

pieni git-opas

Tuisku

8. elokuuta 2019

Sisältö

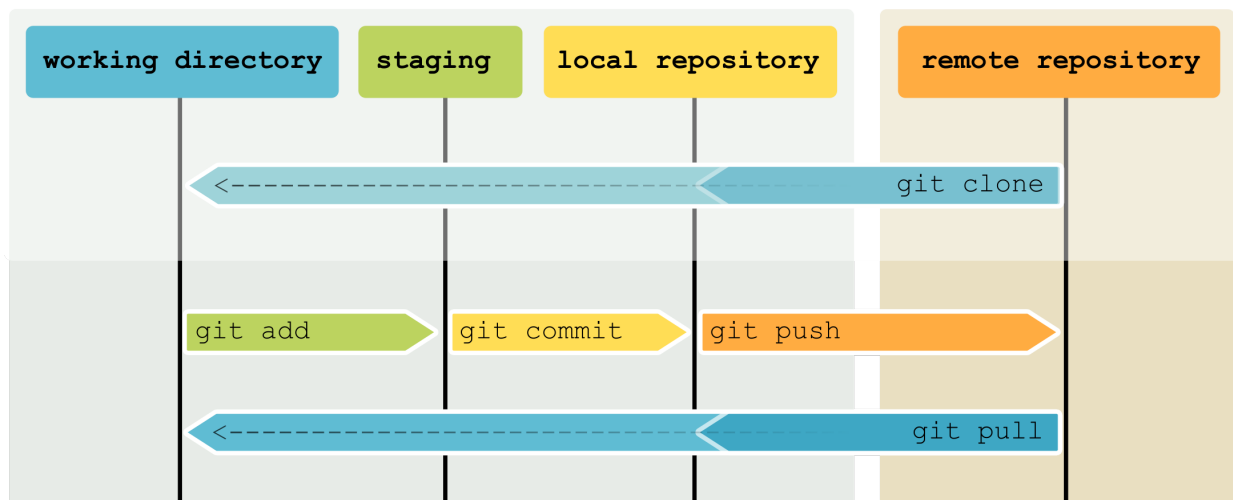
1	Uuden repositorion luominen (git init)	1
1.1	Uusi etärepoitorio GitLabissa	1
1.2	Uusi repositorio paikallisesti	1
2	Valmiin repositorion kloonaminen (git clone)	2
2.1	Valmistelut	2
2.2	Kloonaminen SSH-linkillä	2
2.3	Lopuksi	2
3	Repositorion uusimman version hakeminen (git pull)	3
4	Omien muokkausten lähettäminen etärepoitorioon (git push)	4
4.1	Valmistelut	4
4.2	Muutosten tallentaminen paikalliseen repositorioon (add, commit)	4
4.3	Viimeisimmän commitin lähettäminen etärepoitorioon (push)	4

Git & sen käyttö pähkinäkuoressa

Git on versionhallintajärjestelmä, jonka avulla voi pitää kirjaa projektin tiedostoista ja niiden muutoksista. Se mahdollistaa helpon paluun projektin aiempaan versioon, projektin työstämisen eri koneilta käsin ja useamman kuin yhden hengen yhtäaikaisen osallistumisen samaan projektiin.

Repositorio (repository, repo) eli tietovarasto on .git-kansio, joka sisältää tiedot kaikista projektin tiedostoista. Repositorio hankitaan joko luomalla se itse tai kloonamalla valmis **etärepositorio** esim. GitLabista, jolloin käyttäjä saa itselleen kaikki projektiin sisältyvät tiedostot ja versionhallintatiedot.

Etärepositorio (remote repository) on palvelimella, esim. GitLabissa sijaitseva repositorio, jota useampi muokkaa voi käsitellä kloonamalla siitä itselleen paikallisen repositorion, muokkaamalla tiedostoja omalla koneellaan, tallentamalla muutokset paikalliseen repositorioon ja lähettämällä paikalliset muutoksensa sen jälkeen etärepositorioon. Muut voivat tämän jälkeen hakea etärepositorioon lähetetyt muutokset omiin paikallisiin repositorioihinsa.

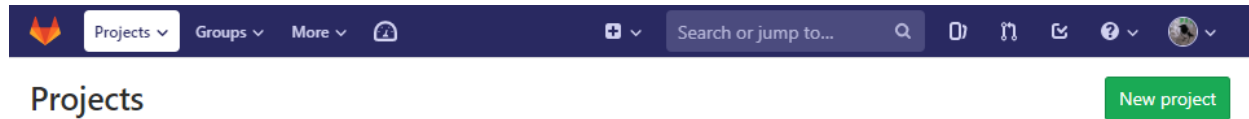


- **git clone** kloonataan etärepositorio (remote repository) omalle koneelle. Kloonaminen lataa myös kaikki repositorion seuraamat tiedostot paikalliseen työskentelyhakemistoon (working directory).
- **git add** listataan työskentelyhakemistossa tehdyt muutokset valmistelualueelle (stage).
- **git commit** vahvistetaan valmistelualueelle lisätyt muutokset pysyväksi muutokseksi (commit) paikalliseen repositorioon.
- **git push** lähetetään pysyvät muutokset paikallisesta repositoriosta etärepositorioon.
- **git pull** haetaan etärepositoriosta muutokset omaan repositorioon ja työskentelyhakemistoon.

