

Versioning & Deployment

Les 10 20181204 - 06

M. DIMA





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](#)



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 ingekijken 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>



Repositories

The <https://www.freecodecamp.org> open source codebase and curriculum. Learn to code for free together with millions of people.

learn-to-code nonprofits programming nodejs react d3 careers education teachers javascript certification curriculum
math community

19,164 commits 4 branches 0 releases 2,310 contributors BSD-3-Clause

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

Latest commit 12ad9eb 30 minutes ago
2 days ago
22 hours ago
30 minutes ago
4 days ago
35 minutes ago
2 days ago
13 hours ago
7 days ago
4 days ago
4 years ago
8 months ago



Repositories

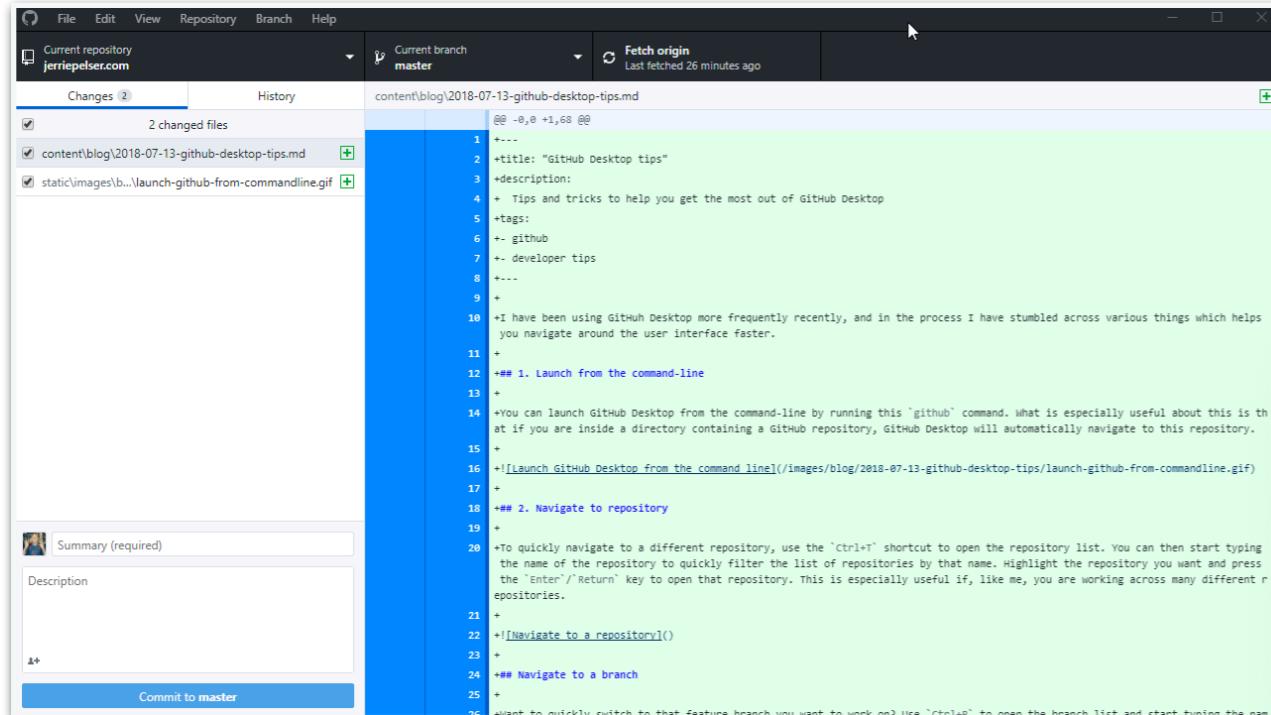
The screenshot shows the GitHub repository page for `freeCodeCamp/freeCodeCamp`. The page displays various metrics: 19,164 commits, 4 branches, 0 releases, and 2,310 contributors. A prominent red circle highlights the **Clone or download** button and its associated modal window. The modal window contains the URL <https://github.com/freeCodeCamp/freeCodeCamp>, options to **Open in Desktop** or **Download ZIP**, and links for **Clone with HTTPS** and **Use SSH**.

