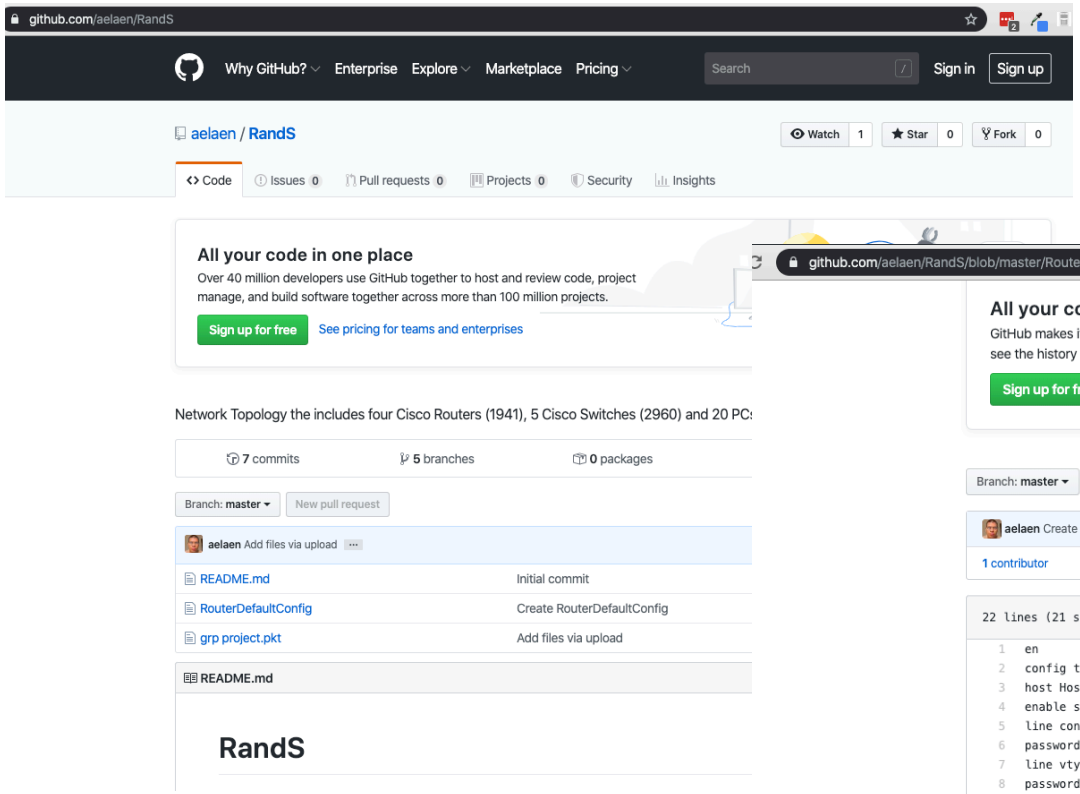
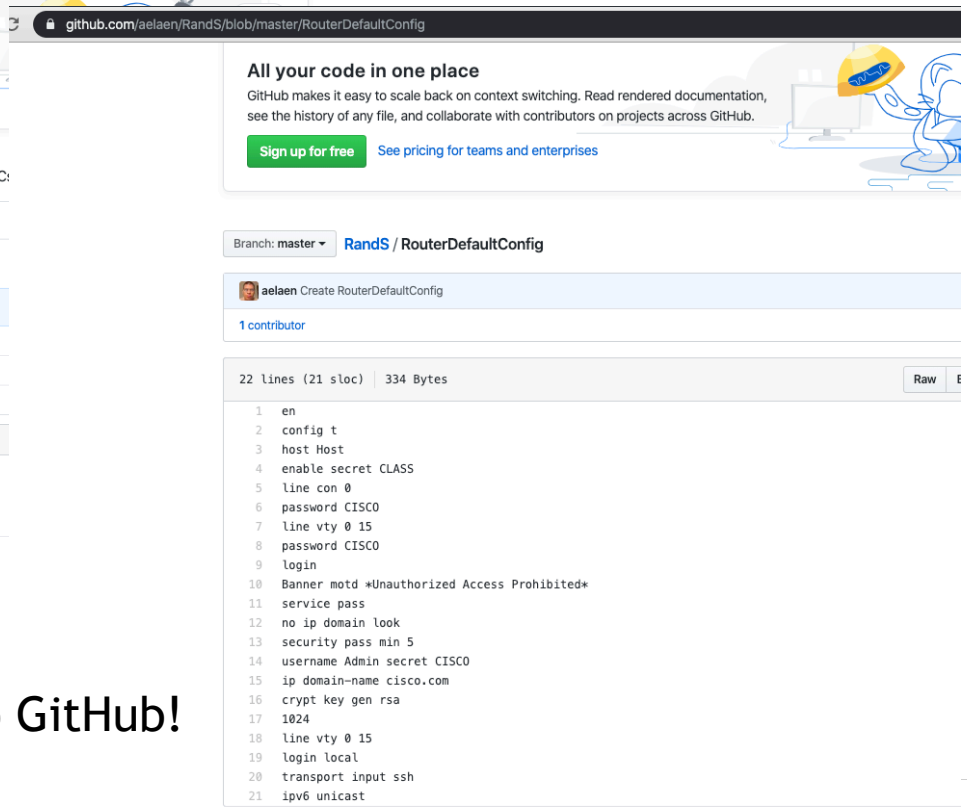
Versioning & Deployment

Les 7 20191114

M. DIMA

Fictieve Usecase

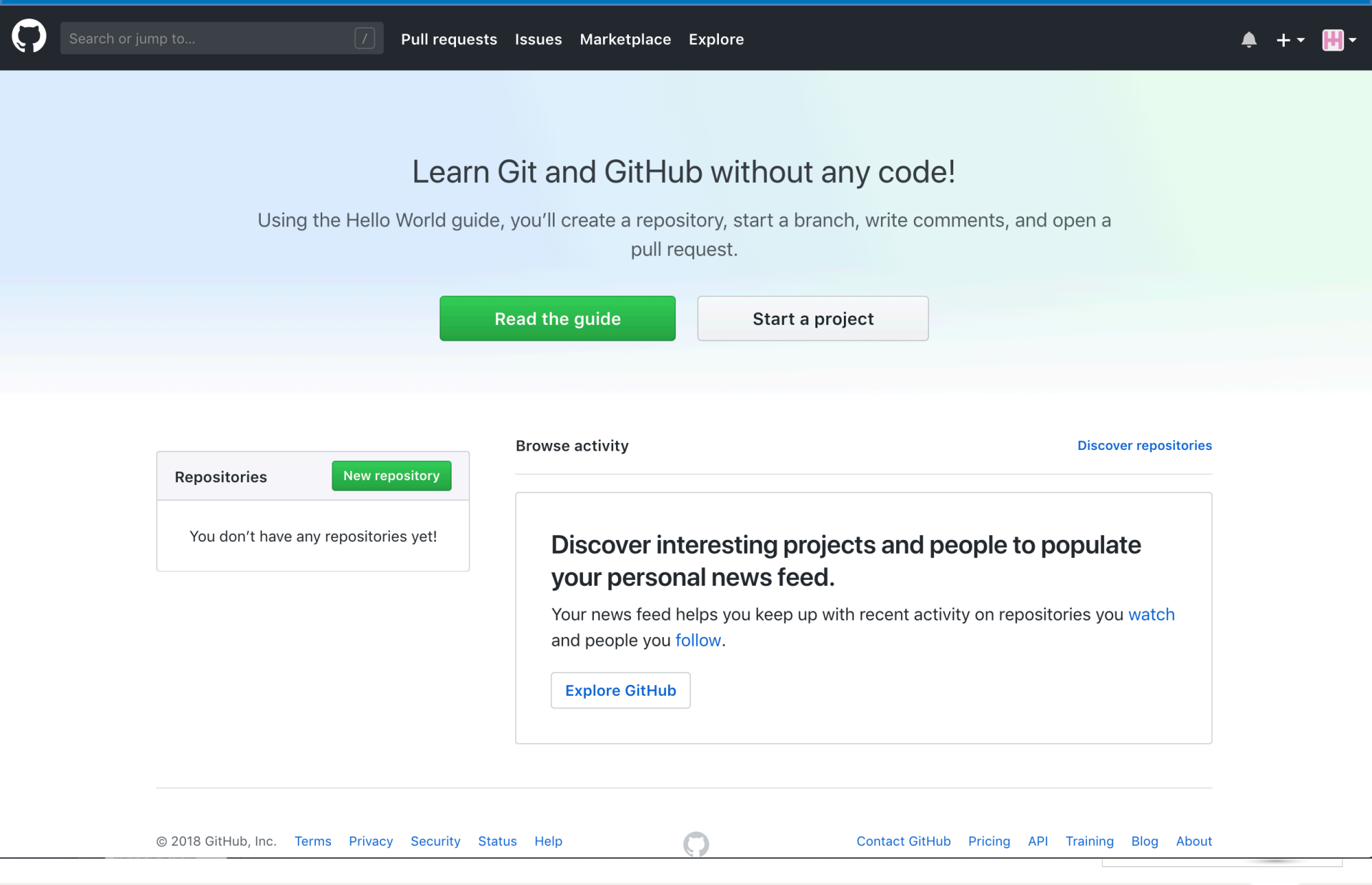




Configuratiebestanden staan op GitHub!
Wat nu?

katholieke hogeschool
associatie KU Leuven
vives

GitHub


Search or jump to...
/
Pull requests
Issues
Marketplace
Explore
Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.
Read the guide
Start a project
Repositories
New repository
You don't have any repositories yet!
Browse activity
Discover repositories
Discover interesting projects and people to populate your personal news feed.
Your news feed helps you keep up with recent activity on repositories you watch and people you follow.
Explore GitHub
© 2018 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About

GitHub

GitHub

GitHub is een populaire[1] website waarop software kan geplaatst worden. GitHub is gebouwd rond het Git-versiebeheersysteem, waardoor GitHub alle mogelijkheden van Git en eigen toevoegingen aanbiedt.

