Git

1.1. Talks

2015-05-14 Xavier de Pedro - Enhance Team Work with DVCS - Git and Bazaar
http://ueb.vhir.org/SeminarDVCS[1]

1.2. Cheatsheets

1.2.1. Git



Github Git Cheat Sheet

See also:
http://www.r-bloggers.com/rstudio-and-github/[2]

1.3. My Online Repositories

1.3.1. Gitlab

https://gitlab.com/xavidp/[3]

Gitlab allows public and private repos, plus self-hosting gitlab in your own server if you wish.

1.3.2. Github

https://github.com/xavidp/[4]

Steps to add to Github.com some git-controlled code in your desktop.
First create a repository at Github.com with the same name of the local repository that you have in your local computer.
Then add the ssh public key from your computer at Github.com, if you haven't done so already.

Then these commands:



```
cd ~/scripts/TikiProfilesTester
git remote add origin https://github.com/xavidp/TikiProfilesTester.git
git remote set-url origin https://github.com/xavidp/TikiProfilesTester.git
git config remote.origin.url git@github.com:xavidp/TikiProfilesTester.git
git commit -m "Added a minor change" # optional, only if requried
git pull https://github.com/xavidp/TikiProfilesTester.git
git push --set-upstream origin master
```

```
git push -u
sed -i -e "s/url = https:\/\/github.com\/xavidp\//url = git@github.com:xavidp\//g"
.git/config # this seems needed from work at VHIR, otherwise it doesn't succeed
pushing through Firewall/Proxy.
```

1.4. Identify username and email



```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

1.5. Rename some author name and email from all git history from a repo

Create a new bash file "git_rewrite_author.sh" with these contents:



```
#!/bin/sh

git filter-branch --env-filter '
OLD_EMAIL="myoldemail@oldexample.com"
CORRECT_NAME="My real name"
CORRECT_EMAIL="myrealemail@newexample.com"
if [ "$GIT_COMMITTER_EMAIL" = "$OLD_EMAIL" ]
then
    export GIT_COMMITTER_NAME="$CORRECT_NAME"
    export GIT_COMMITTER_EMAIL="$CORRECT_EMAIL"
fi
if [ "$GIT_AUTHOR_EMAIL" = "$OLD_EMAIL" ]
then
    export GIT_AUTHOR_NAME="$CORRECT_NAME"
    export GIT_AUTHOR_EMAIL="$CORRECT_EMAIL"
fi
' --tag-name-filter cat -- --branches --tags
```

Then form a terminal at the git project root (the repository root) run this command:



```
bash ./git_rewrite_author.sh
```

If you are in a Windows computer where you don't have access to a terminal, you can run git commands from a bash terminal through the git-for-windows version of Git, run from a USB disk. https://gitforwindows.org[5]

1.6. Add your ssh public key to your gitlab/github accounts

Get a copy of the public ssh keys with commands like

```
# If you have rsa keys
cat ~/.ssh/id_rsa.pub
# If you have dsa keys
cat ~/.ssh/id_dsa.pub
```

And copy&paste in the right place of each website.

1.7. Get code from github

```
# Get the url ending in .git from the repo you want to clone
git clone https://github.com/xavidp/bashscripts.git
git clone git@github.com:uebueb/bpmmirna.git
# edit your files
nano myfile.sh
# Commit your changes adding your editions to the new revision and providing a
commit message
git commit -a -m "My changes info"
# Send your changes to the remote repo where you cloned your code from
git push
```

1.8. Publish on github your local git-controlled code

Steps to add to Github.com some git-controlled code in your desktop.
First create a repository at Github.com with the same name of the local repository that you have in your local computer.
Then add the ssh public key from your computer at Github.com, if you haven't done so already.

Then these commands:

```
cd ~/scripts/TikiProfilesTester
git remote add origin https://github.com/xavidp/TikiProfilesTester.git
git remote set-url origin https://github.com/xavidp/TikiProfilesTester.git
git config remote.origin.url git@github.com:xavidp/TikiProfilesTester.git
#git commit -m "Added a minor change" # optional, only if requried
git config --global push.default matching
git config --global user.name "xavidp"
sed -i -e "s/url = https:\/\/github.com\/xavidp\//url = git@github.com:xavidp\//g"
.git/config # this seems needed from work at VHIR, otherwise it doesn't succeed
pushing through Firewall/Proxy.
git pull https://github.com/xavidp/TikiProfilesTester.git
git push -u -f
git reset HEAD *
```

1.9. Bash script to fix these things for your user and repo

See:
https://github.com/xavidp/bashscripts/blob/master/fixgitueb.sh[6]

1.10. Convert between Git and Bazaar

http://ueb.vhir.org/Bazaar#Convert_between_Git_and_Bazaar_and_others_[7]

1.11. Use your own open-source github-like webapp for your code

See:

- Gitlab: https://gitlab.com[8]
  Or
- Phabricator: http://phabricator.org[9]

1.12. FLOSS Git GUIs

git-cola