The repository page also features a list of recent commits:

- Cyb3rN4u7 and raisedadead fix: site width is increasing for no reason (#34313)
- .github Update CODEOWNERS (#30280)
- api-server fix: Handle no user request
- client fix: site width is increasing for no reason (#34313)
- config Feat: News in the client app (#34392)
- curriculum fix: added consistency to capitalization and punctuation (#29201)
- docs fix: Broken Links and Formatting (#32171)
- guide Updated "More Information" link from 4.0 to 4.1 (#24027)
- mock-guide Add PHP data types article (#25508)
- tools chore(security): Update packages to remove security issue flagged by...
- .editorconfig Add .editorconfig to enforce certain automatic behavior on all editors,
- .eslintignore feat(seed): "unpack" and "repack" scripts
- .eslintrc feat(prop for modern challenges) (#15791)



Desktop VS CLI



```
doc — -bash — 80x44
[milans-MacBook-Pro:~ doc$ git --help
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv        Move or rename a file, a directory, or a symlink
  reset     Reset current HEAD to the specified state
  rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  branch   List, create, or delete branches
  checkout Switch branches or restore working tree files
  commit   Record changes to the repository
  diff     Show changes between commits, commit and working tree, etc
  merge    Join two or more development histories together
  rebase   Reapply commits on top of another base tip
  tag      Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch    Download objects and refs from another repository
  pull     Fetch from and integrate with another repository or a local branch
  push     Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
milans-MacBook-Pro:~ doc$ ]
```



Install Git

Windows: <https://central.github.com/deployments/desktop/desktop/latest/win32>

Mac: <https://central.github.com/deployments/desktop/desktop/latest/darwin>

Or \$ brew install git

Linux: sudo apt-get install git

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

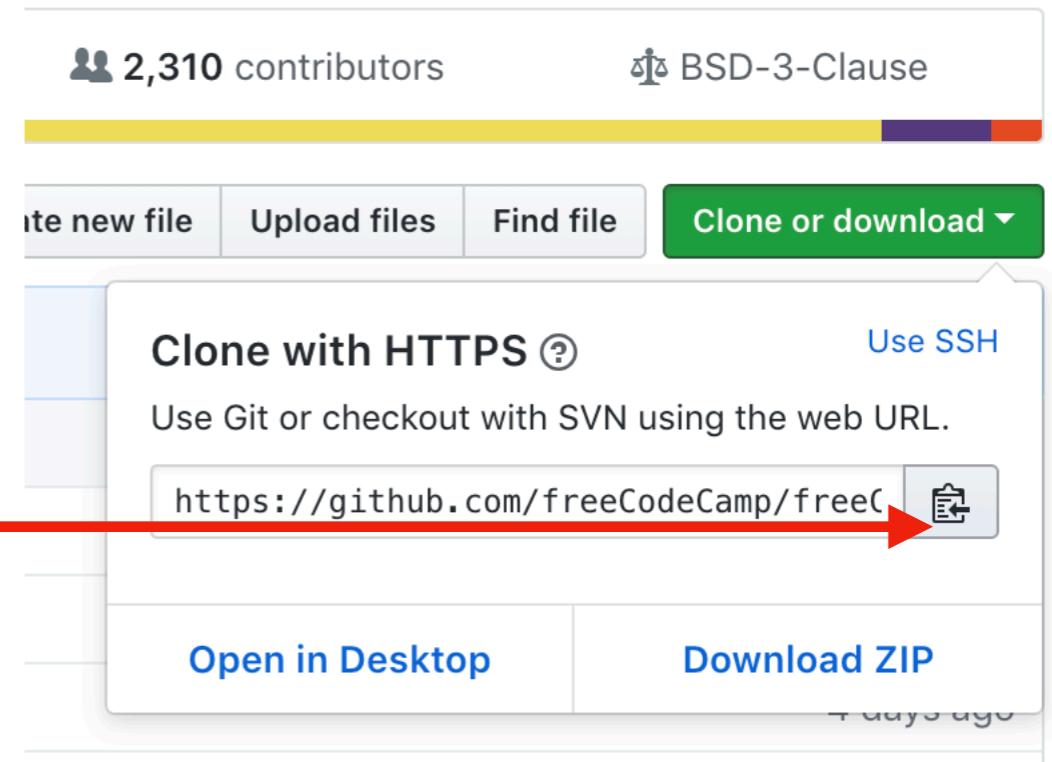


Git clone

= lokaal een kopie maken van een online repo
bv. Alles van freecode camp 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



```
[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$
```

Zelf een repository aanmaken

A screenshot of a web browser displaying the GitHub homepage. The title bar shows 'GitHub' and the URL 'GitHub, Inc. [US] | https://github.com'. The main content area features a large green banner with the text 'Learn Git and GitHub without any code!' and a description below it. At the bottom left, there's a sidebar with a message about updated terms and privacy, a 'Repositories' section showing none, and a prominent green 'New repository' button. A large red arrow points from the top left towards this 'New repository' button. To the right, there are sections for 'Browse activity' and 'Discover repositories', with a call-to-action 'Explore GitHub'. The bottom navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', 'Explore', and user notifications. A yellow 'vives' logo is visible in the bottom right corner.