Het beschikt onder ander over toegangscontrole en verschillende samenwerkingsfuncties, zoals een issue tracker, een forum voor het aanvragen van functies, takenlijsten en wiki's voor ieder project.[2]

Op GitHub staat veel opensourcesoftware omdat bij de gratis versie standaard de broncode kan ingekeken worden door derden.

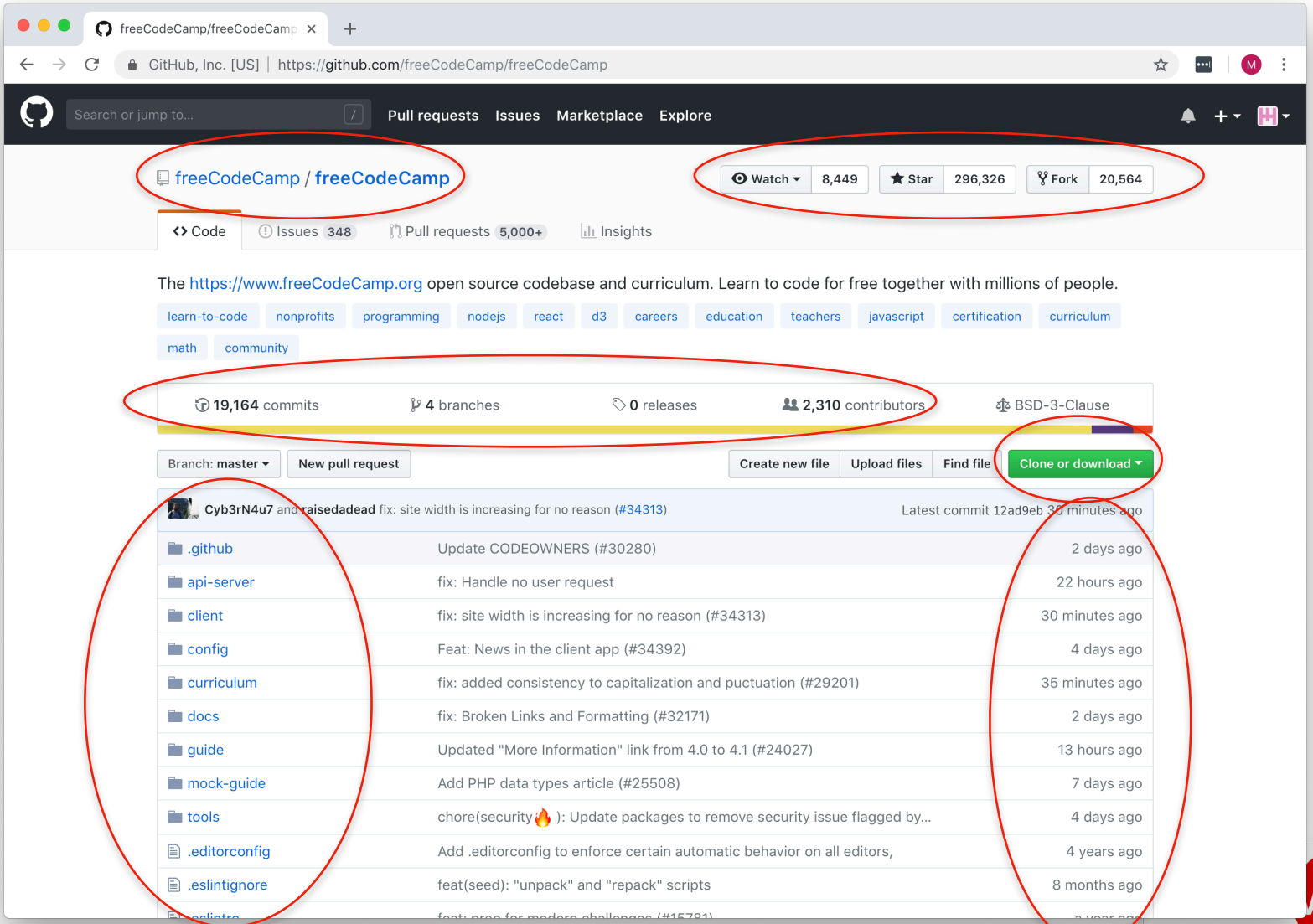
GitHub biedt zowel gratis en privé-repositories. Om privé-projecten te kunnen hosten op Github is een betaald abonnement nodig.[3]

Bron: Wikipedia https://nl.wikipedia.org/wiki/GitHub

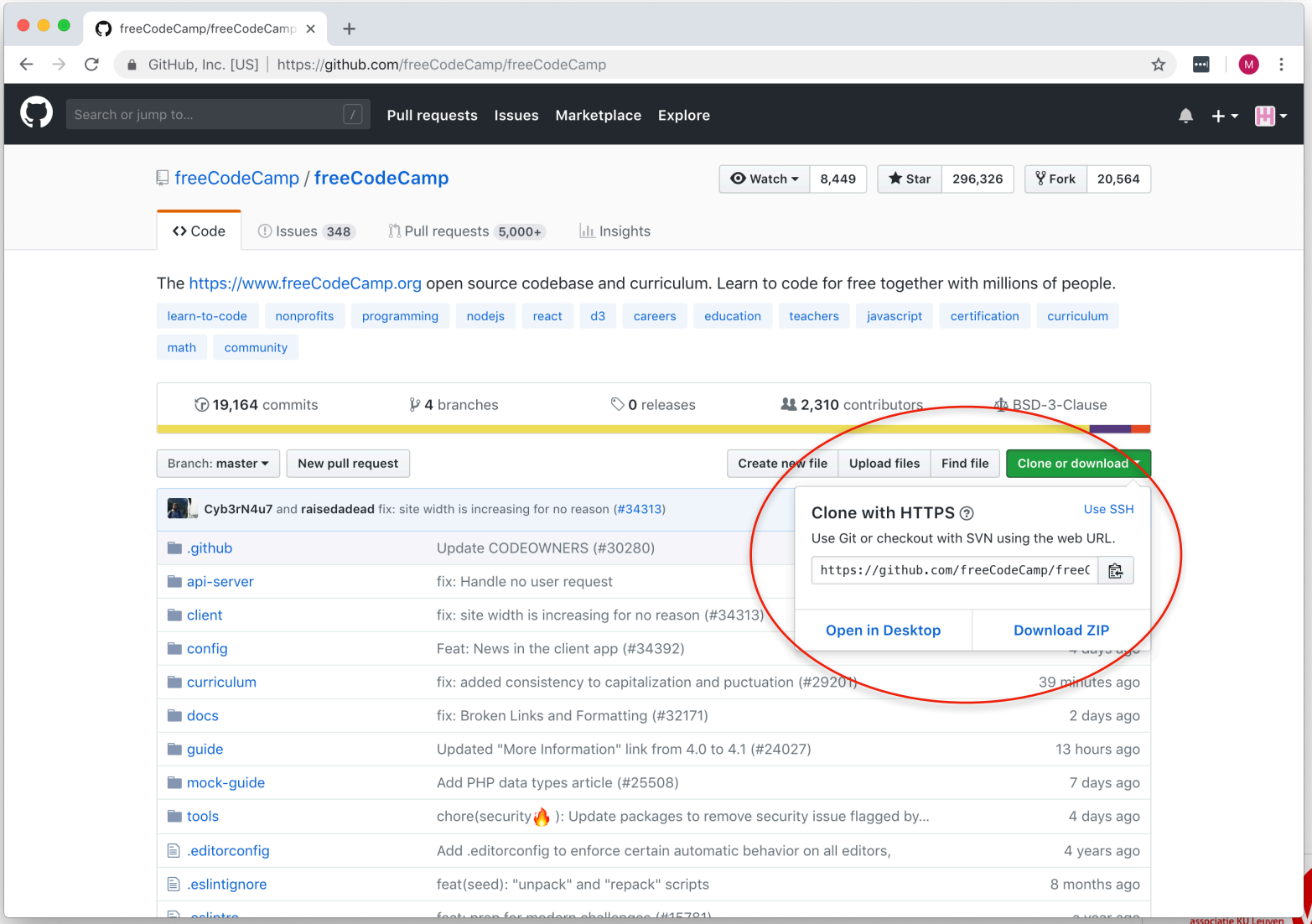
Alternatieven

- Self hosting	- Gogs
- GitLab	- Gitea
- Bitbucket	- Apache Allura
- Beanstalk	- Cloud Source (Google)
- Launchpad	- AWS CodeCommit
- Sourceforge	- GitKraken
- Phabricator	- Buddy
- GitBucket	- RhodeCode
- CodeGiant	- …