1 Uuden repositorion luominen (git init)

1.1 Uusi etärepoitorio GitLabissa

Voit luoda uuden projektin sivulla <https://gitlab.utu.fi/projects/new> jonne pääset myös GitLabin etusivulta.



1. Anna projektillesi nimi ja valitse sijainti. Oletuksena se luodaan oman käyttäjätilesi alle, mutta jos kuulut GitLabissa johonkin ryhmään, voit luoda projektin myös ryhmän alle.
2. Valitse haluamasi näkyvyysasetus. Kurssien harjoitustöissä on sopivaa valita näkyvyydeksi "Private".
3. Paina "Create project" ja pääset uuden projektisi sivuille, jossa näet tyhjän repositoriosi.
4. Seuraava askel on äsken luodun repositorion kloonaaminen omalle koneellesi. Lue GitLabin tarjoamat ohjeet tai katso sivu "Valmiin repositorion kloonaaminen"

1.2 Uusi repositorio paikallisesti

1. Siirry hakemistoon, jonne haluat luoda repositoriosi Git Bashia (windows) tai terminaalia (linux, macOS) käyttäen.
 - `pwd` (present working directory) tulostaa hakemiston jossa olet.
 - `cd <hakemisto>` siirtää sinut haluamaasi hakemistoon (esim. `cd ~/projektit`).
2. `git init <uuden repositoriosi nimi>` luo hakemiston, jonka sisällä on uuden repositoriosi `.git`-tiedosto (esim. `git init harkkatyo`).
 - Pelkkä `git init` luo repositorion suoraan siihen hakemistoon jossa olet luomatta sille omaa hakemistoa. Käytä tätä jos olet jo luonut projektille sopivan hakemiston.
3. `cd <uuden repositoriosi nimi>` siirtää sinut repositoriosi hakemistoon (esim. `cd harkkatyo`).

Esimerkki

Luodaan projektit-kansioon uusi projekti "git-opas" repositorioineen.

- Käyttäjän antamat komennot näkyvät merkin \$ jälkeen ja kommentojen selitykset alkavat merkillä #

```
$ cd ~/projektit # Siirrytään kotihakemiston alikansioon "projektit"
$ git init git-opas # Luodaan projekti
Initialized empty Git repository in C:/Users/user/projektit/git-opas/.git/
$ cd git-opas # Siirrytään projektikansioon
```

2 Valmiin repositorion kloonaaminen (git clone)

2.1 Valmistelut

1. Siirry Git Bashia (windows) tai terminaalia (linux, macOS) käyttäen hakemistoon, jonne haluat kloonatun projektin repositorioineen.
 - `pwd` (present working directory) tulostaa hakemiston jossa olet
 - `cd <hakemisto>` (change directory) avaa halutun hakemiston (esim. `cd ~/projektit`)

2.2 Kloonaaminen SSH-linkillä

1. `git clone <git-tiedoston ssh-linkki>` kloonaa haluamasi repositorion (esim. `git clone gitlab@gitlab.utu.fi:opiskeliija/harkkatyo.git`).
 - Jos projektin nimessä on välilyöntejä, merkitään ne SSH-linkissä yhdysmerkillä (-).
 - GitLabista löydät projektin sivulta Clone-valikon, josta saat kopioitua kloonaukseen tarvittavan linkin.
2. syötä SSH-avainfraasisi Gitin tulostettua `Enter passphrase for key '/c/Users/user/.ssh/id_rsa':`

2.3 Lopuksi

`cd <kloonatun repositorion nimi>` siirtää sinut hakemistoon jossa juuri kloonattu repositorio on (esim. `cd harkkatyo`).

Esimerkki

Kloonataan GitLabista pienen git-oppaan projekti repositorioineen käyttäjän kansioon "projektit".

- Käyttäjän antamat komennot näkyvät merkin \$ jälkeen ja komentojen selitykset alkavat merkillä #.
- Gitin tulostamat rivit ovat harmaita. Huomiota vaativat välikommentit alkavat merkillä >.

```
$ cd ~/projektit # Siirrytään kansioon jonne projekti halutaan kloonata
$ git clone gitlab@gitlab.utu.fi:tuisku/pieni-git-opas.git # Kloonataan etärepositorio
Cloning into 'pieni-git-opas'...
Enter passphrase for key '/c/Users/user/.ssh/id_rsa':
> tässä kohtaa Git pyytää avainfraasia ja jatkaa vasta kun se on syötetty.
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 95 (delta 41), reused 92 (delta 39)
Receiving objects: 100% (95/95), 1.50 MiB | 11.32 MiB/s, done.
Resolving deltas: 100% (41/41), done.
$ cd git-opas # Siirrytään projektikansioon
```

3 Repositorion uusimman version hakeminen (git pull)

Jos etärepositoriota muokataan useammalta taholta, on vähän väliä päivitettävä oma paikallinen repositorio ajan tasalle hakemalla etärepositorion uusin versio. Tätä ei voi tehdä liian usein. Esimerkiksi omien muutosten lähettäminen muuttuneeseen etärepositorioon aiheuttaa konflikteja, sillä ei ole mahdollista automaattisesti tietää, mitkä muutokset ovat oikeellisempia.

