

JOT2013 Demo1-ohje

Ville Isomöttönen, Jonne Itkonen

10. syyskuuta 2013

Varmista, että tunnuksesi on aktivoitu Unix-etäkäyttökoneet

Näet IT-palvelujen käyttäjätunnuksen tietojen päivitys ja salasanan vaihto -palvelusta (salasana.jyu.fi) onko sinulla aktivoituna UNIX-etäkoneiden käyttöoikeus. Tästä tiedotimmekin aiemmin sähköpostitse.

Varmista, että U-levy on kytketty Windows-profiiliisi

Nykyisellään U-levyn (kotihakemistosi) pitäisi olla automaattisesti kytketty Windows-profiiliin ja näkyä siis Windowsin resurssien hallinnassa. Jos jostain syystä näin ei ole, seuraa ohjetta osoitteessa <https://www.jyu.fi/itp/ohjeet/how-to-nain-tehdään/verkkolevyjen-kayttaminen-kotikoneelta-ka> ja mounttaa kotihakemistosi Windows-profiiliisi (tämä helpottaa demoissa harjoittelua).

SSH-avainparin luonti

Git Bash -ohjelmalla

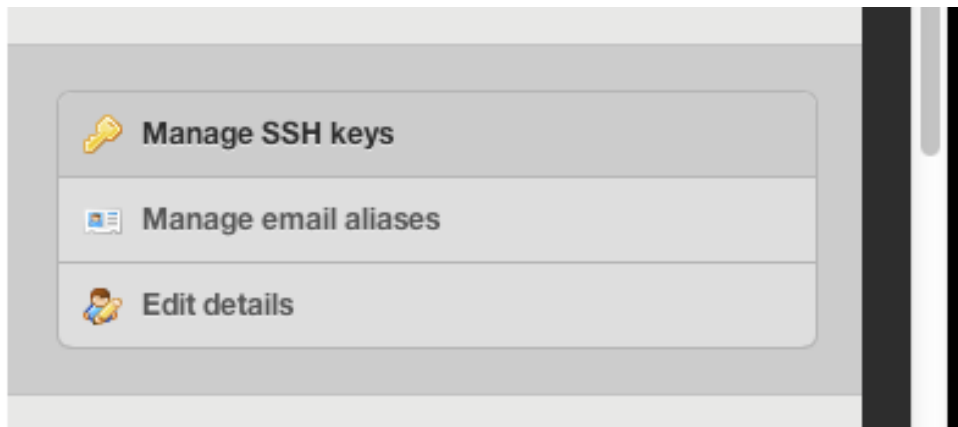
Git-versionhallintaohjelmiston asennuksen myötä mikroluokkien Windows-koneissa on asennettuna Git Bash -ohjelma, ts. komentorivitulkki, jota voit käyttää Git-komentojen antamiseen, mutta myös SSH-avainparin luontiin. Avainparia luodessasi toimit siis nyt vastavasti kuin Linuxin ja Mac OSX:n komentoriviympäristöissä.

Huomioi kun luet seuraavaa, että U:-asema Windowsissa yliopistolla on täsmälleen sama kuin jalavan ja muiden etäkäyttökoneiden kotihakemistosi!

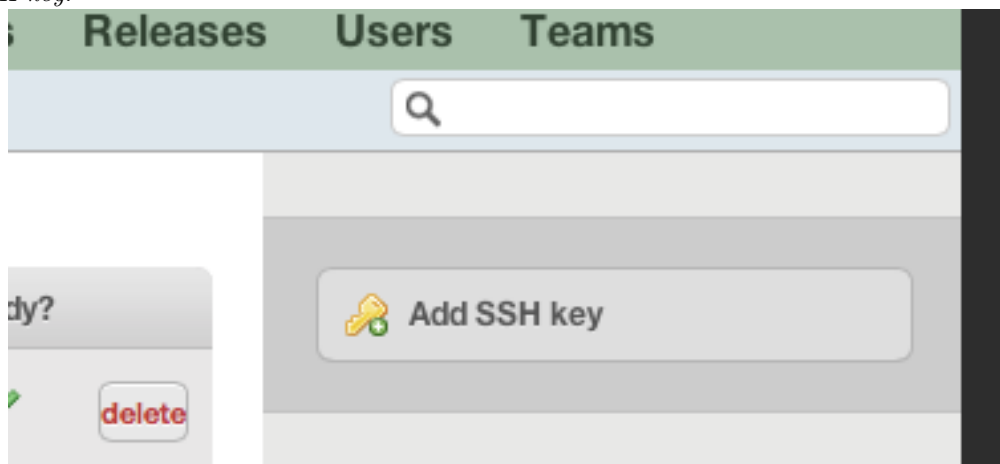
Avaa Git Bash -ohjelma ja luo SSH-avainpari komentoriviltä komennolla `ssh-keygen -t rsa`. Ohjelma `ssh-keygen` ehdottaa avaimille oletusnimiä, jotka kannattaa hyväksyä, jollet ole aiemmin tehnyt avaimia jalavaan (ts. yliopiston UNIX-etäkoneille). Jos olet aiemmin luonut avaimet jalavaan (tai vastaavaan etäkäyttökoneeseen), tämä avainpari-vaihe on turha.