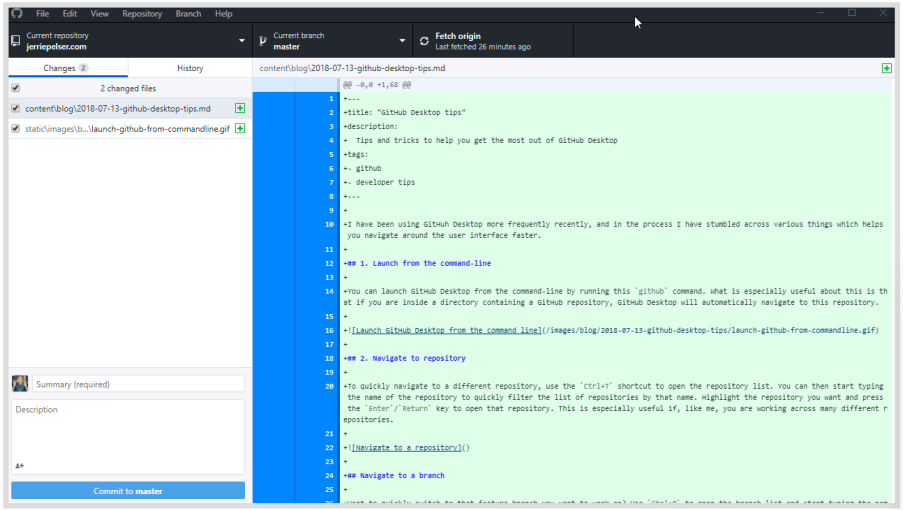
Repositories



Repositories



Desktop VS CLI



Install Git

Git: https://git-scm.com/downloads

MacOS brew: $ brew install git
Linux: sudo apt-get install git

GitHub
Windows: https://central.github.com/deployments/desktop/desktop/latest/win32
Mac: https://central.github.com/deployments/desktop/desktop/latest/darwin

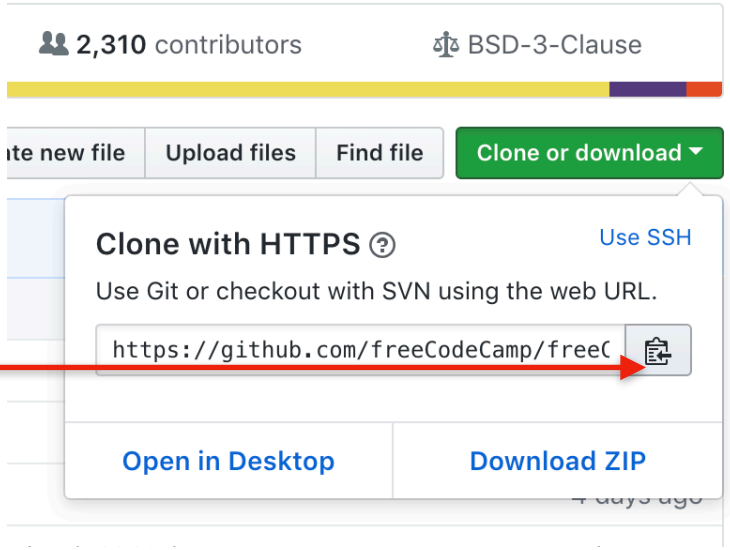
https://gist.github.com/derhuerst/1b15ff4652a867391f03

Git clone

= lokaal een kopie maken van een online repo
bv. Alles van freeCodeCamp lokaal kopiëren

Stappenplan:

1. Maak een lokale directory
2. Kopieer de clone link van git
3. In de lokale directory:

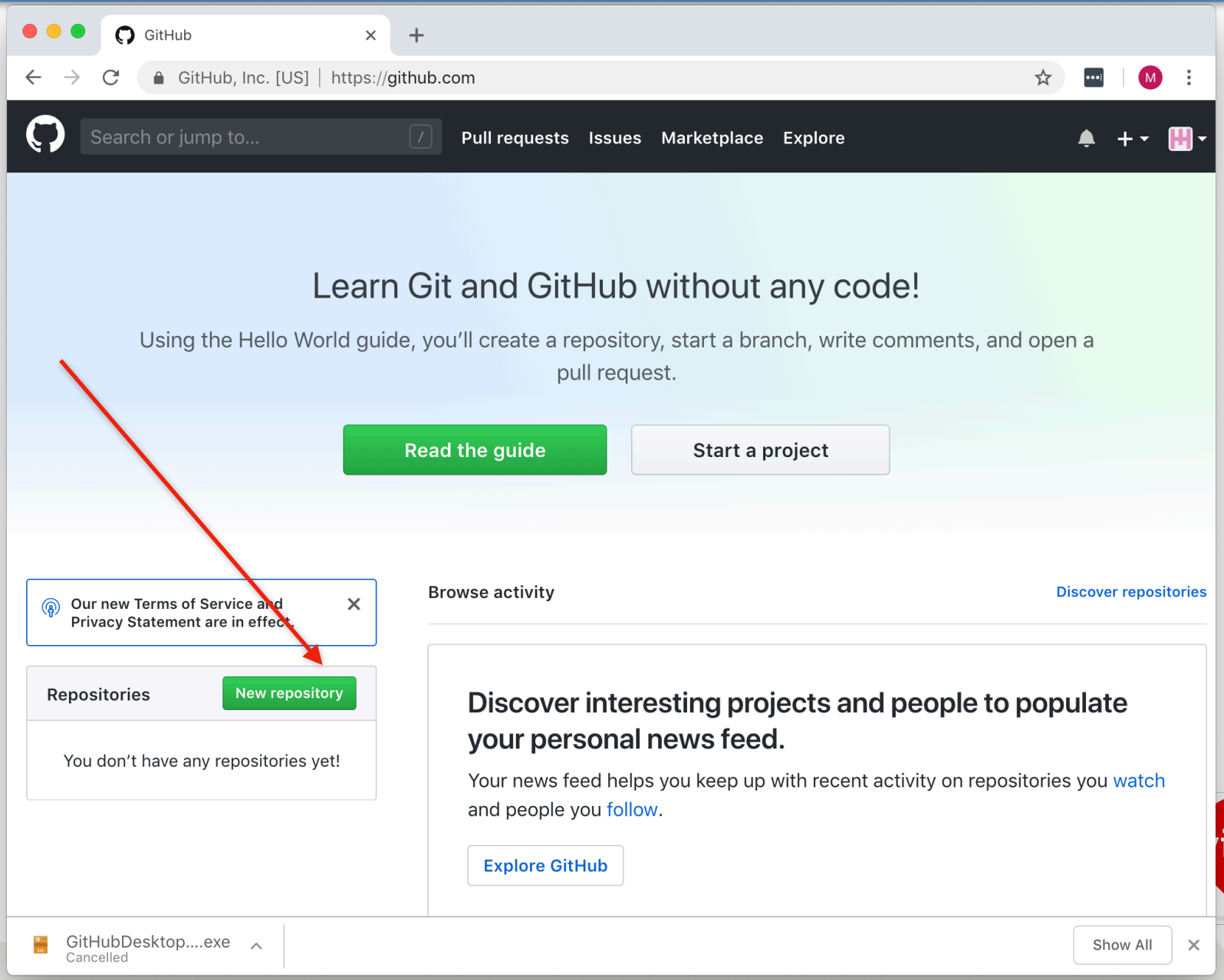


git clone https://github.com/freeCodeCamp/freeCodeCamp.git

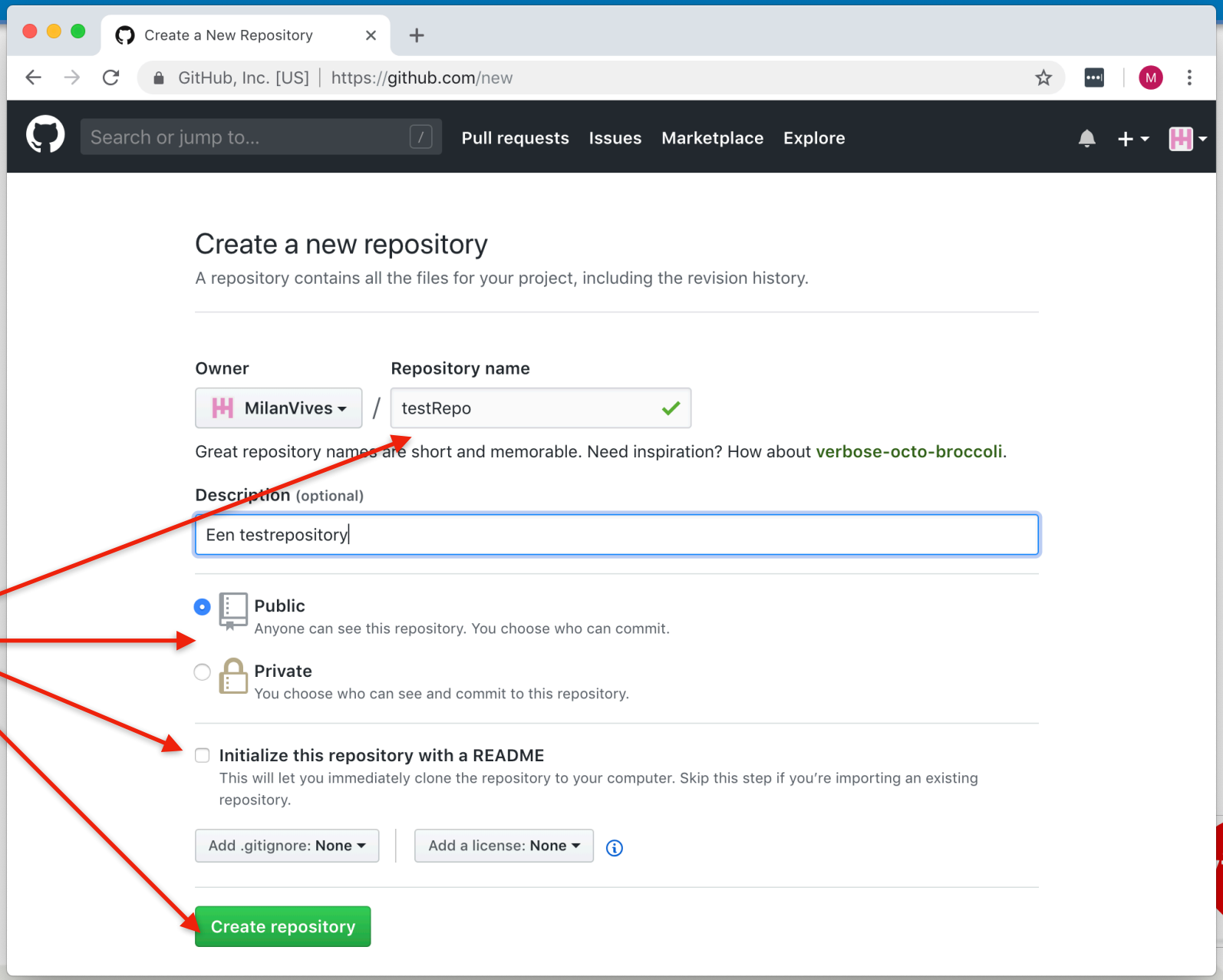
```
freecodecamp — -bash — 100×15
[milans-MacBook-Pro:~ doc$ mkdir freecodecamp
[milans-MacBook-Pro:~ doc$ cd freecodecamp/
[milans-MacBook-Pro:freecodecamp doc$ git clone https://github.com/freeCodeCamp/freeCodeCamp.git
Cloning into 'freeCodeCamp'...
remote: Enumerating objects: 108, done.
remote: Counting objects: 100% (108/108), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 176227 (delta 48), reused 55 (delta 37), pack-reused 176119
Receiving objects: 100% (176227/176227), 100.03 MiB | 2.03 MiB/s, done.
Resolving deltas: 100% (84100/84100), done.
Checking out files: 100% (36465/36465), done.
[milans-MacBook-Pro:freecodecamp doc$ ls
freeCodeCamp
milans-MacBook-Pro:freecodecamp doc$
```

katholieke hogeschool associatie KU Leuven vives

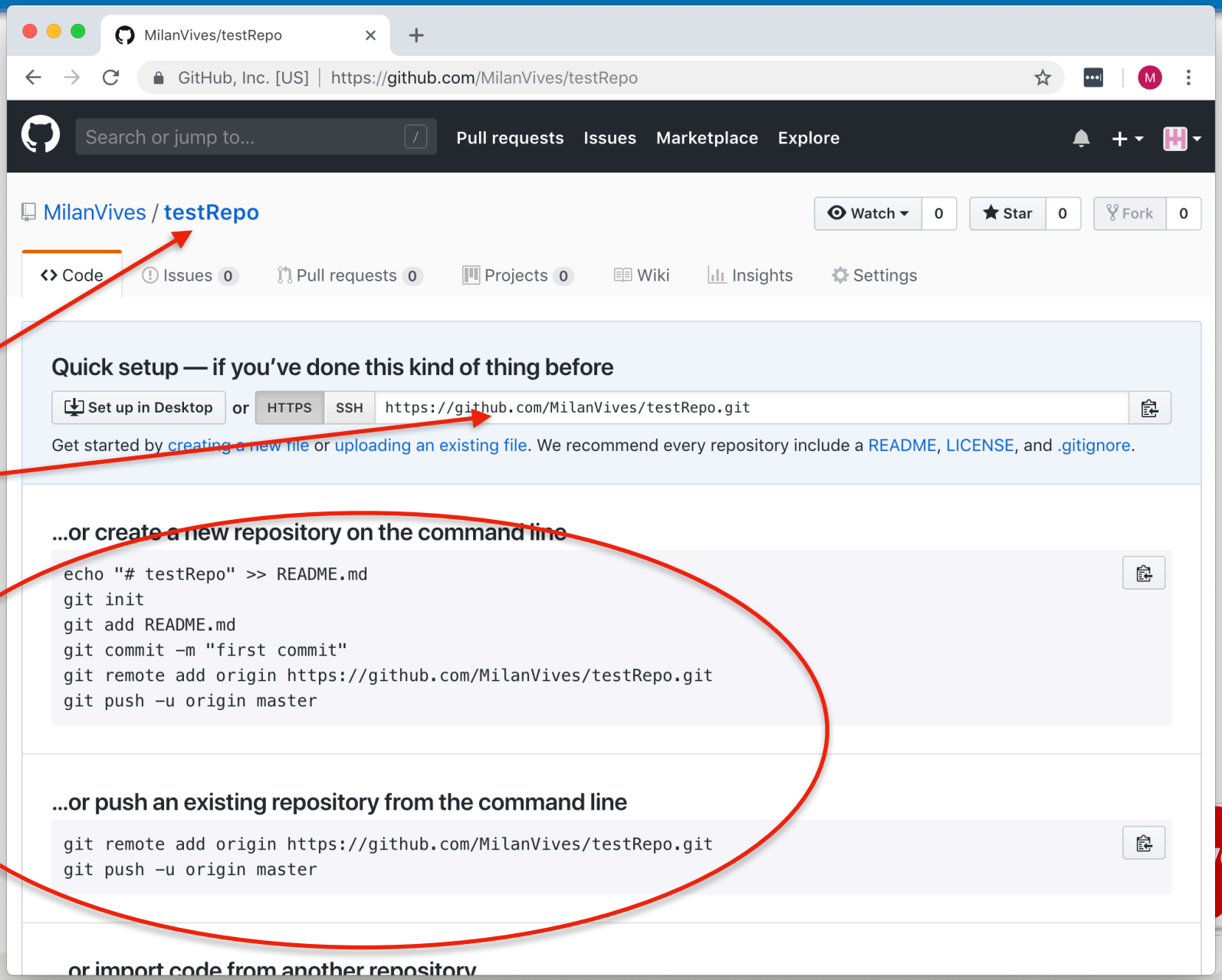
Zelf een repository aanmaken



Zelf een repository aanmaken



Zelf een repository aanmaken



Zelf een repository aanmaken

Create a new repository on the command line

```
echo "# testRepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/MilanVives/testRepo.git
git push -u origin master
```

STAPPENPLAN

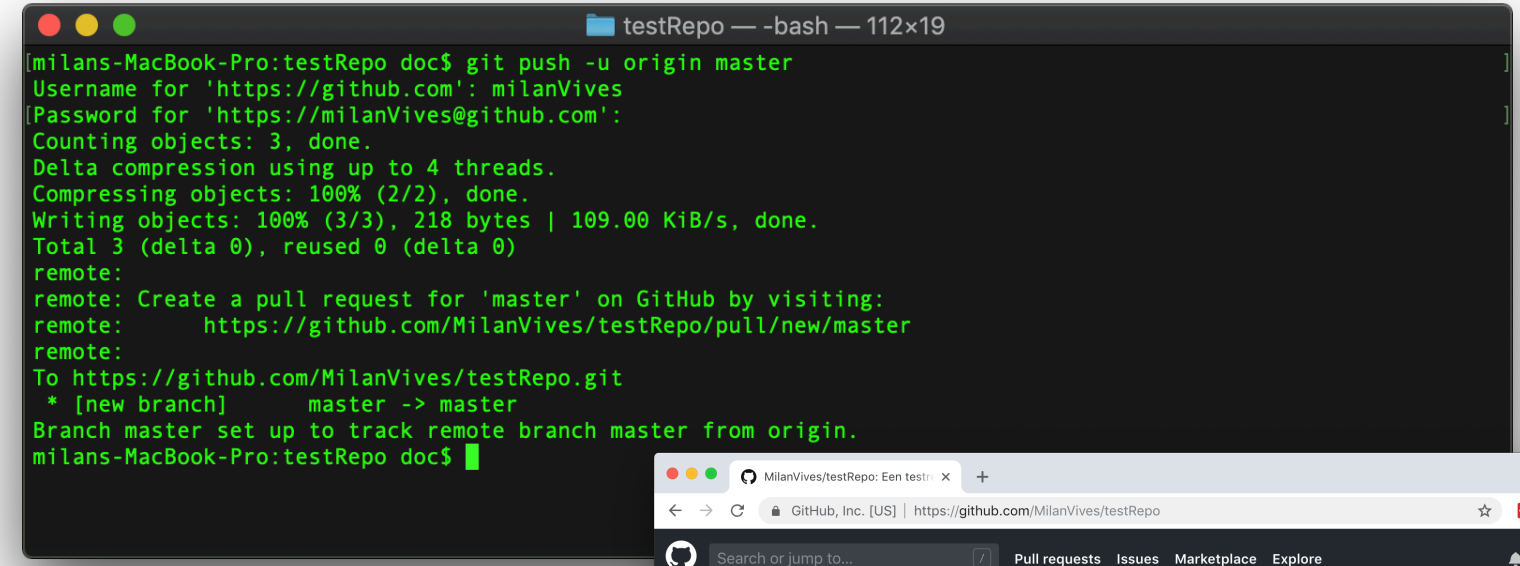
1.git init // git lokaal initialiseren
2.git add . // alle files toevoegen
3. git remote add origin https://github.com/MilanVives/testRepo.git
// naar waar wil ik pushen?
4. git commit -m “commitboodschap” // wijzigingen committen
5. git push -u origin master // push alles naar remote

