



git

Introduction to the version-control system

By Nicolas Mattelaer

infoep
org



Preface

- **Nicolas Mattelaer**
- nmattela@infogroep.be
- discord.infogroep.be
- Having your laptop with you is recommended.
- Slides will become available on seminars.infogroep.be
- Questions? just ask.

Contents

- **What is Git?**
- Installation of Git
- Git usage

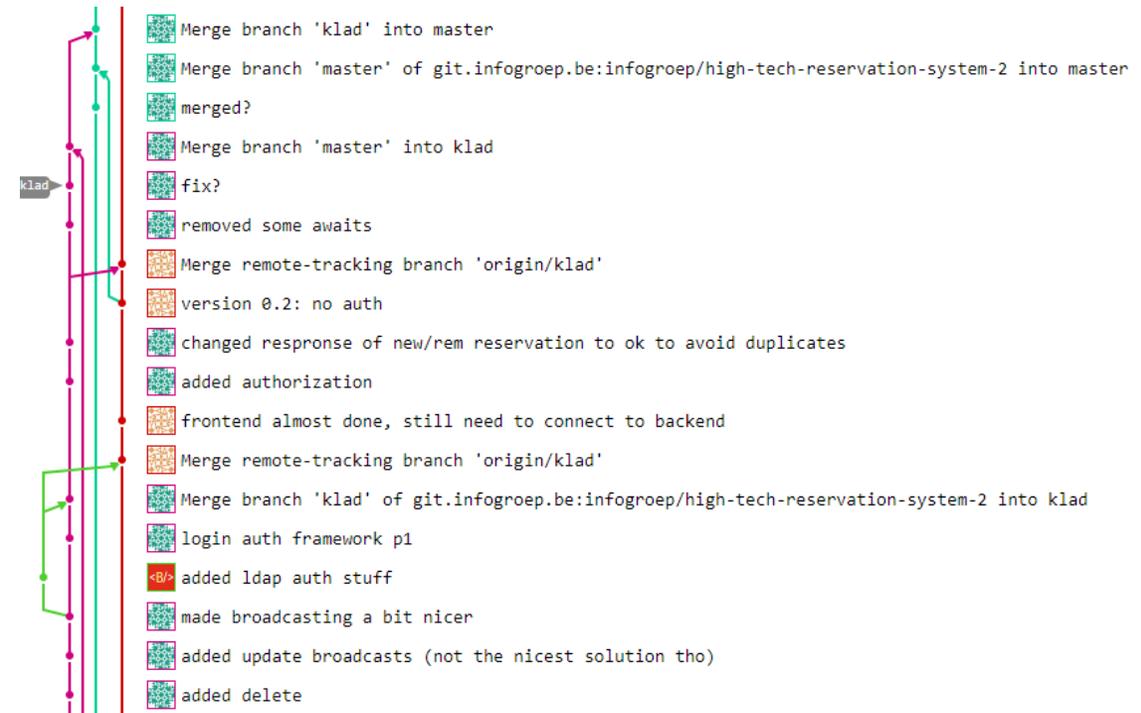
What is Git?

- A version-control system
- Invented by Linus Torvalds
- Keeps track of project versions
 - Keeps history of changes
 - Always have a backup available
- Easy code collaboration
 - Share code with others
 - Accept/reject code changes by others



“Controlling of versions”

- Multiple **branches**
 - Either one branch per developer
 - Or one branch per feature...
- One ‘master’/‘main’ branch
 - Represents the current, stable version
- Branches can
 - Spawn (branching)
 - Merge



“The social media for developers”

- Git services
 - Github
 - Gitlab
- Where people can
 - Publish projects
 - Contribute to project
 - Star projects
 - Report issues/bugs

The screenshot displays a GitHub profile for Nicolas Mattelaer (nmattela). The profile shows 0 starred topics and 5 starred repositories. The main content area is titled "Here's what we found based on your interests..." and is divided into two sections:

- Based on topics you're interested in:** This section features the repository `godotengine / godot` with 34.2k stars. It is described as a "Multi-platform 2D and 3D game engine" and includes tags for `open-source`, `gamedev`, `multi-platform`, `game-engine`, and `game-development`. It also lists associated topics like `godotengine`, `godot`, and `hacktoberfest`. The repository was updated 2 hours ago and is written in C++.
- Based on repositories you've starred:** This section features the repository `cmulk / wireguard-docker` with 224 stars. It is described as "Wireguard setup in Docker meant for a simple personal VPN" and was updated 16 days ago. It is written in Shell.