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/nashome1/uid/.ssh/id_rsa): _
```

Kun `ssh-keygen` ohjelma kysyy tunnuslauseita (*passphrase*) paina vain enteriä mennäksesi eteenpäin. SSH-keygen luo jalavan `.ssh` hakemistoon avain parin, ts. kaksi tiedostoa: `id_rsa` ja `id_rsa.pub`. Näistä jälkimmäinen tulee siirtää yliopiston yousource-koneelle, jota kurssilla käytetään versiohallintaan. Tämä tehdään menemällä osoitteeseen <https://yousource.it.jyu.fi/> ja kirjautumalla omilla käyttäjätunnuksillasi sisään YouSource-palveluun.



Valitse oikealta *Manage SSH-keys*, jonka jälkeen avautuvalta sivulta edelleen oikealta *Add SSH key*.



Kopioi luomasi avain parin julkinen avain (tiedoston `id_rsa.pub` sisältö) sivun lomakkeeseen ja paina *Save*.

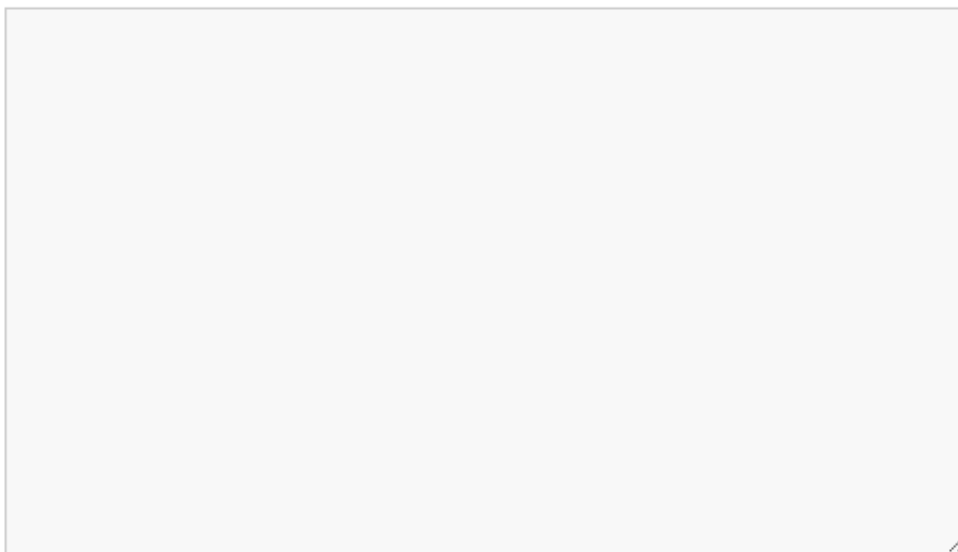
Add a new public SSH key

Your public key

It is generally located in `~/.ssh/id_rsa.pub` or `~/.ssh/id_dsa.pub`. If you want to use multiple keys you will have to add each of them separately.

The key should be in the format of:

```
ssh-algorithm base64-content you@somehost
```



Save

Kun käytät tällä kurssilla versiohallintaa, ts. siirrät tiedostoja jalava-koneen ja yousource-koneen välillä, luomasi avainpari mahdollistaa tiedostojen siirron ilman jatkuvaa autentikointia. Huomaa siis, että yksityinen avain (tiedosto `id_rsa`) sijaitsee jalava-koneella ja sen julkinen pari (tiedosto `id_rsa.pub`) yousource-koneella. SSH-yhteys luodaan avainparin avulla automaattisesti tietojen siirron ajaksi.

Pieni ohje UNIX-komentoihin löytyy osoitteesta <http://www.mit.jyu.fi/opetus/kurssit/jot/2012/unix.html>.

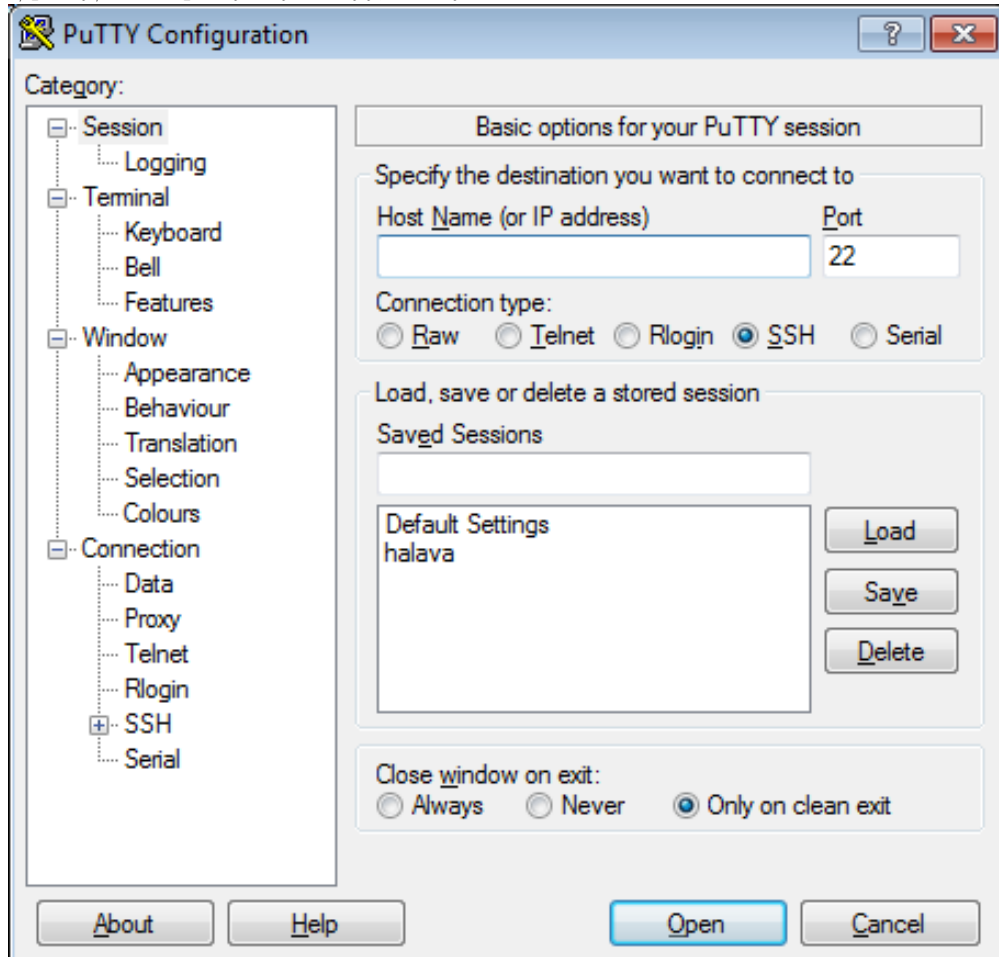
Huomioi, että poistut hakemistosta `.ssh` ennen kuin jatkat harjoituksia. Älä koskaan työskentele hakemistossa `.ssh` tai myöhemmin esiteltävässä `.git`-hakemistossa, ellet todella tiedä, mitä olet tekemässä.

Mikroluokat Windowsilla, jos Git Bash ei löydy

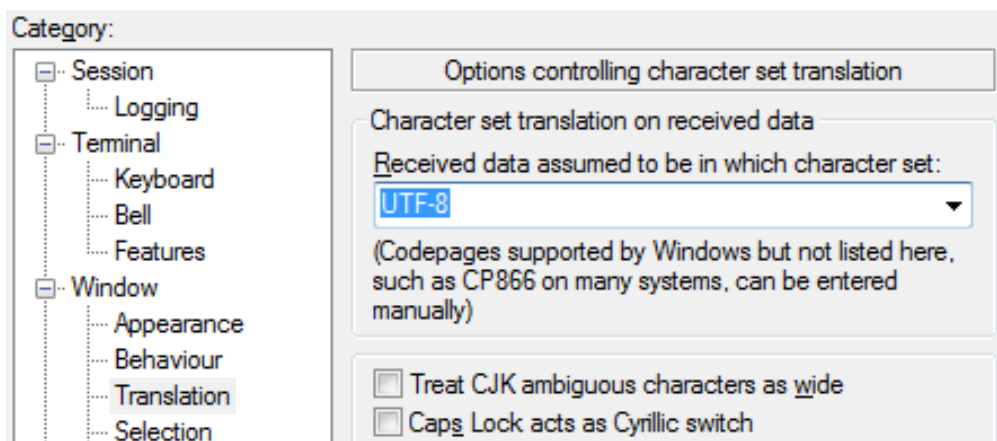
Jos mikroluokkaan ei ole asennettu Git Bash -ohjelmaa, voit muodostaa etäyhteyden jalava-koneeseen Putty-ohjelmalla. (Etäkäyttökoneiden kotihakemisto on sama kuin Windows-profiiliisi)

mapattu U-asema.)

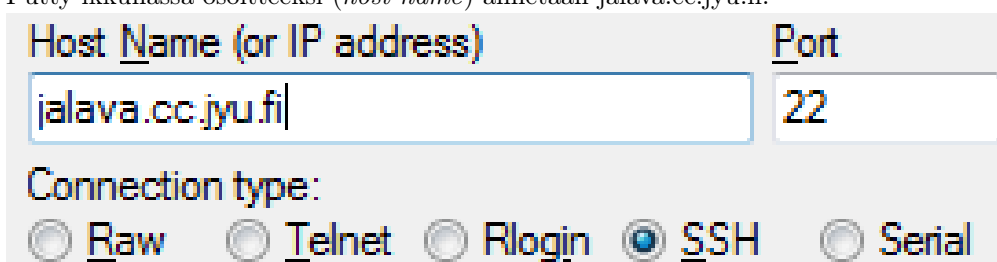
Windows-koneella avaa Putty-ohjelma, ja ota sillä yhteys jalava-etäkoneeseen. Jos kotikoneellasi ei ole Puttya, se löytyy osoitteesta [http://www.chiark.greenend.org.uk/~sim sgtatham/putty/](http://www.chiark.greenend.org.uk/~sim%20sgtatham/putty/). Kun putty käynnistyy, se näyttää seuraavanlaisen ikkunan.



Aseta ensin käytettäväksi merkistöksi UTF-8 valitsemalla *Category*-laatikosta vasemmalta *Window* → *Traslation*, ja syöttämällä kohtaan *Character set translation...* teksti UTF-8, joka löytyy myös ponnahdusvalikosta.



PuTTY-ikkunassa osoitteeksi (*host name*) annetaan `jalava.cc.jyu.fi`.



Hyväksy tulostuva turvallisuuskehoite. Syötä tämän jälkeen tunnuksesi ja salasanasasi, ja olet muodostanut yhteyden jalava-koneeseen. SSH-avainparin voit tämän jälkeen tehdä samoilla komennoilla kuin yllä Git Bash -komentorivitulkin tapauksessa.

*** Muut kuin yliopiston Windows-asennukset #Jos asennat Gitin kotiin, unohda myös `putty`. OpenSSH on parempi, ja #jos et sitä tarvitse, tulee Gitin Windows-version mukana jo #tarvittavat ohjelmistot (Git Bash).

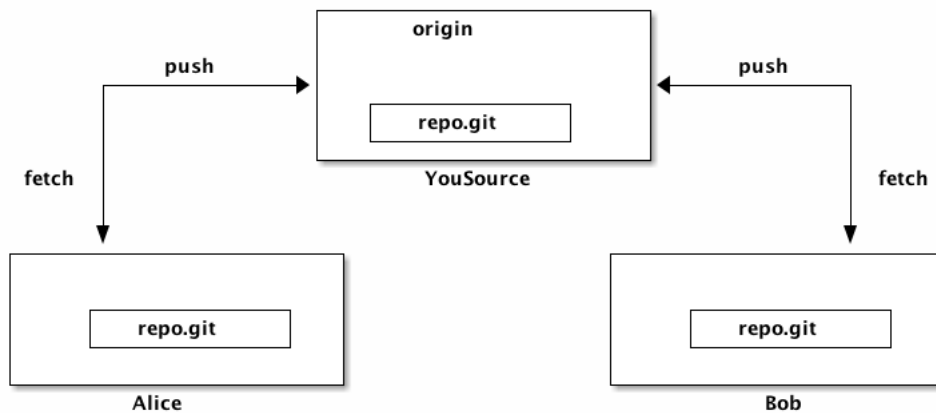