```
sudo apt-get install git-cola
```

See: http://git-cola.github.io/downloads.html[10]

git-gui

```
sudo apt-get install git-gui
```

See: https://www.maketecheasier.com/6-useful-graphical-git-client-for-linux/[11]

GitKraken

FLOSS (?) Crossplatform Git GUI
See: https://www.gitkraken.com/[12]

1.13. Git AjBCN (with portable apps)

At Aj.BCN you can install Git on Windows in a USB disk with portable apps:
https://portableapps.com[13]

And install there also GitforWindows:
https://gitforwindows.org[14]

1.13.1. GitForWindows

Git for Windows focuses on offering a lightweight, native set of tools that bring the full feature set of the Git SCM to Windows while providing appropriate user interfaces for experienced Git users and novices alike.

Git BASH

Git for Windows provides a BASH emulation used to run Git from the command line. *NIX users should feel right at home, as the BASH emulation behaves just like the "git" command in LINUX and UNIX environments.

Git GUI

As Windows users commonly expect graphical user interfaces, Git for Windows also provides the Git GUI, a powerful alternative to Git BASH, offering a graphical version of just about every Git command line function, as well as comprehensive visual diff tools.

Shell Integration

Simply right-click on a folder in Windows Explorer to access the BASH or GUI.

1.13.2. RStudio + Git

You can use Git from RStudio portable version also.

1.13.3. Rewrite author and email from all your commits locally

The computer from AjBCN uses by default your windows name (in my case, my Surnames in uppercase) and the InternalUserId@bcn.cat as your email, which is likely to not match at all with your github username and email.
Therefore, you might want to fix your user name and email so that they can show up in your github contributions graphs and stats, etc.

I fixed locally this information running the script below in a Git BASH window openend from the root folder of the git-controlled project in your computer (go with Windows explorer to that folder, right click and select `Git Bash Here`).
Make a file with this contents in that folder:

Contents of gitrewriteauthor.sh

```
#!/bin/sh

git filter-branch --env-filter '
OLD_EMAIL="myoldemail@oldexample.com"
CORRECT_NAME="Foo Bar"
CORRECT_EMAIL="mynewemail@oldexample.com"
if [ "$GIT_COMMITTER_EMAIL" = "$OLD_EMAIL" ]
then
    export GIT_COMMITTER_NAME="$CORRECT_NAME"
    export GIT_COMMITTER_EMAIL="$CORRECT_EMAIL"
fi
if [ "$GIT_AUTHOR_EMAIL" = "$OLD_EMAIL" ]
then
    export GIT_AUTHOR_NAME="$CORRECT_NAME"
    export GIT_AUTHOR_EMAIL="$CORRECT_EMAIL"
fi
' --tag-name-filter cat -- --branches --tags
```

Then run this gitrewriteauthor.sh bash script there

1.13.4. Generate your ssh key pair (if needed still)

Open a Git Bash terminal window and run the command to generate your new ssh Key pair.
Follow the instructions in github to generate your key pair.
See:

- https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#generating-a-new-ssh-key[15]

Copy the contents of your local public key into the corresponding place of your profile account in github.com
See:

- https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/[16]

1.13.5. Commit from Local to Github

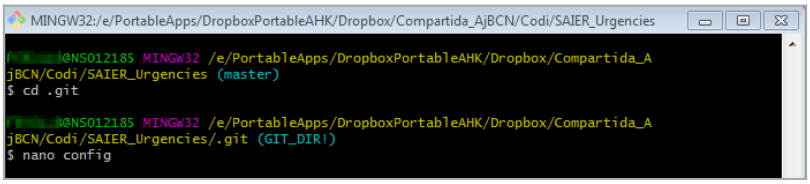
Code to commit from local to Github, after you create the branch with equivalent name in github as usual, etc (see above for further generic instructions).