Our new Terms of Service and Privacy Statement are in effect.

Repositories

New repository

You don't have any repositories yet!

Read the guide

Start a project

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

GitHubDesktop...exe Cancelled

Show All

Zelf een repository aanmaken

The screenshot shows the GitHub 'Create a New Repository' interface. At the top, the title 'Create a New Repository' is visible, along with the GitHub logo and navigation links for Pull requests, Issues, Marketplace, and Explore.

Create a new repository
A repository contains all the files for your project, including the revision history.

Owner: MilanVives / **Repository name**: testRepo

Great repository names are short and memorable. Need inspiration? How about [verbose-octo-broccoli](#).

Description (optional): Een testrepository

Visibility Options:

- Public**: Anyone can see this repository. You choose who can commit.
- Private**: You choose who can see and commit to this repository.

Initialization Options:

- Initialize this repository with a README**: This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Red arrows point from the bottom right towards the repository name input field, the visibility radio buttons, and the 'Create repository' button.



Zelf een repository aanmaken

A screenshot of a GitHub repository page for 'MilanVives/testRepo'. The page shows basic repository statistics: 0 pull requests, 0 issues, 0 projects, 0 wiki pages, 0 insights, and 0 settings. A red arrow points from the top left towards the repository name 'testRepo' in the header. Another red arrow points from the bottom left towards the 'HTTPS' link in the 'Quick setup' section. A large red circle highlights the '...or create a new repository on the command line' section, which contains the following command-line steps:

```
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
```

Below this, another red circle highlights the '...or push an existing repository from the command line' section, which contains the following command-line steps:

```
git remote add origin https://github.com/MilanVives/testRepo.git
git push -u origin master
```

The bottom of the page features a yellow footer bar with the text 'VIVES H' on the left and a yellow balloon icon with the word 'vives' on the right.

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 "commitbericht" // wijzigingen committen
5. git push -u origin master // push alles



Zelf een repository aanmaken

```
[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$
```

```
[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

The screenshot illustrates the workflow for creating a new GitHub repository. It consists of two main parts: a terminal window at the top and a browser window below it.

Terminal Window:

```
[milans-MacBook-Pro:testRepo doc$ git push -u origin master
Username for 'https://github.com': milanVives
[Password for 'https://milanVives@github.com']:
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 218 bytes | 109.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/MilanVives/testRepo/pull/new/master
remote:
To https://github.com/MilanVives/testRepo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
milans-MacBook-Pro:testRepo doc$ ]
```

Browser Window:

The browser window shows the GitHub repository page for "MilanVives / testRepo". The page title is "MilanVives/testRepo: Een testrepository". The repository summary indicates 1 commit, 1 branch, 0 releases, and 0 contributors. The latest commit is "doc and doc eerste commit" by "milanVives" 4 minutes ago. Below the commit list is a note: "Help people interested in this repository understand your project by adding a README." with a "Add a README" button.

Bottom Right Corner:



A yellow balloon-shaped icon with the word "vives" written on it in white.

Zelf een repository aanmaken

Push an existing repository from the command line

```
git remote add origin https://github.com/MilanVives/testRepo.git  
git push -u origin master
```



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
```

Commit previously staged changes

```
$ git commit
```

-m (message)

Change the last commit

Don't amend published commits!