*** Windows-koneet Soviet- ja Finland-mikroluokissa (8.2.2013 tilanne): #Käytä Git Bash- ja/tai Git GUI-ohjelmaa. Unohda alla oleva `putty`-sekoilu, #ja toimi Git Bashissa kuten Unix-, Linux- tai OSX-tietokoneessa.

Linux tai Mac OSX

Voit tehdä avainparit komentoriviltä vastaavasti kuin yllä Git Bash -ohjelmalla. Avaat vain Terminal-ohjelman, jossa yleensä oletuksena on käytössä Bash-komentotulkki. Voit tehdä kaiken jalavassa tehtävän omassa tietokoneessasi. Tarvittavat ohjelmat `git` ja `ssh` löytyvät usein jo asennettuina, ja jos eivät, ne asentaa helposti (Linux Redhat ja vastaavat: `rpm`, Linux Debian / Ubuntu ja vastaavat: `apt-get` (asentaminen), `apt-cache` (haku), ja `synaptic` (GUI), OS X: hae git sen kotisivulta osoitteesta <http://git-scm.com>), tai käytä Homebrewtä gitin asentamiseen (Applen XCode-kehitystyökalu oltava asennettuna).

Git-versiohallinta

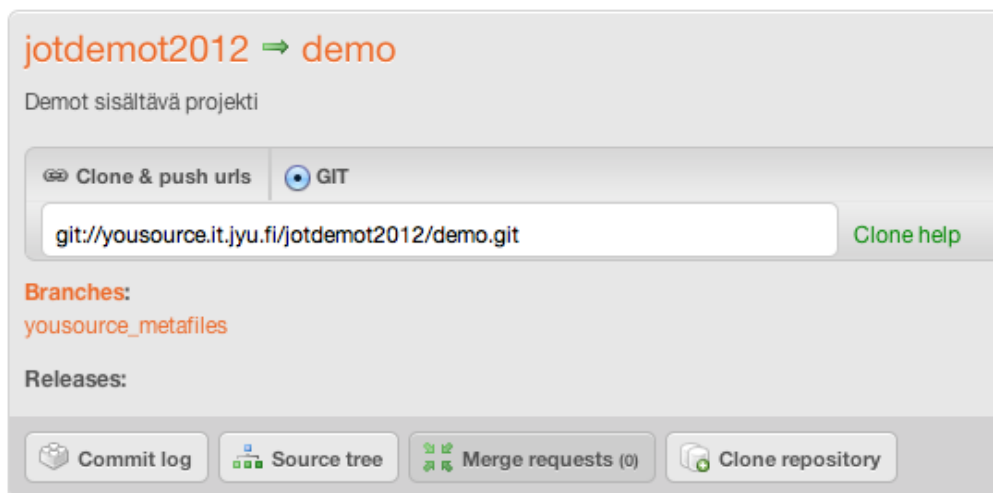
Idea on yksinkertaisuudessaan näkyvissä seuraavassa kuvassa:



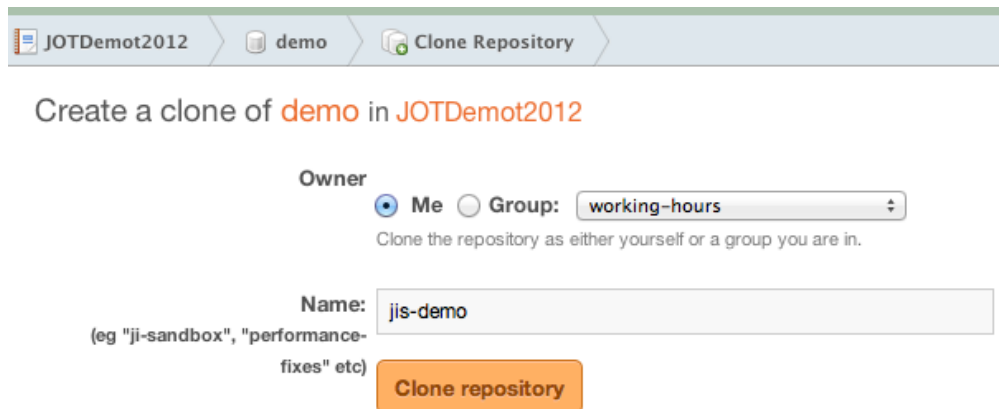
Alice ja Bob työskentelevät saman projektin parissa. Molemmilla on omassa lokaalissa tietokoneessaan oma git-versiohallinnalla hallintoitu varasto, johon he tekevät omat muutoksensa. Lisäksi heillä on YouSourceen yhteinen varasto, jonka välityksellä he jakavat muutoksensa toisilleen. Demoissa tarkoituksena on luoda ryhmälle yksi jaettu varasto, ja jokaiselle ryhmän jäsenelle oma varastonsa.

Git-versionhallinnan aloittaminen

Demoissa harjoitellaan versionhallintaa suoraan komentoriviltä. Ensin on kuitenkin luotava git-varasto YouSourceen. Avatkaa JOTDemot2012-projekti, joka löytyy YouSourcesta osoitteesta <https://yousource.it.jyu.fi/jotdemot2012>. Projektista löytyy git-varasto nimeltä **demo**. Kloonatkaa tämä varasto itsellenne napista **Clone repository**.



YouSourcen seuraava näkymä kysyy kloonatulle varastolle nimeä ja muita asioita, antaa nimeksi jotain omaa tai hyväksyykää oletus.



Kloonausta ei tarvitse tehdä kuin yhden ryhmästä, mutta kaikki voivat toki harjoitella tätä. Kun varasto on kloonattu näyttää YouSource kloonatun varaston sivua. Jatkossa kaikki ryhmän jäsenet käyttävät tätä kloonattua varastoa. Ensimmäinen kloonin tehnyt antaa nyt muille ryhmäläisille oikeuden tähän varastoon YouSourcen asetuksista, jotka löytyvät ikkunan oikeasta laidasta löytyvän napin **Manage collaborators** takaa. Tämä tosin onnistuu vain, jos lisättävä ryhmän jäsen on jo aiemmin kertaalleen kirjautunut YouSourceen. Sen ylälaidasta, kloonivaraston nimen alta löytyy git-alkuinen osoite, jota tarvitaan, kun varasto kloonataan vielä toistamiseen kunkin ryhmäläisen omaan tietokoneeseen. Tämä kloonaus tapahtuu antamalla komentoriviltä komento `git clone URL`, missä URL on yllä YouSourceessa näytetty URL. Kun jälkimmäinen kloonaus on valmis, on harjoitteluhakemisto koneessasi valmis.

