Gitting Started

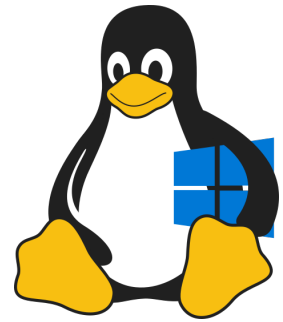
Practical for the git course

K. Müller[1] J. Verzijden[2]

[1]MasterCLASS
E.T.S.V. Scintilla

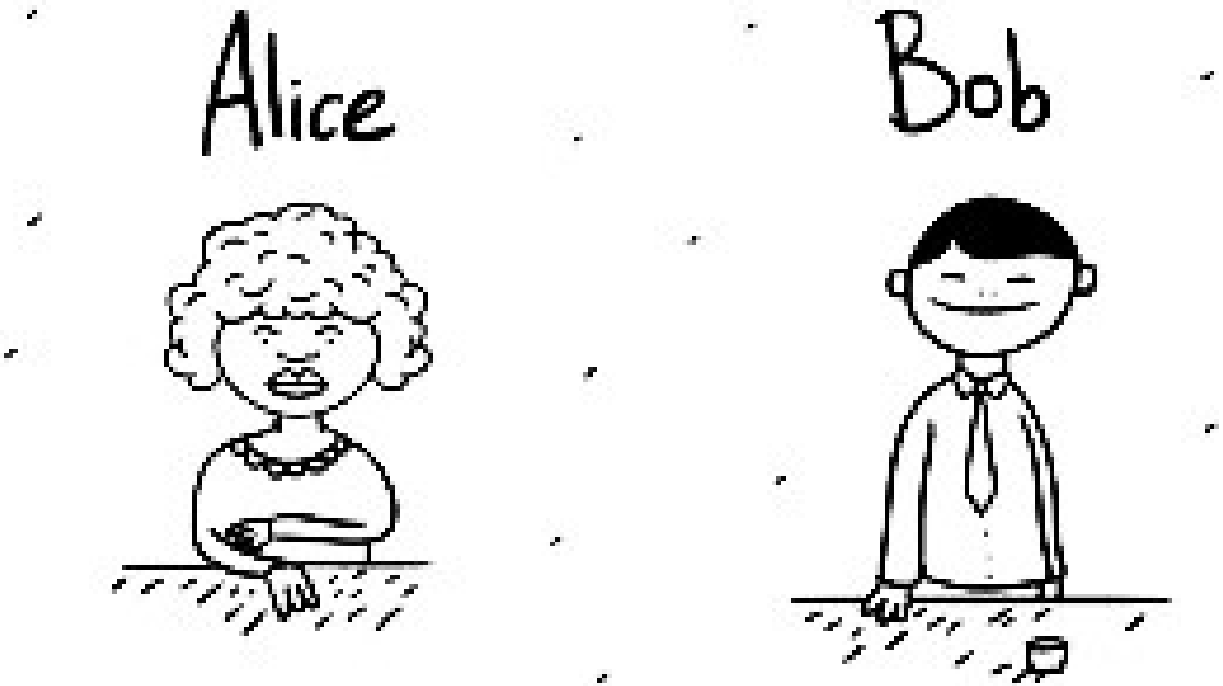
[2]Scintilla Operator Team
E.T.S.V. Scintilla

June 27, 2023



Let's imagine...

Alice and Bob (you and your friend) want to make a portfolio website.



1 - Prerequisite: Installing the CLI

- Windows
  - Using *winget*:

    ```
    winget install --id Git.Git -e --source winget
    ```
  - Using an installer or portable version:
    https://git-scm.com/downloads/win
- Linux
  - Don't, you already have it (-:
  - Through your package manager:
    - `apt install git`
    - `dnf install git`
    - `pacman -S git`
- MacOS
  - Using Homebrew:
    - `brew install git`
  - Install Xcode, that ships with Git

2 - Installation successful?

```
johanv@my-machine: ~$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <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
   restore   Restore working tree files
   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
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
```

3 - Configuration

How:

- Using an editor:
  `git config --global --edit`
- Using commands:
  `git config --global user.name 'Johan Verzijden'`

What:

- Name (*user.name*)
- Email address (*user.email*)
- Default branch name (*init.defaultBranch*)
- Fast-forward (*pull.ff*)