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

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
```

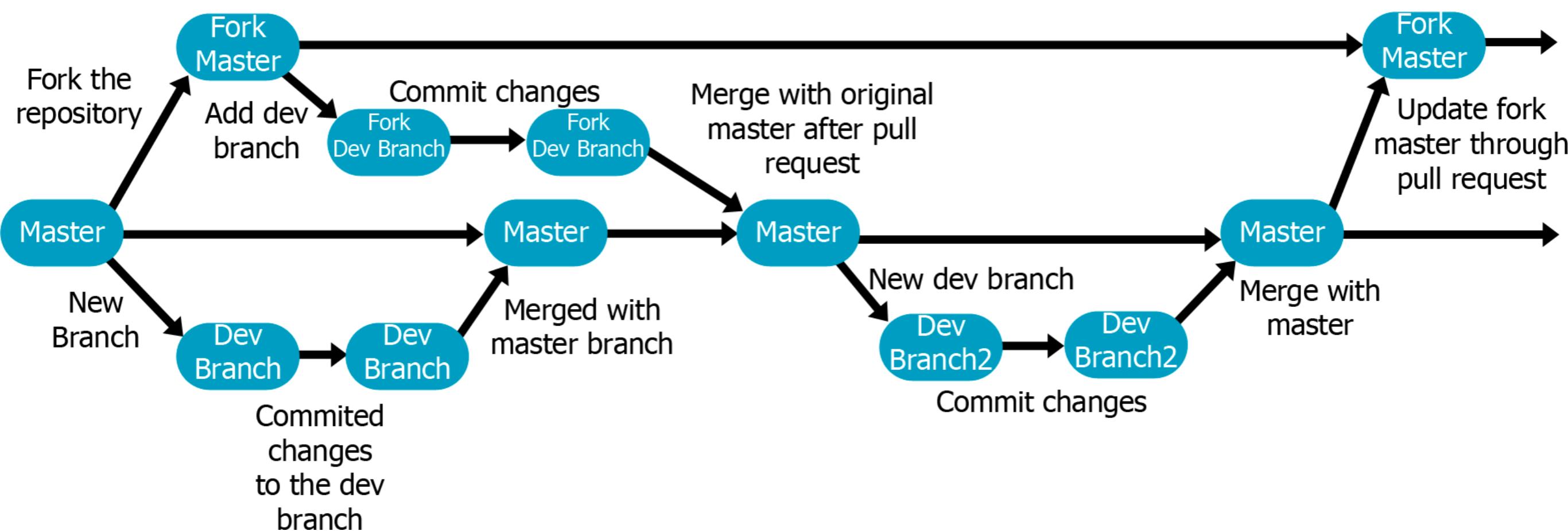
Use your editor to manually solve conflicts
and (after resolving) mark file as resolved

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

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



Fork - Branch - Pull request - Merge

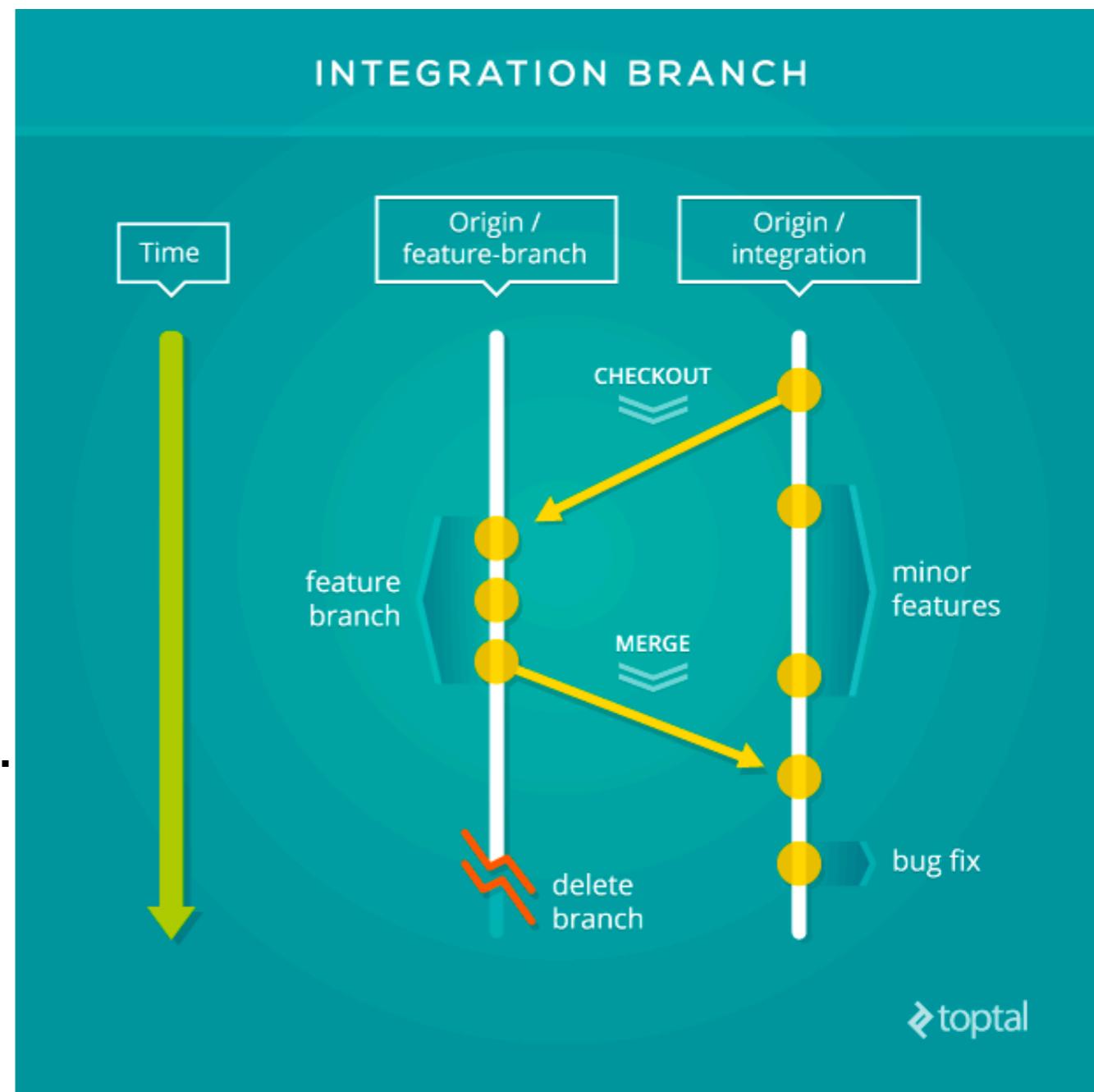


Bron afbeelding: <https://citymayhem.net/thread-4577.html>



Branch - Merge

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



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
file1file2file3
milans-MacBook-Pro:GitTest mbpt$ git checkout master
Switched to branch 'master'
milans-MacBook-Pro:GitTest mbpt$ ls
file1file2
milans-MacBook-Pro:GitTest mbpt$
```