Tee seuraavaksi jokin pieni tekstitiedosto, kuten `test.txt` harjoitteluhakemistoosi. Kirjoita tiedostoon vapaamuotoista tekstiä ja tallenna. Palaa jalavan komentoriville ja anna komento `git add test.txt`. Tällä komennolla kerrot, että haluat ottaa tiedoston `test.txt` mukaan versioitavaksi. Anna tämän jälkeen komento `git commit`, joka siirtää tiedostosi tiedot (pikemminkin version) lokaaliin git-varastoosi. Itse asiassa `git commit` siirtää versiotiedot kaikista niistä tiedostoista, joille olet sanonut `git add` osaksi lokaalia git-varastoasi.

Jos olette muuttaneet vain yhtä tiedostoa, tai haluatte tallentaa kaikki muutetut tiedostot lokaaliin versiohallintaan, voitte käyttää komentoa `git commit -a`. Uudemmissa git-versioissa toimii myös muoto `git commit tiedostot`, jolla voitte lisätä ja tallentaa versiohallintaan samalla komentorivillä useita tiedostoja kerralla. Tuo siis vastaa `git add` ja `git commit` komentojen antamista jokaiselle mainitulle tiedostolle.

Git Commit ja kommentin kirjoittaminen

Kun ajat jalavassa (tai muissa Unix-koneissa) komennon `git commit ilman` tarkennetta `-m`, se käynnistää editorin, jolla kommentti muutoksista voidaan kirjoittaa. Ohjeet kirjoittamisesta ovat itse kommentissa kommentoituna (heh!), mutta editorista ei juuri sanota mitään. Oletuksen editorin on usein `vim` tai `vi`, joka poikkeaa hieman totutusta. `Vim` on tilallinen editor, joten päästäksesi kirjoittamaan sinun on siirryttävä muokkaustilaan painamalla `i` (*insert*). Kun olet valmis, tulee sinun siirtyä muokkaustilasta komentotilaan painamalla `<ESC>`, eli sitä näppäimistö vasemmasta yläkulmasta löytyvää `esc`-painiketta. Tämän jälkeen tallennus ja poistuminen tapahtuu joko näppäilyllä `:wq<RET>`, missä `<RET>` tarkoittaa rivinvaihtoa (*enter*, *return*), tai lyhyemmin `:x<RET>`. Vaikeata on osua `esc`-painikkeeseen, mutta pienellä harjoituksella se onnistuu huomaamatta.

Jos haluat vaihtaa oletuseditorin, tapahtuu se muokkaamalla kuoren (*shell*) aloitusskriptissä ympäristömuuttuja `EDITOR` haluamaksesi, esim. `bash`-kuorella