```
cd ~/yourpath/SAIER_Urgencies
git remote add origin https://github.com/xavidp/SAIER_Urgencies.git
git remote set-url origin https://github.com/xavidp/SAIER_Urgencies.git
git config remote.origin.url git@github.com:xavidp/SAIER_Urgencies.git
#git commit -m "Added a minor change" # optional, only if requried
git config --global push.default matching
git config --global user.name "xavidp"
sed -i -e "s/url = https:\/\/github.com\/xavidp\//url = git@github.com:xavidp\//g"
.git/config # this seems needed from work at VHIR, otherwise it doesn't succeed
pushing through Firewall/Proxy.
git pull https://github.com/xavidp/SAIER_Urgencies.git
git push -u -f
git reset HEAD *
```

1.13.6. Fix permanently your user and email for this project

You can add some new parameters to the `.git/config` file under your project folder. This will stick to these new user and email values for this specific folder, without touching the other git projects in your computer.
Therefore, open a new Git Bash terminal in your project folder, and then move the .git subfolder and open the **config** file with the `nano` editor:

```
cd .git
nano config
```

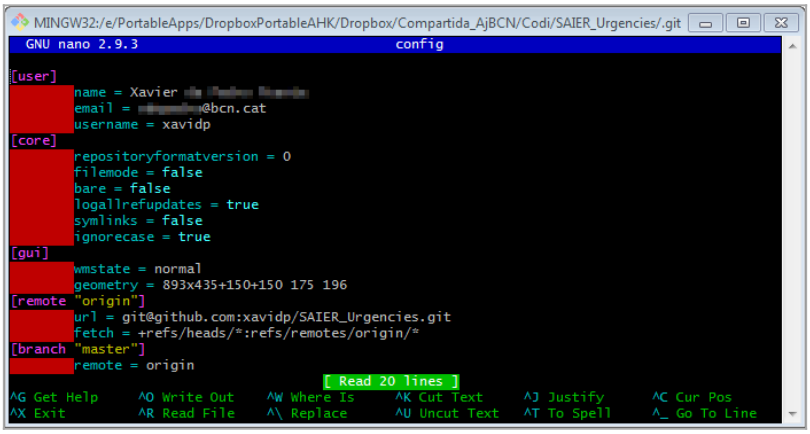


Click to expand

Then add this section at the top of the file:

```
[user]
        name = YourName And Surnames
        email = youremail@bcn.cat
        username = yourusername
```

You can copy from here or elsewhere and paste through right-cliking on the terminal window to select **Paste** (Shit + Ins, in `Git Bash` terminal windows), so that after that it looks like:



Click to expand

and press `Ctr + x` later on to save the file and exit the nano editor in the same go.

From now on, when you commit changes from RStudio (or other git based programs and workflows), you will get this name, email and usersname linked to those commits.

1.13.7. Push through Git BASH

Pushing from RStudio doesn't seem to succeed getting the right ssh keys to push your changes automagically to github. Therefore, you'd better open a Git Bash terminal window in your project root folder, and push from there with a standard command to do so:

```
git push
```

1.14. Gitlab prune big-files from project

Reference:

- https://docs.gitlab.com/ee/user/project/repository/reducing_the_repo_size_using_git.html[17]
- https://rtyley.github.io/bfg-repo-cleaner/[18]

```
xavi@tricholoma:~/tmp$ git clone --mirror
https://gitlab.com/radup/curs-r-introduccio.git
S'està clonant al dipòsit nu «curs-r-introduccio.git»...
remote: Enumerating objects: 311, done.
remote: Counting objects: 100% (311/311), done.
remote: Compressing objects: 100% (165/165), done.
remote: Total 854 (delta 174), reused 276 (delta 145)
S'estan rebent objectes: 100% (854/854), 383.42 MiB | 714.00 KiB/s, fet.
S'estan resolent les diferències: 100% (494/494), fet.
xavi@tricholoma:~/tmp$ java -jar ../bfg-1.13.0.jar --strip-blobs-bigger-than
22Mcurs-r-introduccio.git
Error: Unable to access jarfile ../bfg-1.13.0.jar
xavi@tricholoma:~/tmp$ java -jar ../bfg-1.13.0.jar --strip-blobs-bigger-than
22Mcurs-r-introduccio.git
Error: Unable to access jarfile ../bfg-1.13.0.jar
xavi@tricholoma:~/tmp$ cp ../bfg-1.13.0.jar .
cp: ha fallat stat() sobre '../bfg-1.13.0.jar': El fitxer o directori no existeix
xavi@tricholoma:~/tmp$ java -jar bfg-1.13.0.jar --strip-blobs-bigger-than 22M curs-
r-introduccio.git

Using repo : /media/xavi/Maxtor/XAVI_01/tmp/curs-r-introduccio.git

Scanning packfile for large blobs: 854
Scanning packfile for large blobs completed in 374 ms.
Found 7 blob ids for large blobs - biggest=89772799 smallest=24005151
Total size (unpacked)=433099439
Found 131 objects to protect
Found 2 commit-pointing refs : HEAD, refs/heads/master

Protected commits
-----------------

These are your protected commits, and so their contents will NOT be altered:

 * commit 7b14cd56 (protected by 'HEAD')

Cleaning
--------
```

```
Found 166 commits
Cleaning commits:       100% (166/166)
Cleaning commits completed in 263 ms.