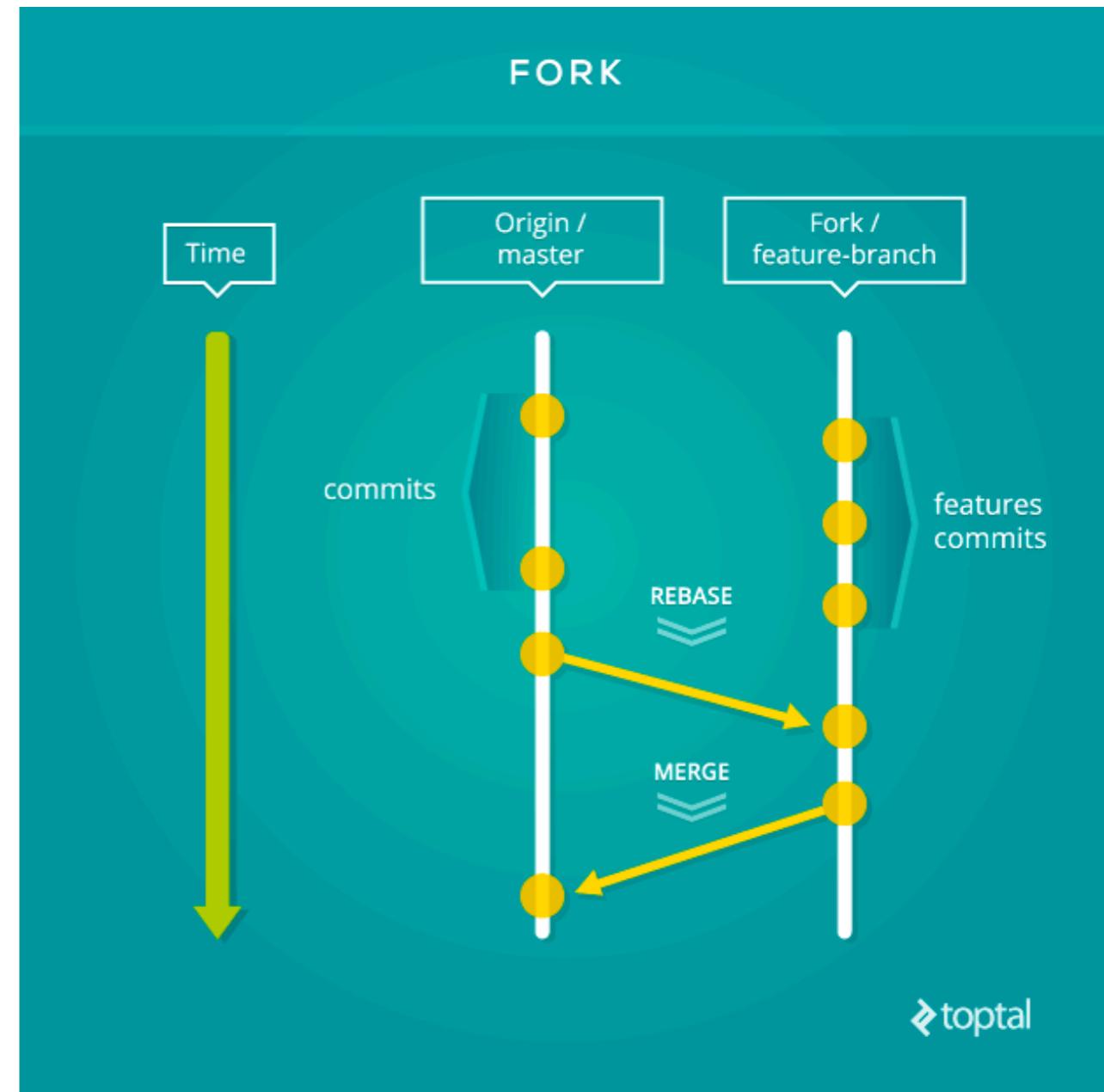
Branch - Merge

```
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$
```