1. Tarkista että olet oikeassa hakemistossa. Komento `pwd` tulostaa sijaintisi.
 - Voit tarvittaessa siirtyä hakemistoissa komennolla `cd <hakemistopolku>`
2. Hae etärepositorion uusin versio komennolla `git pull`.
 - `Already up to date` tarkoittaa, että hakemistosi oli jo ajan tasalla.
 - Jos hakemistosi ei ollut ajan tasalla, hakee Git muutokset repositoriosta ja päivittää paikallisen hakemiston.

Esimerkki

```
$ git pull # Vedetään repositorion uusin versio
Enter passphrase for key '/c/Users/user/.ssh/id_rsa':
> Syötetään avainfraasi
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From gitlab.utu.fi:tuisku/pieni-git-opas
   de1655c..93520b1  master       -> origin/master
Updating de1655c..93520b1
Fast-forward
 readme.md | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
> Repositoriossa oli yksi muuttunut tiedosto, jonka muutokset haettiin.
```

Kokeillaan uudestaan:

```
$ git pull # Vedetään repositorion uusin versio
Enter passphrase for key '/c/Users/user/.ssh/id_rsa':
> Syötetään avainfraasi
Already up to date
> Työhakemisto oli nyt ajan tasalla.
```

4 Omien muokkausten lähettäminen etärepositorioon (git push)

Kun projektin tiedostoja on muokattu ja halutaan lähettää muokatut tiedostot GitLabiin, tehdään se luomalla ensin omaan paikalliseen repositorioon Git Bashin (windows) tai terminaalin kautta ns. pysyvä muutos (commit), joka lähetetään (push) etärepositorioon.

4.1 Valmistelut

1. Tarkista että olet oikeassa hakemistossa. Komento `pwd` (present working directory) tulostaa sijaintisi.
 - Voit tarvittaessa siirtyä hakemistoissa komennolla `cd <hakemistopolku>`
2. `git status` näyttää viimeisimmän pysyvän muutoksen (commit) jälkeen tehdyt muutokset.
 - Muokkaamasi ja luomasi tiedostot näkyvät punaisella, jos niitä ei ole vielä lisätty valmistelualueelle (stage). Jos et näe tiedostojasi ja git sanoo `nothing to commit, working tree clean`, tarkista, että olet varmasti muistanut tallentaa tiedostoissa tekemäsi muutokset.
3. Jos etärepositoriota käsitellään useammalta koneelta, saattaa useampi henkilö olla tehnyt ja lähettänyt muokkauksia samanaikaisesti. Hae repositorion senhetkinen tila itsellesi komennolla `git pull` konfliktien välttämiseksi.

4.2 Muutosten tallentaminen paikalliseen repositorioon (add, commit)

1. `git add` lisää työhakemistosi tiedostot valmistelualueelle (stage).
 - `git add --all` tai `git add .` lisää kaikki tiedostot
 - Rajataksesi osan tiedostoista vakituisesti all-komennon ulkopuolelle katso sivu gitignore.
 - `git add <tiedostonimi>` lisää yksittäisen tiedoston
 - `git add <tiedosto1> <tiedosto2> <tiedosto3>` lisää useamman tiedoston kerralla
 - `git add <hakemisto>` lisää hakemiston

Voit käyttää tabulaattorinäppäintä täydentääksesi automaattisesti tiedostopolut ja -nimet kirjoittaessasi.

- o Nyt `git status` näyttää valmistelualueelle lisäämäsi tiedostot vihreänä.
2. `git commit -m "<viesti>"` rekisteröi valmistelualueen tiedostot pysyvään tilannekuvaan (commit).
 - Viestistä olisi hyvä käydä ilmi, mitä muutoksia olet tehnyt. Ääkkösiä on hyvä välttää.

4.3 Viimeisimmän commitin lähettäminen etärepositorioon (push)

1. Ennen lähettämistä on aina hyvä hakea etärepositorion mahdolliset muutokset komennolla `git pull`.
2. `git push` lähettää (push) paikalliseen repositorioosi rekisteröidyt muutoksesi (commit) etärepositorioon.
 - Jos kaikki sujui hyvin, Git tulostaa muutaman rivin tietoa, viimeisenä lähetettyjen tiedostojen määrän.

Esimerkki

Tiedostoa `pieni_git_opas.pdf` on muokattu. Uusi versio halutaan lähettää etärepositorioon.

```
$ cd ~/projektit/pieni-git-opas # Siirrytään projektin kansioon
$ git status # Tarkistetaan näkyykö muutettu tiedosto
$ git add pieni_git_opas.pdf # Kaikki ok, lisätään tiedosto valmistelualueelle
$ git pull # Vedetään repositorion uusin versio
$ git commit -m "lisetty kappale gitin push-ohjeesta" # Tehdään pysyvä muutos
$ git push # Pusketaan muutos repositorioon
```