Read `git help config` or `man git config` for all configuration options

4 - Starting a new project

Make a new directory and do a git init.

```
~$ mkdir my-portfolio
~$ cd my-portfolio
~/my-portfolio$ git init
```

5 - Making changes

Make some basic files, for instance, index.html:

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="style.css">
        <title>Portfolio of Johan</title>
    </head>

    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

5 - Making changes

And also style.css:

```
body {
    background-color: slategray;
}
h1 {
    text-align: center;
}
```

6 - Do your first commits

```
~/my-portfolio$ git add index.html style.
    css
~/my-portfolio$ git commit -m 'Initial
    commit'
```

7 - Making changes (second commit)

Add some welcome text in your index.html:

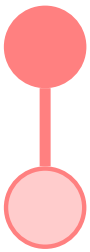
```
<!DOCTYPE html>
<html>
    <head>
    <link rel="stylesheet" href="style.css">
    <title>Portfolio of Johan</title>
    </head>

    <body>
    <h1>Johan Verzijden</h1>
    <p>
    Welcome to my portfolio. Here you can find my
        projects.
    </p>
    </body>
</html>
```

8 - Do your second commit

```
~/my-portfolio$ git commit -am 'Add
    heading and welcome text'
# See the current git log:
~/my-portfolio$ git log
```

9 - Create a new branch and switch to it



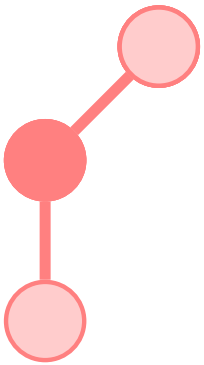
```
~/my-portfolio$ git switch -c add-git-
    project-info
#### Alternatives
~/my-portfolio$ git branch add-git-
    project-info
~/my-portfolio$ git checkout add-git-
    project-info
```

10 - Make change in the seperate branch

Adding something to the index.html:

```
<div class="projects">
    <div class="project-1">
        <h2>Learning about Git</h2>
        <p>Tonight I was at a great course given
           by two members of E.T.S.V. Scintilla,
           in which I learned how to use Git for
           my projects.</p>
    </div>
</div>
```

11 - Commit and switch back



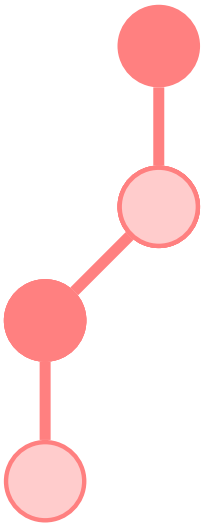
```
# Add info about your latest project
~/my-portfolio$ git commit -am 'Add info
    about latest project'
```

12 - Make another change

Adding something to the index.html:

```
<div class="projects">
    <div class="project-1">
        <h2>Learning about Git</h2>
        <p>Tonight I was at a great course given
           by two members of E.T.S.V. Scintilla,
           in which I learned how to use Git for
           my projects.</p>
    </div>
    <div class="project-2">
        <h2>Learning about Git</h2>
        <p>Tonight safat for my projects.</p>
    </div>
</div>
```

13 - Commit and reset



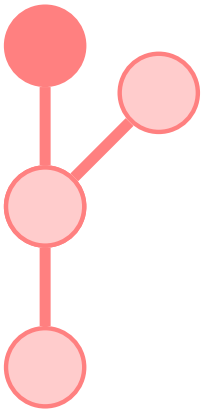
```
# Add info about your latest project
~/my-portfolio$ git commit -am 'Add
    info about latest project'
# Add info about another project
~/my-portfolio$ git commit -am 'Add
    info about another project'
# Oh wait, I don't want this in this
    branch. Let's go back
~/my-portfolio$ git log
~/my-portfolio$ git reset <previous
    commit hash>
~/my-portfolio$ git switch main
```

14 - Another change on the main branch

Adding something to the index.html:

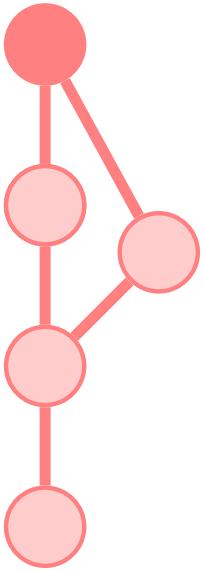
```
<div class="contact-details">
    <p>If you have any questions about my
       projects, you can contact me on my email
       address</p>
</div>
```