On the right side of the page, there are two additional sections:

- Trending repositories today:** This section lists several trending repositories, including `jonasschmedtmann / complete-javascript-course` (4.9k stars), `tomor8007 / widevine-l3-decryptor` (398 stars), `hyb1996 / Auto.js` (7.9k stars), and `pytorch / fairseq` (9.8k stars).
- Trending developers:** This section is currently empty.

Contents

- What is Git?
- **Installation of Git**
- Git usage

Install Git

- Linux
 - Ubuntu/Debian: `sudo apt install git-core`
 - Arch: `sudo pacman -S git`
- MacOS
 - Install Xcode Command Line Tools
- Windows
 - Install Git Bash: <https://git-scm.com/download/win>

Contents

- What is Git?
- Installation of Git
- **Git usage**

Github Account

- We will use Github in this tutorial
- Create an account on github.com



Setting up SSH keys



- `ssh-keygen -t ed25519 -C "your_email@example.com"`
- When prompted for password, just press Enter
- `eval "$(ssh-agent -s)"`
- `ssh-add ~/.ssh/id_ed25519`
- `git config --global user.name "Your name"`
- `git config -global user.email "Your email"`

Setting up SSH keys

- Go to settings in Github (not profile)
- Go to SSH and GPG keys
- Copy output of `cat ~/.ssh/id_ed25519.pub`
- Click "New SSH key" and paste in "Key" field



Creating our first repository



- Create a new repository on github.com/new

Connect to repository

1. Create a new directory

```
mkdir <directory name> && git init <directory name>
```

2. Connect with remote

```
cd <directory name> && git remote add origin <link to repo>
```

1. Or... do

```
git clone <link to repo>
```

Adding files

1. Create a new file
2. Update the index

`git add <name of file>` for a single file

`git add .` for all files (recommended)

3. Update the local repository

`git commit -m "a concise message"` (good practice to use meaningful messages)

4. Update the remote repository

`git push origin master`

`origin` = remote repo

`master` = branch to push to

gitignore

- A special file `.gitignore`
- Keep some files from being pushed along
- List all the names of unnecessary files
- You can also use patterns
 - The wildcard `*`

```
.gitignore 318 Bytes
1 # See https://help.github.com/articles/ignoring-files/ for more about ignoring files.
2
3 # dependencies
4 /node_modules
5 /.pnp
6 .pnp.js
7
8 .idea/
9
10 # testing
11 /coverage
12
13 # production
14 /build
15
16 # misc
17 .DS_Store
18 .env.local
19 .env.development.local
20 .env.test.local
21 .env.production.local
22
23 npm-debug.log*
24 yarn-debug.log*
25 yarn-error.log*
```