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

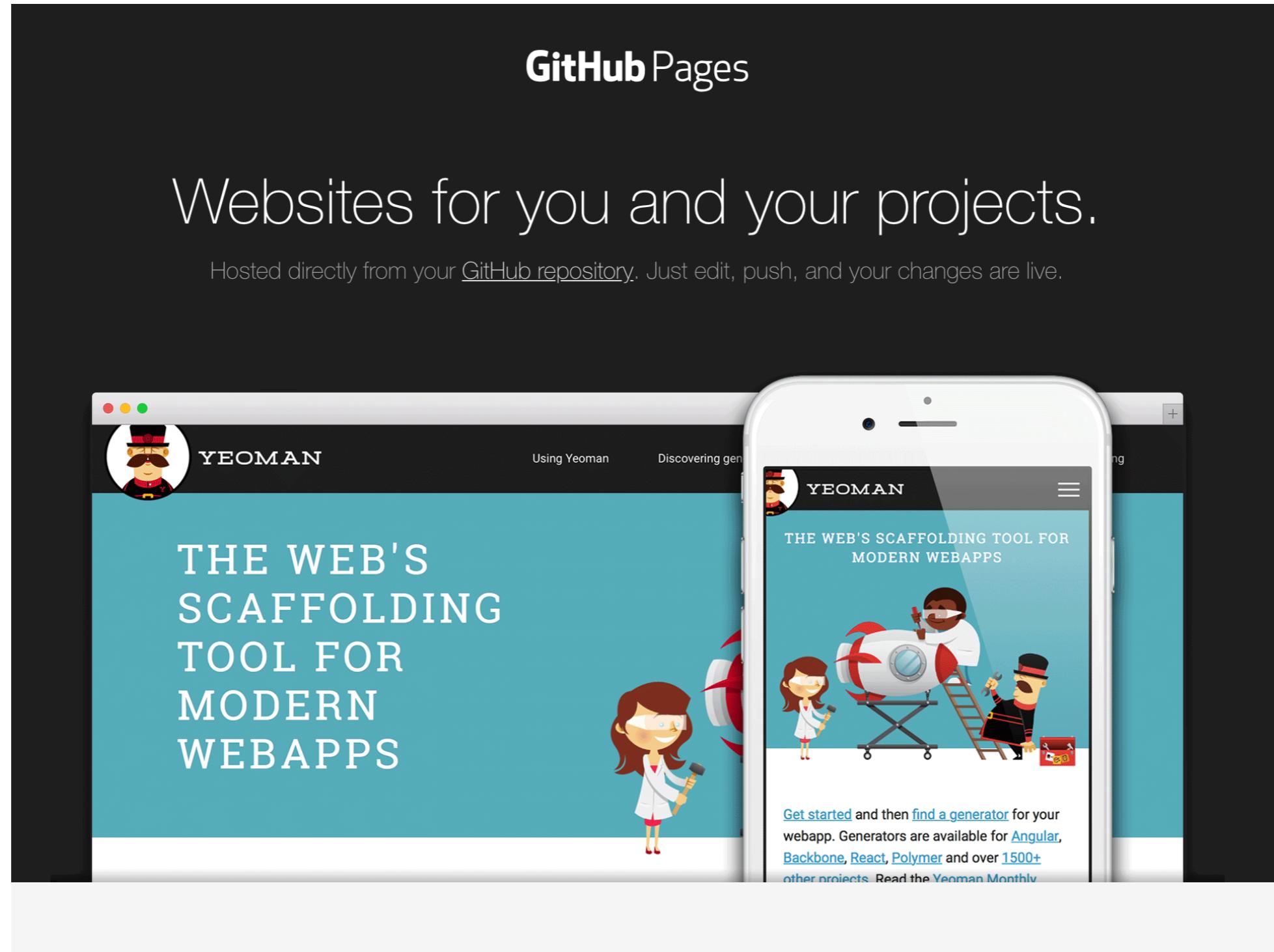


Fork != Clone!

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



Website Deployment - GitHub Pages



<https://pages.github.com/>



Website Deployment - GitHub Pages

Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

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.

Owner

 MilanVives ▾

Repository name

MilanVives.github.io



Great repository names are short and memorable. Need inspiration? How about [vigilant-memory](#).

Description (optional)

 Public

Anyone can see this repository. You choose who can commit.

 Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾

Add a license: None ▾



Create repository



Website Deployment - GitHub Pages

2

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
```

3

Hello World

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

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

4

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

Pull requests Issues Marketplace Explore

MilanVives / MilanVives.github.io

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

MilanVives initial commit Latest commit 1fd9bac a minute ago

index.html initial commit a minute ago

Help people interested in this repository understand your project by adding a README. Add a README

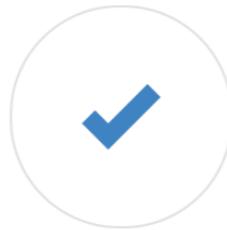


Website Deployment - GitHub Pages

5

...and you're done!

Fire up a browser and go to **<https://username.github.io>**.

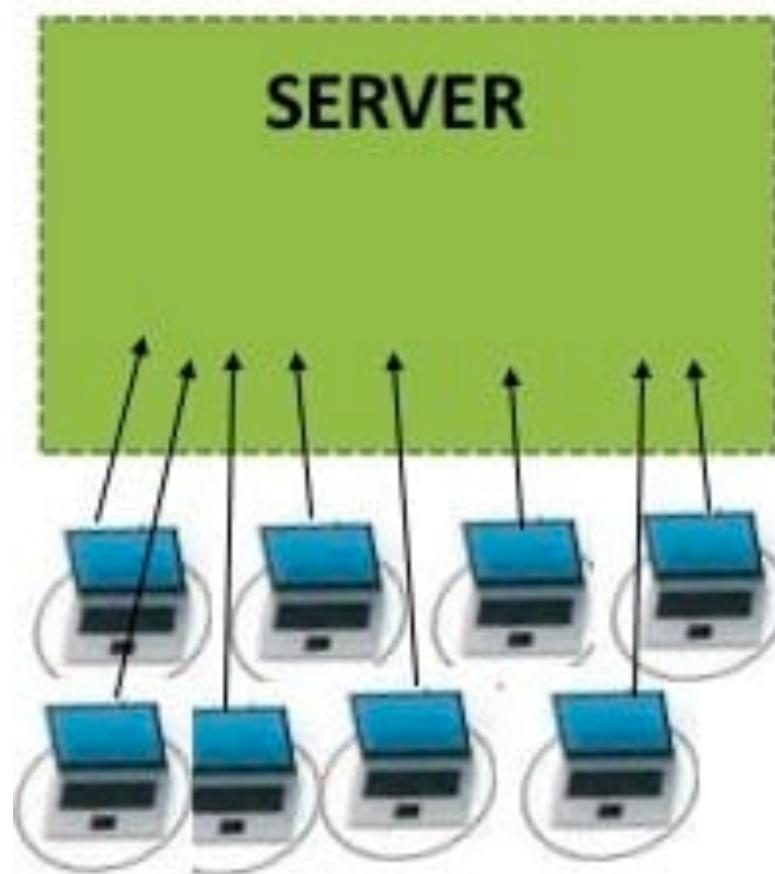


Hello World

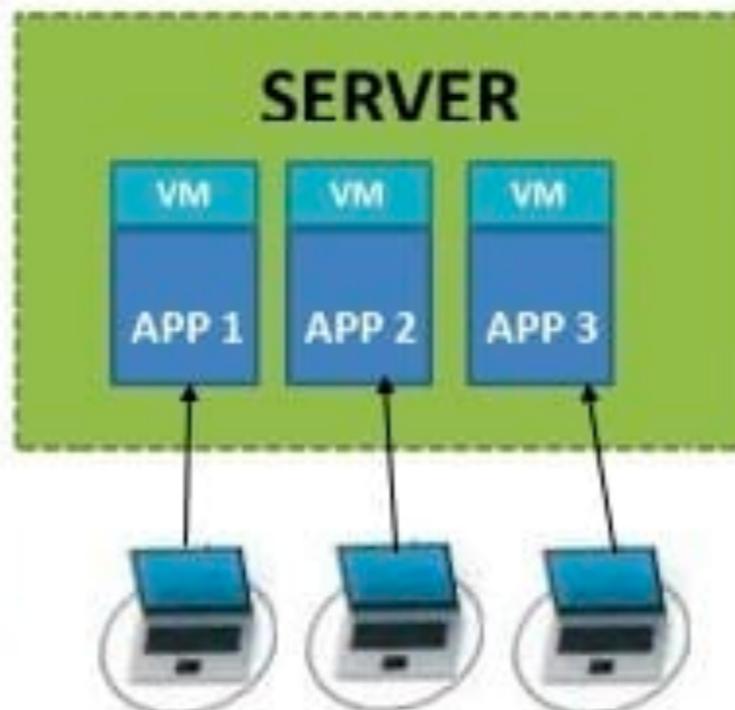


Website Deployment - Andere

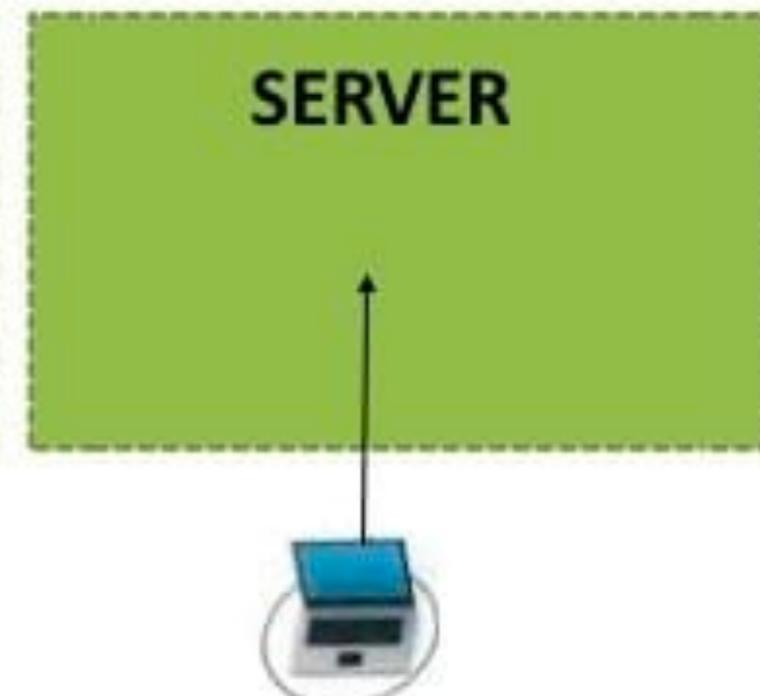
Shared Hosting



VPS hosting



Dedicated Server



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)

3. Link en upload website



Oefeningen

Permanente Evaluatie: indienen voor 13/12/2018 8:30 via smartschool in de juiste uploadmap

1. Registreer bij GitHub <https://github.com/>
2. Maak een private repository aan met als naam het “CursusC4”
3. 2B. Maak een map op je PC met alle slides van de lessen internettechnologie. Respecteer de mappenstructuur. Push alle bestanden naar je GitHub repo. Bezorg me de link van je Repo en screenshots van
4. 2C. Gebruik bij het committen een passende commit boodschap
5. Registreer naar een website van je keuze voor een gratis hostingpakket
6. 5B Verbind via FTP met je hostingserver en upload een beperkte HTML website. Bezorg me de website link en screenshots van de FTP verbinding.
7. Zoek via Wireshark hoe de FTP connectie gemaakt wordt (screenshot)
8. Leg me uit in enkele zinnen wat de volgende softwarepakketten doen:
 - Cpanel
 - Directadmin
 - Plesk
9. Zoek zoveel mogelijk informatie over de website www.miras.be. (IP adres, wie is de eigenaar, waar wordt het gehost, enz...)