15 - Commit on main branch



```
# Add contact details
~/my-portfolio$ git commit -am 'Add
    contact details'
```

16 - Merge the two branches



```
~/my-portfolio$ git merge add-git-project
    -info
# Merge conflict!
```

17 - Resolve the merge conflict

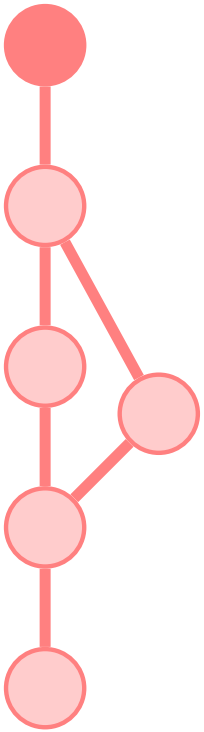
```
# Go into the file and resolve the conflict
~/my-portfolio$ git add index.html
~/my-portfolio$ git merge --continue
```

18 - Change some color

Adding something to the style.css:

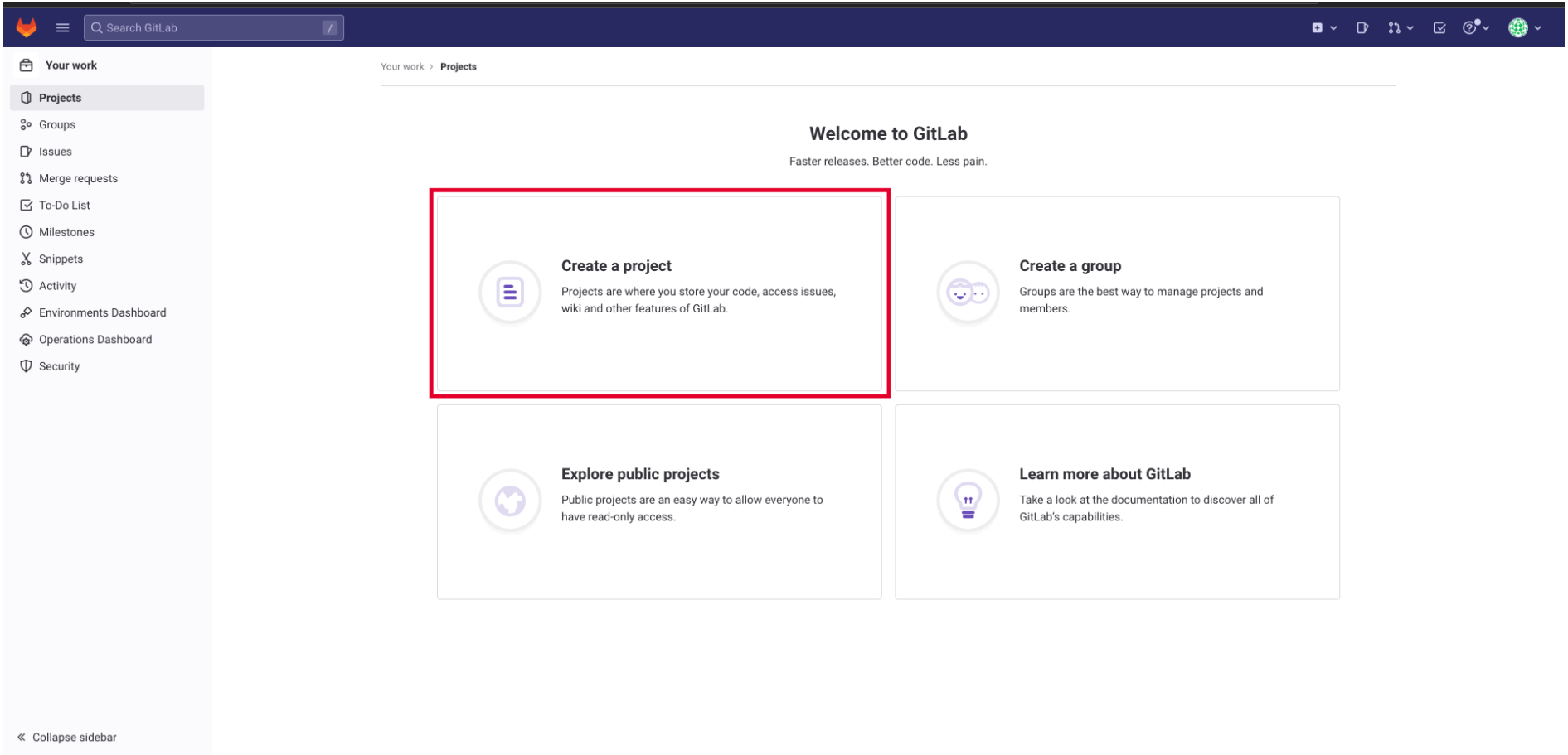
```
body {
    color: slategray;
    background-color: slategray;
}
```