```
export EDITOR=emacs
```

Jos et, tai tuntuu liian nörtiltä, unohda tämä :-)

Emacsilla tiedoston tallentaminen tapahtuu komennolla `C-x C-s`, eli `ctrl`-nappi pohjaan ja napautat `x`, ihan niinkuin Windowsin `Cut`-komennossa, ja `ctrl`-nappi pohjassa pitäen napautat vielä `s`. Poistuminen tapahtuu vastaavasti näppäilyllä `C-x C-c`. Hankalaa on nyt pitää `ctrl`-nappi pohjassa. Jos haluatte, aktivoituvat valikot painamalla `F10`-näppäintä, ja nyt komentojen antaminen on yksittäisten, korostettujen kirjainten painamista.

Paniikkikomento vimissä on `<ESC><ESC><ESC>`, emacsissa `C-g`.

Jo nyt on hyvä varoittaa kahdesta asiasta: demo-ohjaajien editorimieltymykset ovat toistensa vastaiset, ja ainoa väärä valinta on jokin muu editori kuin Emacs tai vim ;-)

Lokaalista koneesta versiot etäkoneelle

Jotta versiohallinta-ohjelma voisi helpottaa ryhmässä tapahtuvaa työskentelyä, on tekemäsi muutokset saatettava muiden projektiin osallistuvien saataville.

Kirjaudu jälleen `yousource`-palveluun osoitteessa <https://yousource.it.jyu.fi/> ja siirry tunnuksesi alla näkyvään, äsken luomaasi varastoon (muista nimi, jonka annoit varastollemme). Valitse oikealta *Edit repository*. Tee luomastasi varastosta yksityinen (*private*) rastittamalla kohta *Private repo* ja tallenna lomake alhaalta (nappi *Edit repository*). Tämän jälkeen valitse oikealta kohta *Manage collaborators*, jossa voit lisätä harjoittelua varten vieruskaverin tunnukselle oikeudet luomaasi varastoon.

Vaihtakaa rooleja siten, että vieruskaveri haluaa nyt saada sinun luomasi ja `yousource`en siirtämäsi varaston käyttöönsä. Vieruskaverin tulee ottaa Puttylla etäyhteys jalava-koneeseen. Toisin kuin äsken, hänen *ei tarvitse* itse luoda uutta lokaalia git-varastoa, vaan tämä voidaan kloonata `yousource`esta komennolla: `git clone OSOITE`, missä `OSOITE` on tunnuksesi alla näkyvän aiemmin kloonamasi git-varaston osoite.

Seuraavassa kuvassa on koottuna kaikki gitin peruskomennot.

http://upload.wikimedia.org/wikipedia/commons/thumb/4/44/Git_data_flow_simplified.svg/200px-Git_data_flow_simplified.svg.png (Lähde: wikipedia)

Harjoittelua

Varaston kloonannut lisää testitiedostoon uutta tekstiä, tallentaa sen ja ajaa sen jälkeen seuraavat versiohallintakomennot:

- `git add test.txt`
- `git commit test.txt` (Anna tässä aukeavaan editoriin muutoksiasi kuvaava viesti.)
- `git push origin master` (Nyt siirrät oman varastosi tilan etäkoneelle, ts. `YouSource`en.)

Vaihtakaa jälleen rooleja. Nyt vieruskaverisi saa muutokset itselleen seuraavasti:

- `git fetch -v origin` (`Fetch` noutaa etäkoneen varaston tilan omalle koneellesi. Optio `-v`, *verbose*, tulostaa runsaamman kuvauksen komennon aiheuttamista operaatioista)