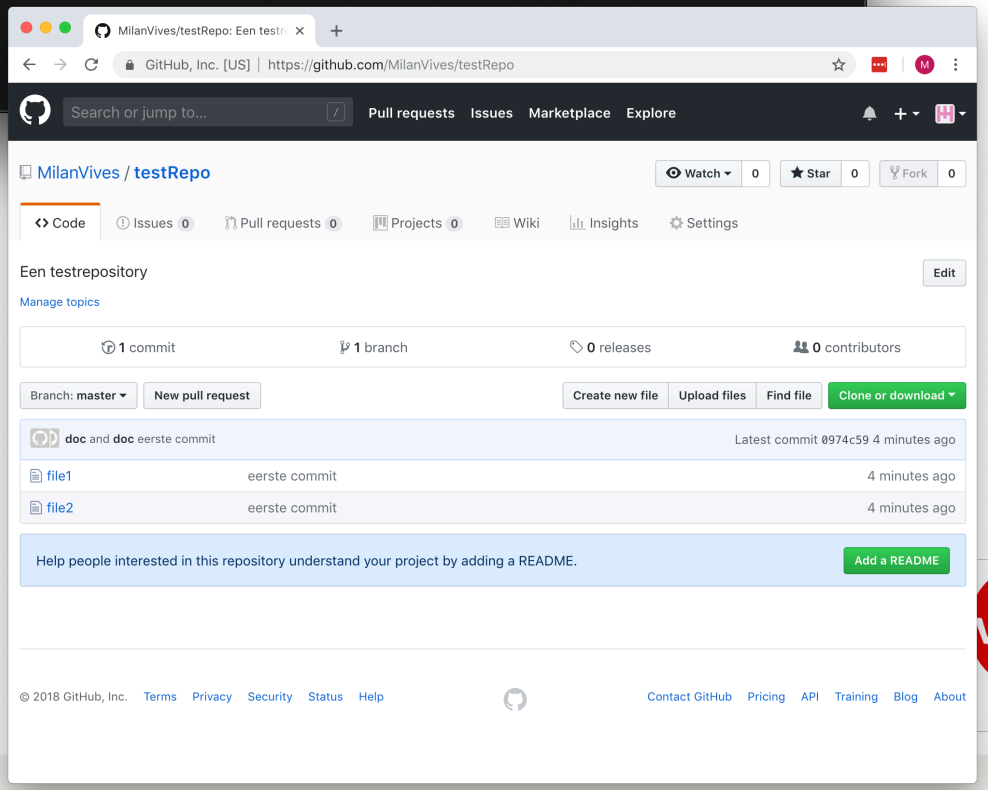
Zelf een repository aanmaken

```
testRepo — -bash — 112×43

milans-MacBook-Pro:~ doc$ mkdir testRepo
milans-MacBook-Pro:~ doc$ cd testRepo/
milans-MacBook-Pro:testRepo doc$ touch file1
milans-MacBook-Pro:testRepo doc$ touch file2
milans-MacBook-Pro:testRepo doc$ git init
 Initialized empty Git repository in /Users/doc/testRepo/.git/
milans-MacBook-Pro:testRepo doc$ git add .
 milans-MacBook-Pro:testRepo doc$ git remote add origin https://github.com/MilanVives/testRepo.git
milans-MacBook-Pro:testRepo doc$
```

```
testRepo — -bash — 112×43

milans-MacBook-Pro:testRepo doc$ git add .
milans-MacBook-Pro:testRepo doc$ git commit -m "eerste commit"
[master (root-commit) 0974c59] eerste commit
 Committer: doc <doc@milans-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1
 create mode 100644 file2
milans-MacBook-Pro:testRepo doc$ git push -u origin master
Username for 'https://github.com': diMilan
Password for 'https://diMilan@github.com':
```

Zelf een repository aanmaken





Zelf een repository aanmaken

Push an existing repository from the command line

```
git remote add origin https://github.com/MilanVives/testRepo.git
K
```

Commands

CREATE

Clone an existing repository

```
$ git clone ssh://user@domain.com/repo.git
```

Create a new local repository

```
$ git init
```

LOCAL CHANGES

Changed files in your working directory

```
$ git status
```

Changes to tracked files

```
$ git diff
```

Add all current changes to the next commit

```
$ git add .
```

Add some changes in <file> to the next commit

```
$ git add -p <file>
```

Commit all local changes in tracked files

```
$ git commit -a
```

-m (message)

Commit previously staged changes

```
$ git commit
```

Change the last commit

Don't amend published commits!

```
$ git commit --amend
```

olieke hogeschool associatie KU Leuven vives

Commands

UPDATE & PUBLISH

List all currently configured remotes

```
$ git remote -v
```

Show information about a remote

```
$ git remote show <remote>
```

Add new remote repository, named <remote>

```
$ git remote add <shortname> <url>
```

Download all changes from <remote>, but don't integrate into HEAD

```
$ git fetch <remote>
```

Download changes and directly merge/integrate into HEAD

```
$ git pull <remote> <branch>
```

Publish local changes on a remote

```
$ git push <remote> <branch>
```

Delete a branch on the remote

```
$ git branch -dr <remote/branch>
```

Publish your tags

```
$ git push --tags
```

MERGE & REBASE

Merge <branch> into your current HEAD

```
$ git merge <branch>
```

Rebase your current HEAD onto <branch>

Don't rebase published commits!

```
$ git rebase <branch>
```

Abort a rebase

```
$ git rebase --abort
```

Continue a rebase after resolving conflicts

```
$ git rebase --continue
```

Use your configured merge tool to solve conflicts

```
$ git mergetool
```

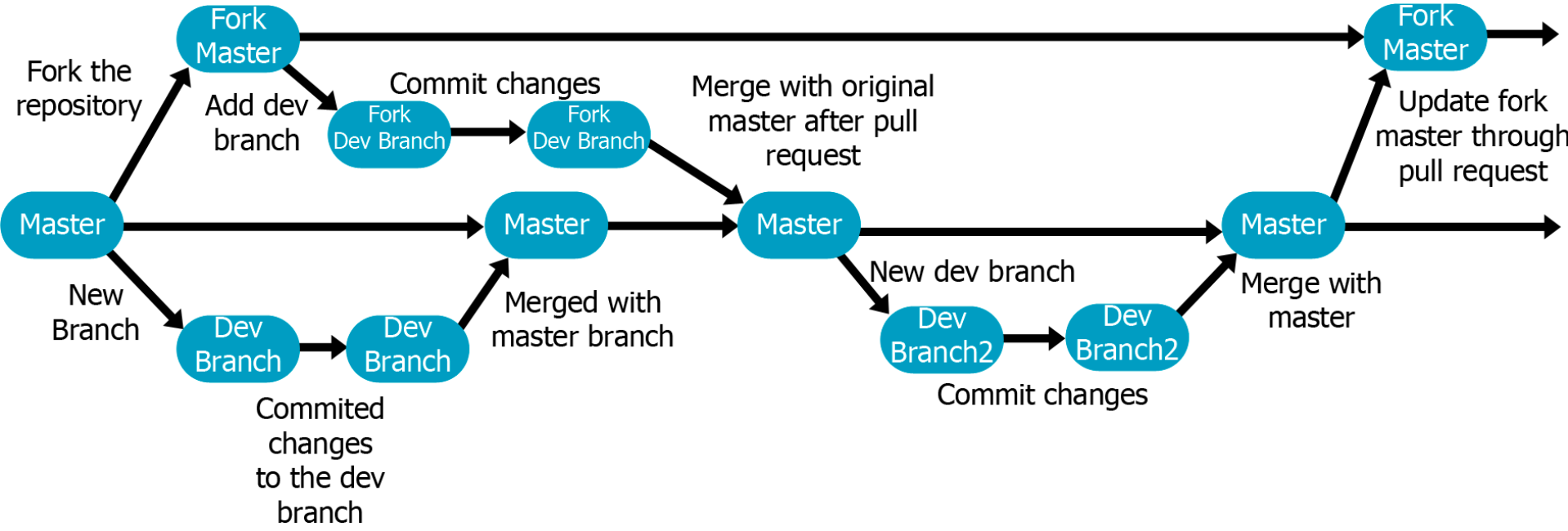
Use your editor to manually solve conflicts and (after resolving) mark file as resolved

```
$ git add <resolved-file>
```

```
$ git rm <resolved-file>
```