Branching & Merging

- Create a new branch

`git branch <new branch name>` Creates the branch

`git checkout <new branch name>` Moves local repo to new branch

`git push origin <new branch name>` Pushes to new branch

- Merge the new branch with another

`git checkout master` Move back to master branch

`git merge <new branch name>` Merge new branch with master

Cloning repositories

1. Clone my test repository

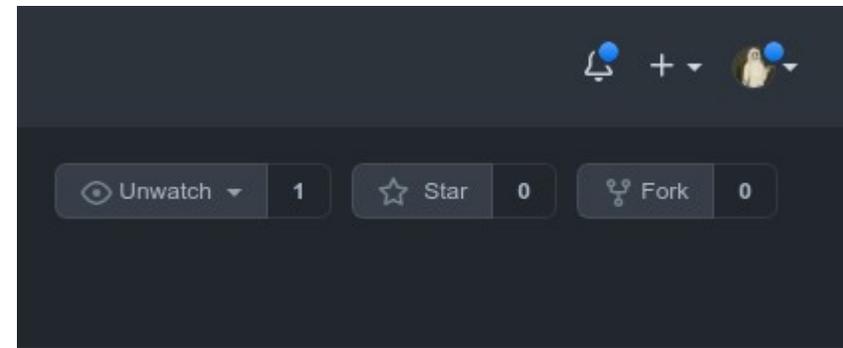
- `cd ~/Documents` Return to where you cloned your previous repo
- `git clone git@github.com:nmattela/git_lesje.git` Creates a new directory where the project appears

2. Pushing changes will not work!

- I have not granted you permission to make changes, but...
- You can fork the repository (make a clone), then open a pull request

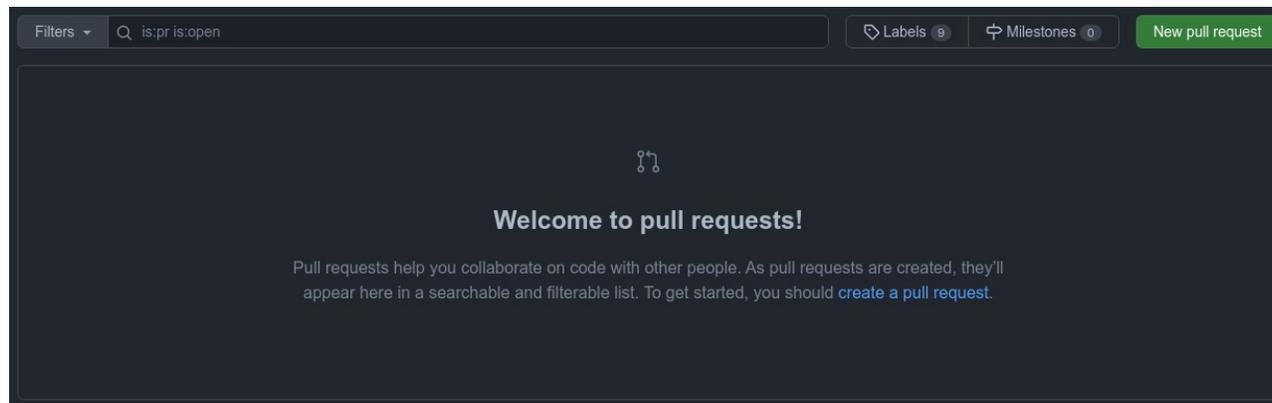
Forking a repo

1. Click on the fork button
 - This will create a new repo owned by you
2. Make changes to this repo & push to remote
- 3.



Pull request

- Pushing on the forked repo will not change the original repo
- Open a pull request on the forked repo to the original
- If accepted, it will be added to the original repo



Merge conflicts

- 2 people adapted the same file and pushed to remote
- Git does not know which version is the correct one
- Message appears in file with the conflict
 - Choose which version to keep
 - Or adapt

```
14 <tr>
15 <td>Harvey</td>
16 <td>Jennings</td>
17 <td>Leaving Soon</td>
18 <td>hjennings@atlassian.com</td>
19 <<<<<<< HEAD
20 <td>2</td>
21 <td>B</td>
22 =====
23 <td>2-B</td>
24 </tr>
25 <tr>
26 <td>Ryan</td>
27 <td>Lee</td>
28 <td>New Hire</td>
29 <td>rlee@atlassian.com</td>
30 <td>5-E</td>
31 >>>>>> 2a04e55405e527fb7924888b3ee0336a24849cf1
32 </tr>
33 <tr>
34 <td>Alana</td>
35 <td>Grant</td>
36 <td>Current</td>
37 <td>agrant@atlassian.com</td>
```

END :)