01-GitBasics

August 11, 2020

1 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Official website https://git-scm.com

1.1 GitHub

- Web-based hosting service for version control using Git.
- Offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features.
- Provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.
- Github is the largest host of source code in the world.

1.2 About SCM

- Records changes to a file or set of files over time.
- You can recall specific versions later.
- You can use it with nearly any type of file on a computer.
- This is the better way to collaborate on the same document.
- Every change is committed with an author and a date.
- Figures are downloaded from Pro Git book.
- "Become a git guru" tutorial (https://www.atlassian.com/git/tutorials).

1.3 Local Version Control System

- Collaboration is not really possible.

1.4 Distributed Version Control Systems

- Clients fully mirror the repository.
- You can collaborate with different groups of people in different ways simultaneously within the same project.

- No need of network connection.
- Multiple backups.

1.5 Configure Git

Settings are saved on the computer for all your git repositories.

```
git config --global user.name "Prenom Nom"
git config --global user.email "prenom.nom@univ-rennes2.fr"
```

1.6 Cloning the repository

```
git clone ssh://svmass2/git/big-data.git
```

or

```
git clone https://github.com/pnavaro/big-data.git
```

To save your work locally create a branch

```
git checkout -b my_branch
```

1.7 Four File status in the repository

1.8 Git Workflow

1.9 Locally saving your modifications

Get your files status

[1]:
```
%%bash
git status
```

```
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>…" to include in what will be committed)
        ../.doit.db.dat
        ../spark-3.0.0-bin-hadoop2.7.tgz
        ../spark-3.0.0-bin-hadoop2.7/

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

1.10 Add the modified file to the index

```
git add your_notebook_copy.ipynb
```

Checking which files are ready to be committed.

[2]:
```
%%bash
git status
```

```
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>…" to include in what will be committed)
        ../.doit.db.dat
        ../spark-3.0.0-bin-hadoop2.7.tgz
        ../spark-3.0.0-bin-hadoop2.7/

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

Now save your work, the branch is local.

```
git commit -m 'Some modifications'
```

1.11 Updating from the Repository

If the master branch has changed. To get all new updates :

[3]:
```
%%bash
git pull origin master
```

```
Updating ddc43b0..a4097ac
Fast-forward
 .github/workflows/deploy.yml | 18 ------------------
 1 file changed, 18 deletions(-)

warning: Pulling without specifying how to reconcile divergent branches is
discouraged. You can squelch this message by running one of the following
commands sometime before your next pull:

  git config pull.rebase false  # merge (the default strategy)
  git config pull.rebase true   # rebase
  git config pull.ff only       # fast-forward only

You can replace "git config" with "git config --global" to set a default
preference for all repositories. You can also pass --rebase, --no-rebase,
or --ff-only on the command line to override the configured default per
invocation.

From https://github.com/pnavaro/big-data
 * branch            master     -> FETCH_HEAD
   ddc43b0..a4097ac  master     -> origin/master
```

1.12 Solve conflicts

- If you have some conflicts, no problem just do :

```
git checkout notebook.ipynb
```

It will give you back the original version of notebook.ipynb

- If you have big troubles, you can do

```
git reset --hard
```

Be careful with this last command, you remove uncommited changes.

1.13 Git through IDE

- Install bash-completion and source git-prompt.sh.
- Use Gui tools:
  - GitHub Desktop
  - Sourcetree
  - GitKraken
- VCS plugin of IDE
  - RStudio
  - Eclipse
  - JetBrains