Updating 1 Ref
--------------

        Ref                 Before     After
        ---------------------------------------
        refs/heads/master | 7b14cd56 | fc566573

Updating references:    100% (1/1)
...Ref update completed in 19 ms.

Commit Tree-Dirt History
------------------------

        Earliest                                              Latest
        |                                                        |
        ..................................................DDmmmmmmmm

        D = dirty commits (file tree fixed)
        m = modified commits (commit message or parents changed)
        . = clean commits (no changes to file tree)

                              Before     After
        ----------------------------------------
        First modified commit | aa7b8697 | ff4173a2
        Last dirty commit     | aa8623c2 | 051e8b60

Deleted files
-------------

        Filename             Git id
        ---------------------------------------
        arxiu               | e48cc445 (75,9 MB)
        arxiu.50.fst        | 9488b9cf (28,6 MB)
        arxiu.base.csv      | 917953d0 (85,6 MB)
        arxiu.base.csv.gz   | a9f3b581 (22,9 MB)
        arxiu.datatable.csv | e48cc445 (75,9 MB)
        arxiu.feather       | c53071c0 (60,4 MB)
        arxiu.tidyverse.csv | b9e892f8 (75,9 MB)
        arxiu.xlsx          | cfdd5af5 (63,7 MB)


In total, 30 object ids were changed. Full details are logged here:

        /media/xavi/Maxtor/XAVI_01/tmp/curs-r-introduccio.git.bfg-
report/2019-03-19/01-52-58
```

```
BFG run is complete! When ready, run: git reflog expire --expire=now --all && git
gc --prune=now --aggressive


--
You can rewrite history in Git - don't let Trump do it for real!
Trump's administration has lied consistently, to make people give up on ever
being told the truth. Don't give up: https://github.com/bkeepers/stop-trump
--


xavi@tricholoma:~/tmp$ java -jar bfg-1.13.0.jar --delete-files arxiu.gz curs-r-
introduccio.git

Using repo : /media/xavi/Maxtor/XAVI_01/tmp/curs-r-introduccio.git


This repo has been processed by The BFG before! Will prune repo before proceeding -
to avoid unnecessary cleaning work on unused objects...
Completed prune of old objects - will now proceed with the main job!

Found 131 objects to protect
Found 2 commit-pointing refs : HEAD, refs/heads/master

Protected commits
-----------------

These are your protected commits, and so their contents will NOT be altered:

 * commit fc566573 (protected by 'HEAD')

Cleaning
--------

Found 166 commits
Cleaning commits:       100% (166/166)
Cleaning commits completed in 603 ms.

Updating 1 Ref
--------------

        Ref                 Before     After
        ---------------------------------------
        refs/heads/master | fc566573 | 3abf839a

Updating references:    100% (1/1)
...Ref update completed in 6 ms.

Commit Tree-Dirt History
```

```
-----------------------

        Earliest                                            Latest
        |                                                        |
        ..................................................DDmmmmmmmm

        D = dirty commits (file tree fixed)
        m = modified commits (commit message or parents changed)
        . = clean commits (no changes to file tree)

                              Before     After
        ----------------------------------------
        First modified commit | ff4173a2 | 4d1f5d89
        Last dirty commit     | 051e8b60 | 903d8c77

Deleted files
-------------

        Filename   Git id
        ---------------------------
        arxiu.gz | bd2f68cd (20,1 MB)


In total, 60 object ids were changed. Full details are logged here:

        /media/xavi/Maxtor/XAVI_01/tmp/curs-r-introduccio.git.bfg-
report/2019-03-19/01-55-53

BFG run is complete! When ready, run: git reflog expire --expire=now --all && git
gc --prune=now --aggressive


