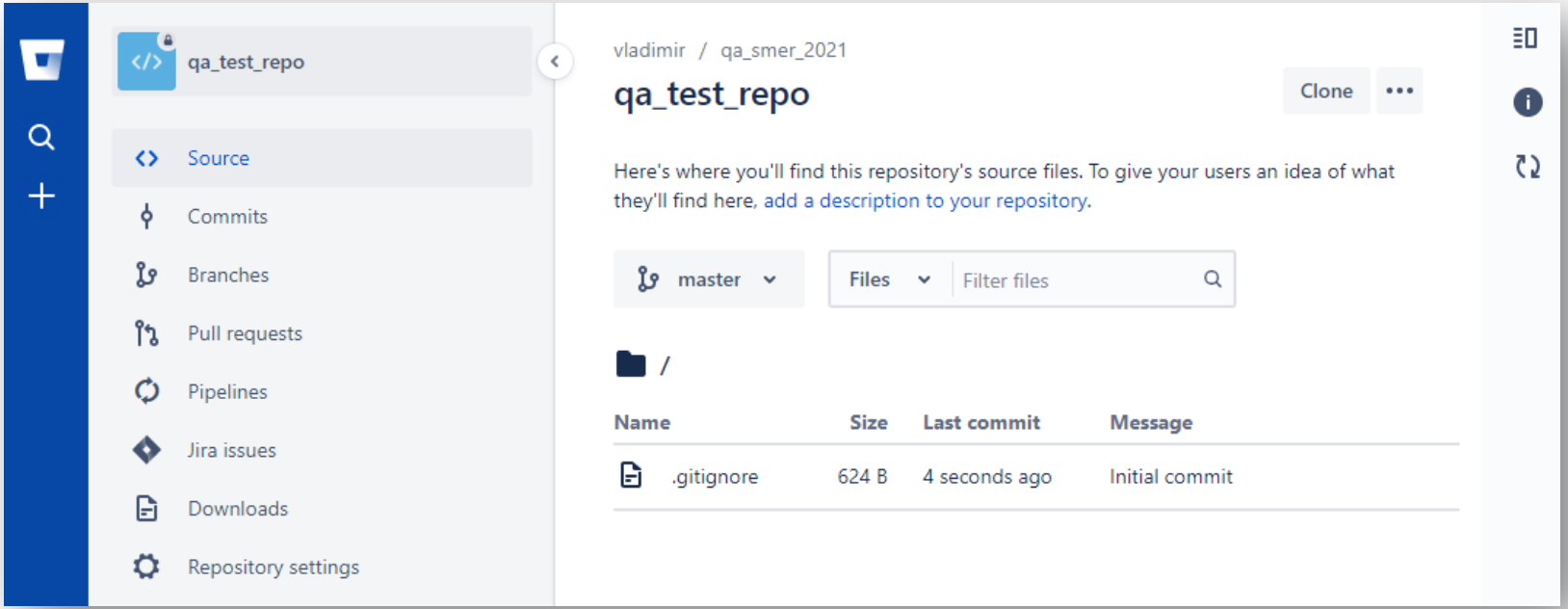


LINKgroup

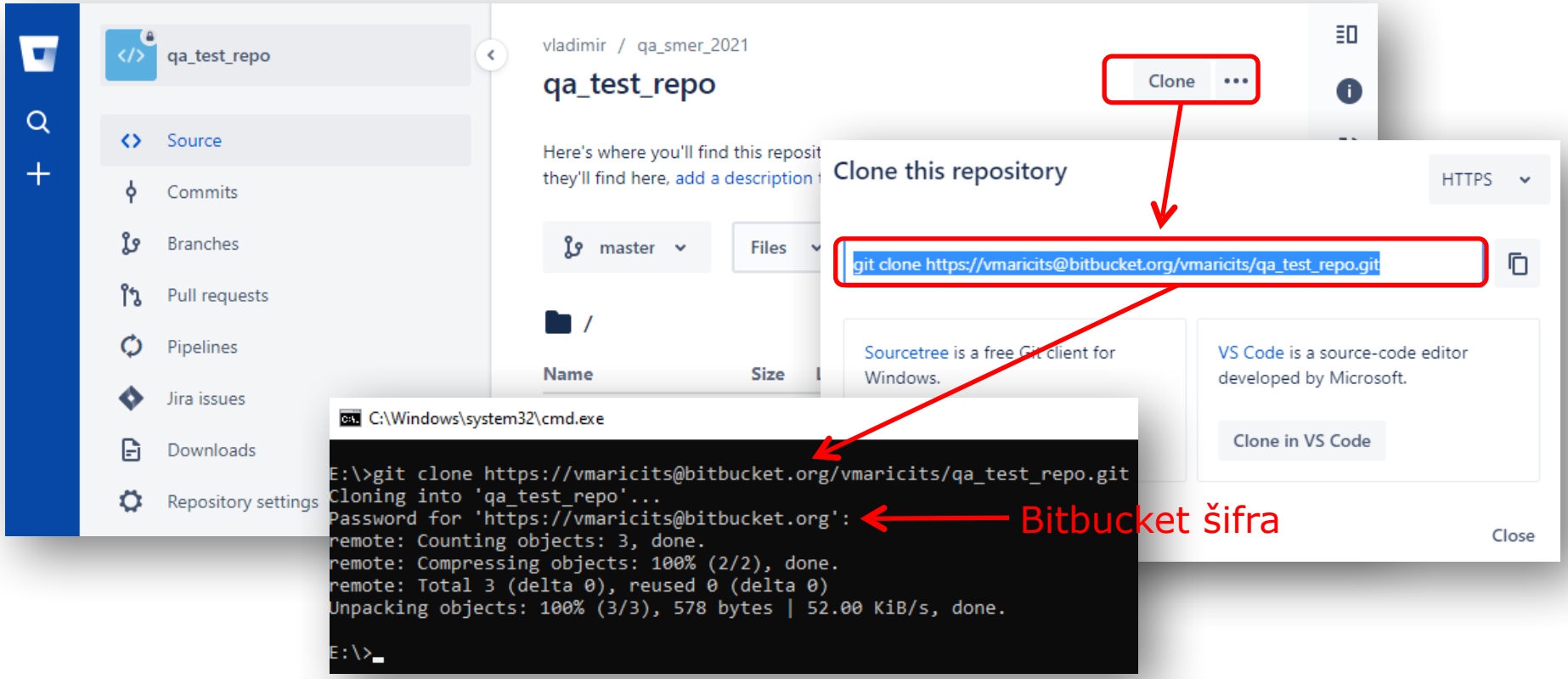
Kreiranje bitbucket repozitorijuma



Kreiranje bitbucket repozitorijuma



Kloniranje repozitorijuma



Rukovanje udaljenim repozitorijumom

- Ukoliko je repozotorijum preuzet pomoću komande git clone, udaljeni server je registrovan automatski, u suprotnom, udaljeni server se mora registrovati ručno komandom **git remote**

```
C:\Users\vpc\myproject>git remote add qa_test https://vmaric@bitbucket.org/vmaric/qa_test_repo.git

C:\Users\vpc\myproject>git remote -v
qa_test https://vmaric@bitbucket.org/vmaric/qa_test_repo.git (fetch)
qa_test https://vmaric@bitbucket.org/vmaric/qa_test_repo.git (push)
```

Rukovanje udaljenim repozitorijumom

- Prikaz podataka o udaljenom repozitorijumu

```
C:\Users\vpc\myproject>git remote show qa_test
* remote qa_test
  Fetch URL: https://vmaric@bitbucket.org/vmaric/qa_test_repo.git
  Push  URL: https://vmaric@bitbucket.org/vmaric/qa_test_repo.git
  HEAD branch: master
  Remote branch:
    master tracked
  Local ref configured for 'git push':
    master pushes to master (local out of date)
```

- Preuzimanje podataka sa udaljenog repozitorijuma (git fetch)