```
$ git fetch -v origin
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 3), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
From yousource.it.jyu.fi:opetus-ji/jot2012
   c39f3f0..f9258c6  master    -> origin/master
   = [up to date]      yousource_metafiles -> origin/yousource_metafiles
$ _
```

- Huomaa, että etäkoneelta tuotu tila pitää vielä yhdistää oman varaston tilaan, jotta muiden tekemät muutokset ovat käytössäsi. Tämän tekemiseksi anna komento `git merge origin/master`. Tällöin muutokset tulevat osaksi omaa varastoasi.
- Jotta molemmat saavat harjoittelua, tehkää vuoronperään muutoksia testitiedostoon omilla Putty-yhteyksillänne ja toistakaa opeteltuja komentoja `add`, `commit`, `push`, `fetch` ja `merge`.
- Kun törmäätte virheeseen, nostakaa käsi ylös. Olette todennäköisesti saaneet aikaan konfliktin, eli olette muokanneet samaa kohtaa tiedostossa eri tavoin. Tämä ratkaistaan muokkaamalla tiedosto sisältämään vain halutun tekstin. Molemmat versiot ovat esillä tiedostossa tunnisteella erotettuina, joten konfliktin löytäminen tiedostosta ja korjauksen tekeminen on helppoa. Konfliktin huomautat yleensä, kun yrität yhdistää toista varastoa omaan varastoosi esimerkiksi `git merge` komennon yhteydessä.