katholieke hogeschool associatie KU Leuven vives

Fork - Branch - Pull request - Merge



Bron afbeelding: https://citymayhem.net/thre

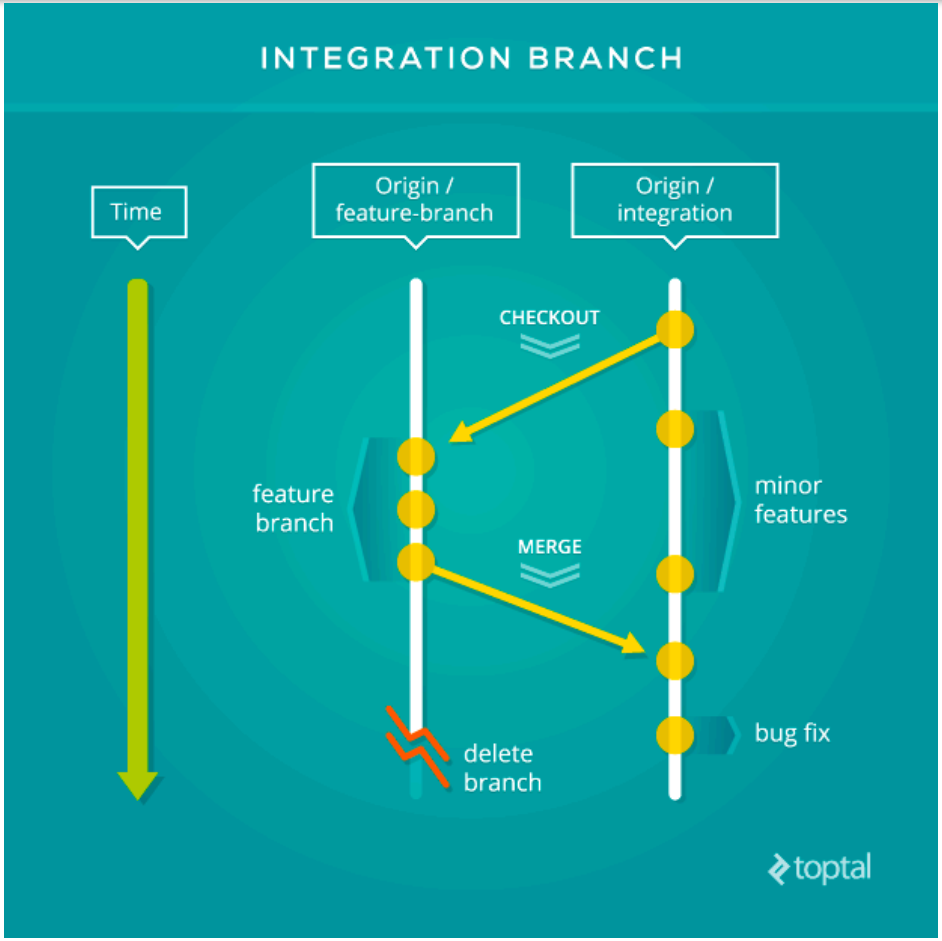
Branch - Merge

- Branch project = (een eigen branch van de repo maken om eigen features aan te maken)

- Je kan veel verschillende branches maken van een Master. Zo kan iedereen aan zijn eigen stuk werken.

- Eenmaal klaar kan je je branch mergen d.w.z. je wijzigingen worden toegevoegd aan de Master-branch.

- Master = default branch aangemaakt door git bij het initialiseren



Fork =/= Branch!

Bron afbeelding: https://www.toptal.com/git/git-workflows-for-pros-a-good-git-guide

Branch - Merge

```
milans-MacBook-Pro:~ mbpt$ mkdir GitTest
milans-MacBook-Pro:~ mbpt$ cd GitTest
milans-MacBook-Pro:GitTest mbpt$ git init
Initialized empty Git repository in /Users/mbpt/GitTest/.git/
milans-MacBook-Pro:GitTest mbpt$ touch file1
milans-MacBook-Pro:GitTest mbpt$ touch file2
milans-MacBook-Pro:GitTest mbpt$ git add .
milans-MacBook-Pro:GitTest mbpt$ git commit -m 'eerste commit'
[master (root-commit) 1efdfb7] eerste commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1
 create mode 100644 file2
milans-MacBook-Pro:GitTest mbpt$ git branch nieuweBranchVoorFixes
milans-MacBook-Pro:GitTest mbpt$ git checkout nieuweBranchVoorFixes
Switched to branch 'nieuweBranchVoorFixes'
milans-MacBook-Pro:GitTest mbpt$ touch file3
milans-MacBook-Pro:GitTest mbpt$ git add .
milans-MacBook-Pro:GitTest mbpt$ git commit -m 'branch commit'
[nieuweBranchVoorFixes 4e5efb2] branch commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file3
milans-MacBook-Pro:GitTest mbpt$ ls
file1 file2 file3
milans-MacBook-Pro:GitTest mbpt$ git checkout master
Switched to branch 'master'
milans-MacBook-Pro:GitTest mbpt$ ls
file1 file2
milans-MacBook-Pro:GitTest mbpt$
```

Branch - Merge

```
milans-MacBook-Pro:GitTest mbpt$ git merge nieuweBranchVoorFixes
Updating 1efdfb7..4e5efb2
Fast-forward
 file3 | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file3
milans-MacBook-Pro:GitTest mbpt$ ls
file1  file2  file3
milans-MacBook-Pro:GitTest mbpt$
```

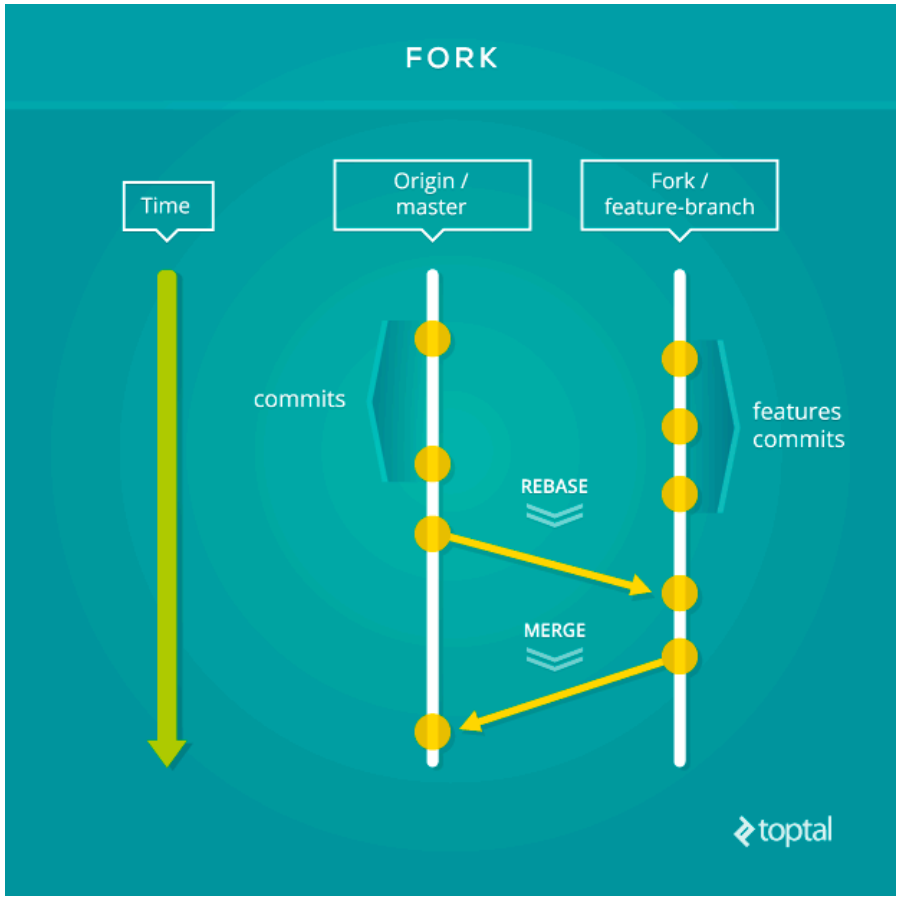
katholieke hogeschool associatie KU Leuven vives

Fork - Pull request

- Fork project = (een eigen kopie van de repo maken om eigen wijzigingen Aan te brengen)

- Later kan je eventueel een pull-request aanvragen (wil je mijn wijzigingen aan de Master repo toevoegen?)