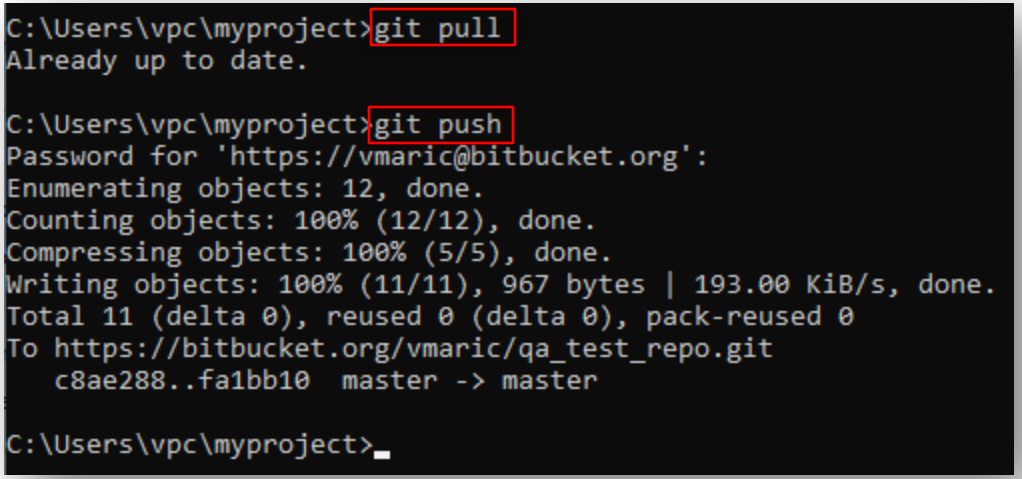
Rukovanje udaljenim repozitorijumom

- Ukoliko udaljeni repozitorijum nije preuzet naredbom git clone mora se ručno sinhronizovati sa lokalnim repozitorijumom

```
C:\Users\vpc\myproject>git pull --set-upstream --allow-unrelated-histories qa_test master
From https://bitbucket.org/vmaric/qa_test_repo
 * branch            master     -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 .gitignore | 50 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 50 insertions(+)
 create mode 100644 .gitignore
```

Rukovanje udaljenim repozitorijumom

- Komandama **push** i **pull** šaljemo izmene u lokalnom repozitorijumu na udaljeni i sa udaljenog na lokalni repozitorijum



Sakrivanje izmena

- Moguće je sakriti izmene kako bi se očistio status git-a, potvrđivanja
- Sakrivanje se obavlja komandom **git stash**

```
C:\Users\vpc\myproject>git status
On branch master
Your branch is ahead of 'qa_test/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   myfile.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\vpc\myproject>git stash
Saved working directory and index state WIP on master: 1d09abd ...

C:\Users\vpc\myproject>git status
On branch master
Your branch is ahead of 'qa_test/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\vpc\myproject>git stash apply
On branch master
Your branch is ahead of 'qa_test/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   myfile.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

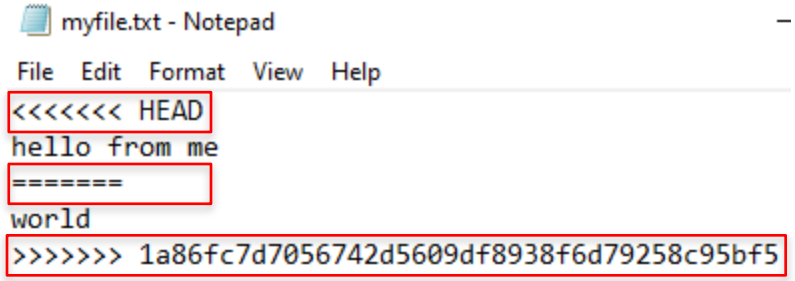
Rukovanje konfliktima

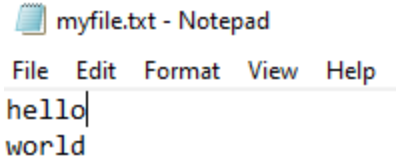
- Konflikti se obično dešavaju prilikom spajanja u kojima je na nekom fajlu rađena različita izmena od strane različitih korisnika
- Ukoliko git detektuje ovakav slučaj, odbija da kompletira spajanje pre nego što se konflikt reši

```
C:\Users\vpc\myproject>git pull
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), 223 bytes | 11.00 KiB/s, done.
From https://bitbucket.org/vmaric/qa_test_repo
   5d6a3e2..1a86fc7  master     -> qa_test/master
Auto-merging myfile.txt
CONFLICT (content): Merge conflict in myfile.txt
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\vpc\myproject>
```

Rukovanje konfliktima

- Git označava problematične delove markerima
- Konflikt se rešava ručnim uklanjanjem markera i sređivanjem problematičnog fajla





Credits

https://www.freepik.com/