```
$ git fetch -v origin
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 3), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
From yousource.it.jyu.fi:opetus-ji/jot2012
   013fel..60f1893  master    -> origin/master
   = [up to date]      yousource_metafiles -> origin/yousource_metafiles
$ git merge origin/master
Auto-mergin demo-ohjeet/demo1-ohje.org
CONFLICT (content): Merge conflict in demo-ohjeet/demo1-ohje.org
Automatic merge failed; fix conflicts and the commit the result.
$ _
```

Alla esimerkki tiedostosta, johon on ilmestynyt konflikti:

- Kun törmäätte virheeseen, nostakaa käsi ylös. Olette todennäköisesti saaneet aikaan konfliktin, eli olette muokanneet samaa kohtaa tiedostossa eri tavoin. Tämä ratkaistaan muokkaamalla tiedosto sisältämään vain halutun tekstin. Molemmat versiot ovat esillä tiedostossa tunnisteella erotettuina, joten konfliktin löytäminen tiedostosta ja korjauksen tekeminen on helppoa. Alla on esimerkki konfliktista tiedostossa:

```
<<<<<<< HEAD
```

```
[[file:git-konflikti.png]]
=====
Hei mites sellainen tähän saadaan
>>>>>> origin/master
```

Yhtäsuuruusmerkit erottavat kulmasulkeiden välissä olevat kaksi versiota toisistaan. Valitse näistä oikea, tai luo uusi yhdiste. Tämän jälkeen aja git-komennot:

- `git add test.txt`
- `git commit` (Anna taas kuvaava viesti muutoksesta.)

Konflikti on nyt korjattu, ja voit siirtää varastosi nykytilan talteen etävarastoon komennolla `git push origin`.

Avatkaa nyt <http://git-scm.com/book>, ja tutustukaa ainakin seuraaviin komentoihin:

- `git status`,
- `git pull`,
- `git log`,
- `git diff`,
- `git branch`,
- `git checkout` ja
- `git tag`.