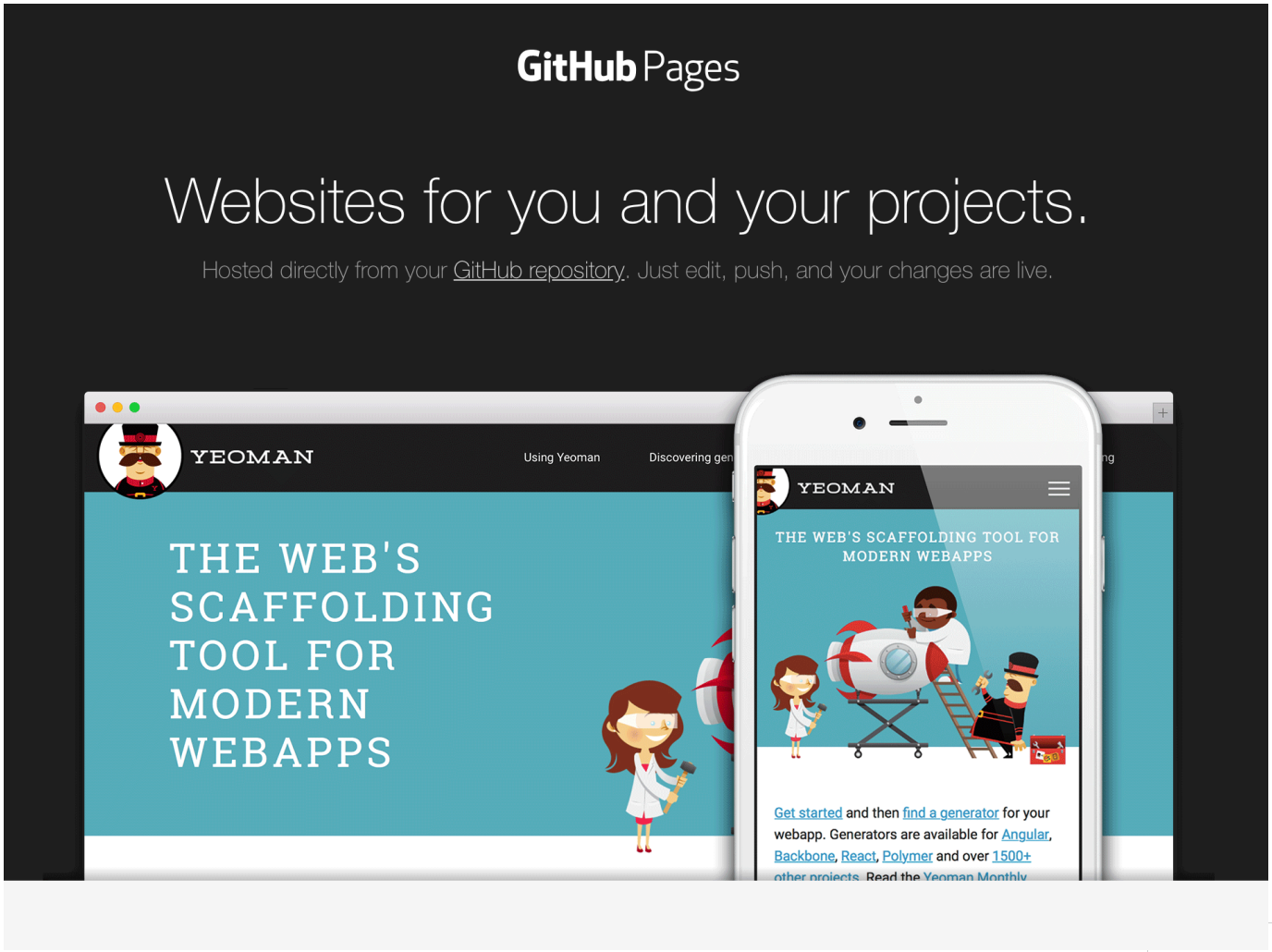
- Als de pull request aanvaard wordt wordt de Master aangepast met de laatste wijzigingen



Fork =/= Clone!

Bron afbeelding: https://www.toptal.com/git/git-workflows-f

Website Deployment - GitHub Pages



https://pages.github.com/

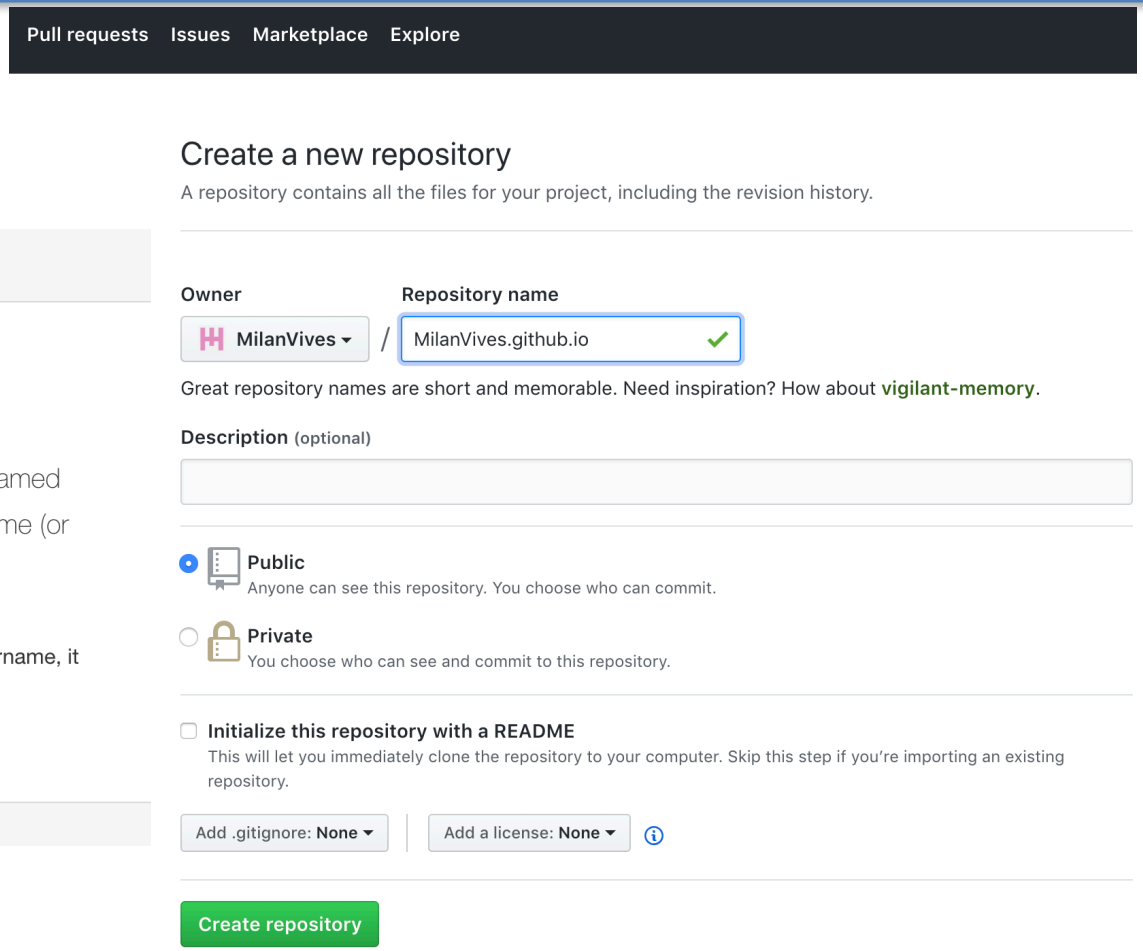
Website Deployment - GitHub Pages

1

Create a repository

Head over to GitHub and create a new repository named *username*.github.io, where *username* is your username (or organization name) on GitHub.

If the first part of the repository doesn't exactly match your username, it won't work, so make sure to get it right.



Website Deployment - GitHub Pages

Clone the repository

Go to the folder where you want to store your project, and clone the new repository:

```
$ git clone https://github.com/MilanVives/MilanVives.github.io.git
```

Hello World

Enter the project folder and add an index.html file:

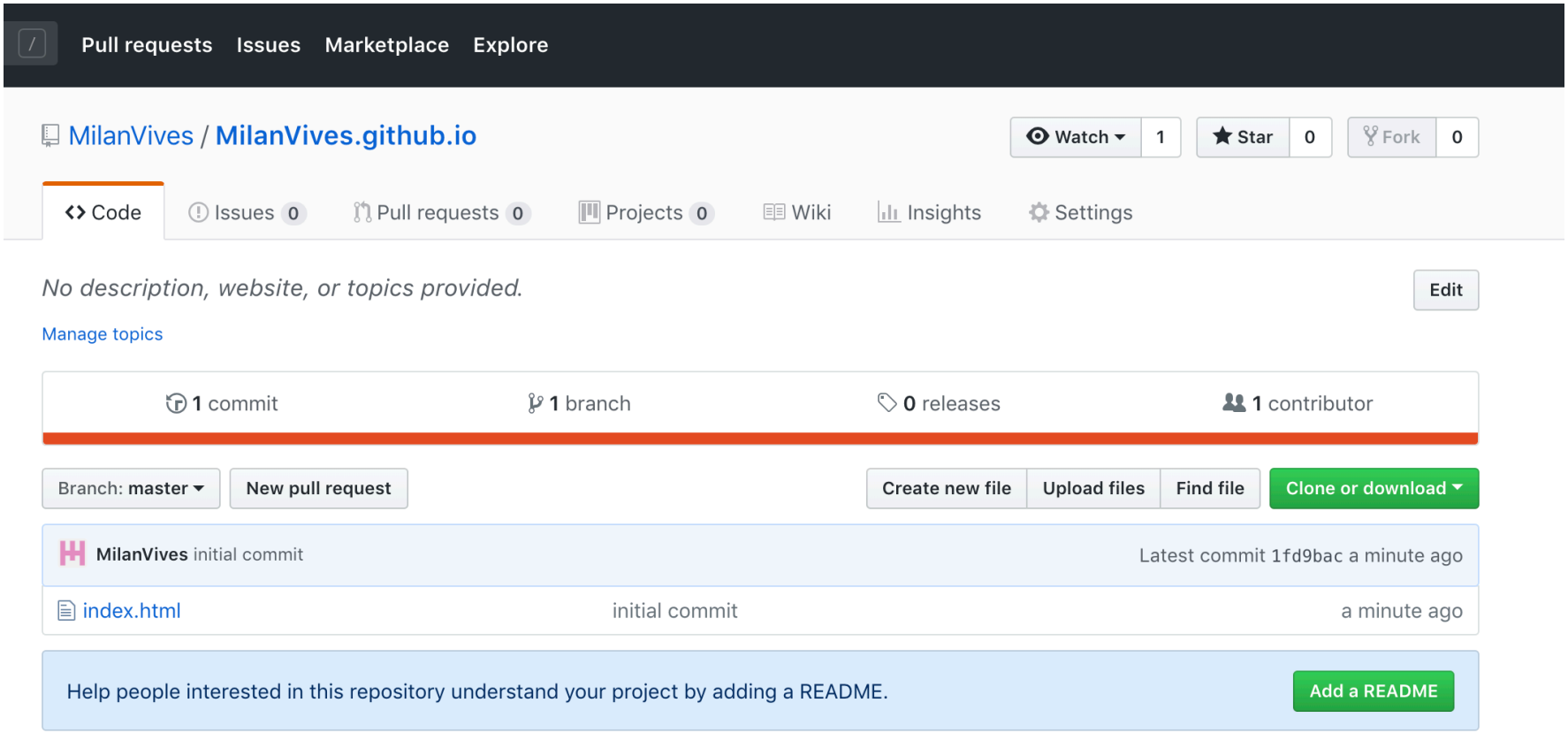
```
$ cd MilanVives.github.io/
$ echo "Hello World" > index.html
```