--
You can rewrite history in Git - don't let Trump do it for real!
Trump's administration has lied consistently, to make people give up on ever
being told the truth. Don't give up: https://www.aclu.org/
--


xavi@tricholoma:~/tmp$ cd curs-r-introduccio.git
xavi@tricholoma:~/tmp/curs-r-introduccio.git$ git reflog expire --expire=now --all
&& git gc --prune=now --aggressive
S'estan comptant els objectes: 854, fet.
Delta compression using up to 4 threads.
S'estan comprimint objectes: 100% (840/840), fet.
S'estan escrivint els objectes: 100% (854/854), fet.
Total 854 (delta 525), reused 299 (delta 0)
xavi@tricholoma:~/tmp/curs-r-introduccio.git$ git push
Username for 'https://gitlab.com': xavidp
```

```
Password for 'https://xavidp@gitlab.com':
S'estan comptant els objectes: 854, fet.
Delta compression using up to 4 threads.
S'estan comprimint objectes: 100% (315/315), fet.
S'estan escrivint els objectes: 100% (854/854), 205.11 MiB | 1.44 MiB/s, fet.
Total 854 (delta 525), reused 854 (delta 525)
remote: Resolving deltas: 100% (525/525), done.
remote: GitLab: You are not allowed to force push code to a protected branch on
this project.
To https://gitlab.com/radup/curs-r-introduccio.git
 ! [remote rejected] master -> master (pre-receive hook declined)
error: s'ha produït un error en pujar algunes referències a
«https://gitlab.com/radup/curs-r-introduccio.git»
xavi@tricholoma:~/tmp/curs-r-introduccio.git$ git push
Username for 'https://gitlab.com': xavidp
Password for 'https://xavidp@gitlab.com':
S'estan comptant els objectes: 854, fet.
Delta compression using up to 4 threads.
S'estan comprimint objectes: 100% (315/315), fet.
S'estan escrivint els objectes: 100% (854/854), 205.11 MiB | 1.44 MiB/s, fet.
Total 854 (delta 525), reused 854 (delta 525)
remote: Resolving deltas: 100% (525/525), done.
To https://gitlab.com/radup/curs-r-introduccio.git
 + 7b14cd5...3abf839 master -> master (forced update)
xavi@tricholoma:~/tmp/curs-r-introduccio.git$ cd ..
xavi@tricholoma:~/tmp$ cd ..
xavi@tricholoma:~$ cd tmp2
xavi@tricholoma:~/tmp2$ ls
xavi@tricholoma:~/tmp2$ git clone https://gitlab.com/radup/curs-r-introduccio.git
S'està clonant a «curs-r-introduccio»...
remote: Enumerating objects: 311, done.
remote: Counting objects: 100% (311/311), done.
remote: Compressing objects: 100% (125/125), done.
remote: Total 854 (delta 179), reused 306 (delta 178)
S'estan rebent objectes: 100% (854/854), 205.47 MiB | 638.00 KiB/s, fet.
S'estan resolent les diferències: 100% (499/499), fet.
xavi@tricholoma:~/tmp2$
```

And after that, you need to Navigate to **Project > Settings > Repository > Repository Cleanup**:



Upload the `object-id-map.old-new.txt` file and press Start cleanup. This will remove any internal git references to the old commits, and run git gc against the repository. You will receive an email once it has completed

1.15. Change origin from https to ssh

Verify origin of your repo through a terminal window in the root folder where your project code is located:

```
git remote -v
```

It should say something like:

```
origin  http://gitlab.com/{USERNAME}/{PROJECTNAME}.git (fetch)
origin  http://gitlab.com/{USERNAME}/{PROJECTNAME}.git (push)
```

Then you can run in your terminal window the following command

```
git remote set-url origin git@gitlab.com:{USERNAME}/{PROJECTNAME}.git
```

And last, you can check again that the change was successful, with:

```
git remote -v
```

Alias names for this page: github | gitlab

---

[1] http://ueb.vhir.org/SeminarDVCS

[2] http://www.r-bloggers.com/rstudio-and-github/

[3] https://gitlab.com/xavidp/

[4] https://github.com/xavidp/

[5] https://gitforwindows.org

[6] https://github.com/xavidp/bashscripts/blob/master/fixgitueb.sh

[7] http://ueb.vhir.org/Bazaar#Convert_between_Git_and_Bazaar_and_others_

[8] https://gitlab.com

[9] http://phabricator.org

[10] http://git-cola.github.io/downloads.html

[11] https://www.maketecheasier.com/6-useful-graphical-git-client-for-linux/

[12] https://www.gitkraken.com/

[13] https://portableapps.com

[14] https://gitforwindows.org

[15] https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#generating-a-new-ssh-key

[16] https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/

[17] https://docs.gitlab.com/ee/user/project/repository/reducing_the_repo_size_using_git.html

[18] https://rtyley.github.io/bfg-repo-cleaner/