Push it

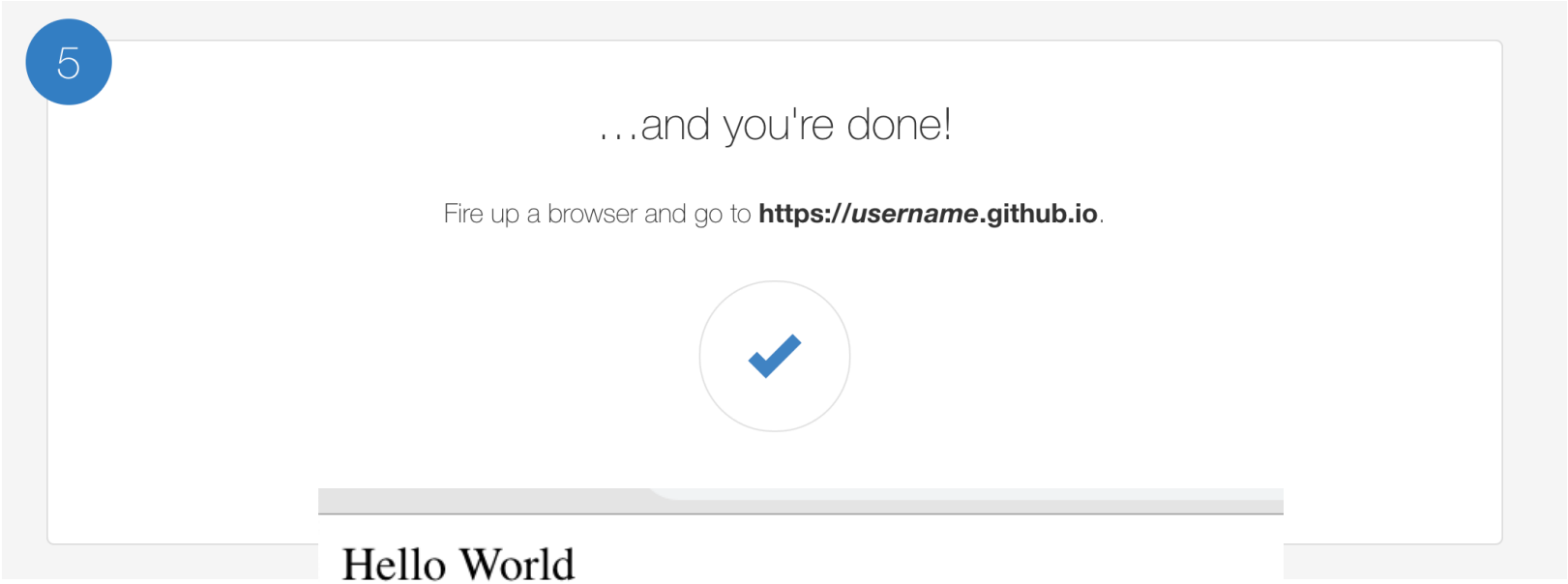
Add, commit, and push your changes:

```
$ git add --all
$ git commit -m 'initial commit'
$ git push -u origin master
```

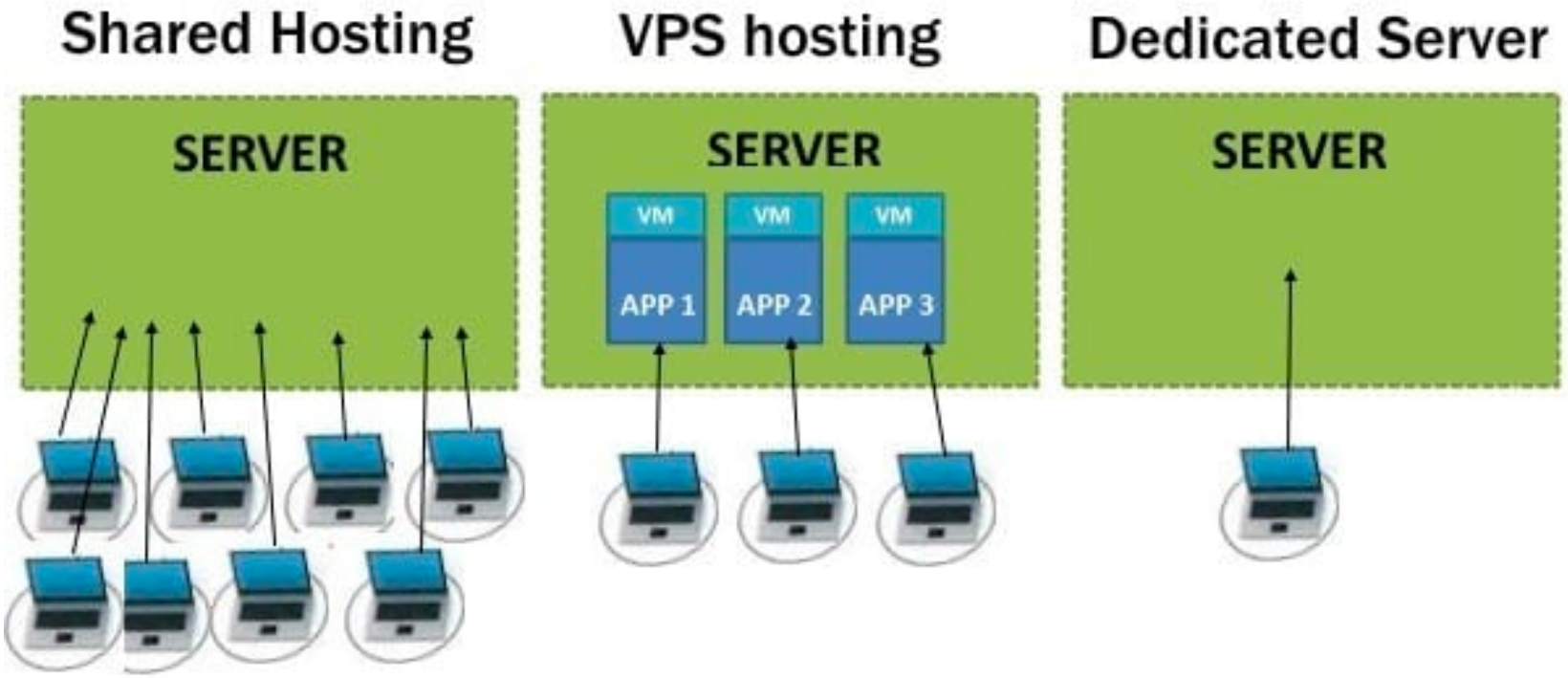
Website Deployment - GitHub Pages



Website Deployment - GitHub Pages



Website Deployment - Andere



Website Deployment - Andere

1. Register Domain name - Nog vrij?

   https://www.dnsbelgium.be/nl

   https://whois.eurid.eu/en/

   https://whois.icann.org/en

   https://www.eurodns.com/whois-search/eu-domain-name

2. Vind webserver hosting

   - eigen (ISP - poorten?)
   - Online (gratis telenet, InfinityFree, Freehostia ….)
   - Online betalend: Transip, Versio, Mijnhostingpartner, godaddy.com, 000webhost.com, …

3. Link en upload website

   - Drag and drop
   - FTP
   - Website form
   - …

Oefeningen

Permanente Evaluatie: zie Toledo

1. Registreer bij GitHub https://github.com/
    1. Maak een private repository aan met als naam het “Internettechnologie”
    2. Maak een map op je PC met alle slides van de lessen internettechnologie. Respecteer de mappenstructuur. Push alle bestanden naar je GitHub repo. Plak de link van je Repo en screenshots in je taak
    3. Gebruik bij het committen een passende commit boodschap
2. Registreer op een website naar keuze een gratis hostingpakket
    1. Verbind via FTP met je hostingserver en upload een beperkte HTML website. Bezorg me de website link en screenshots van de FTP verbinding.
    2. Zoek via Wireshark hoe de FTP connectie gemaakt wordt (screenshot)
3. Leg me uit in enkele zinnen wat de volgende softwarepakketten doen:
    - Cpanel
    - Directadmin
    - Plesk
4. Zoek zoveel mogelijk informatie over de website www.vives.be. (IP adres, wie is de eigenaar, waar wordt het gehost, enz…)

katholieke hogeschool associatie KU Leuven vives