19 - Change the text colour and commit

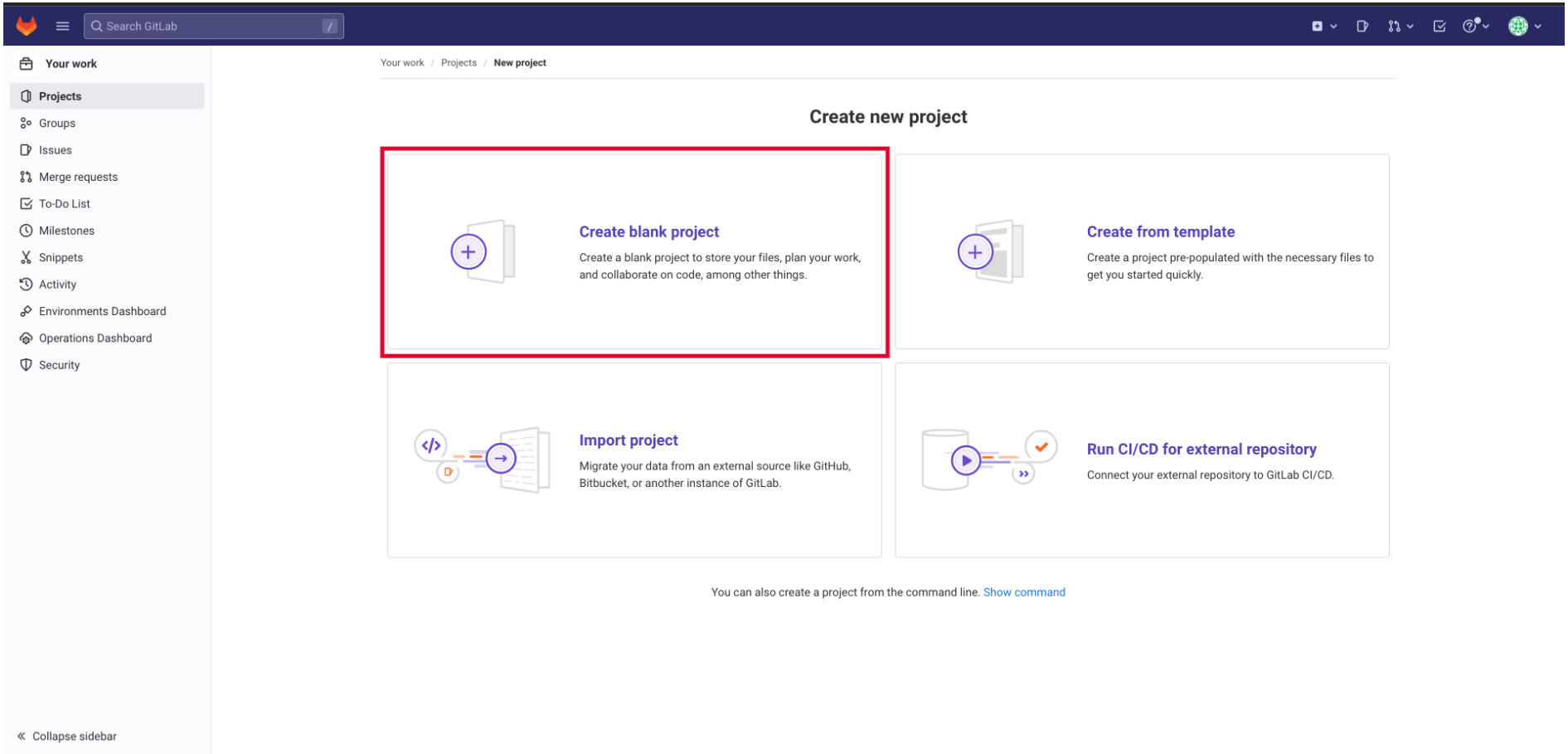


```
# Make the text and background colour the
    same
# (This is on purpose, will become clear
   later)
~/my-portfolio$ git commit -am 'Give the
   text a great colour'
```

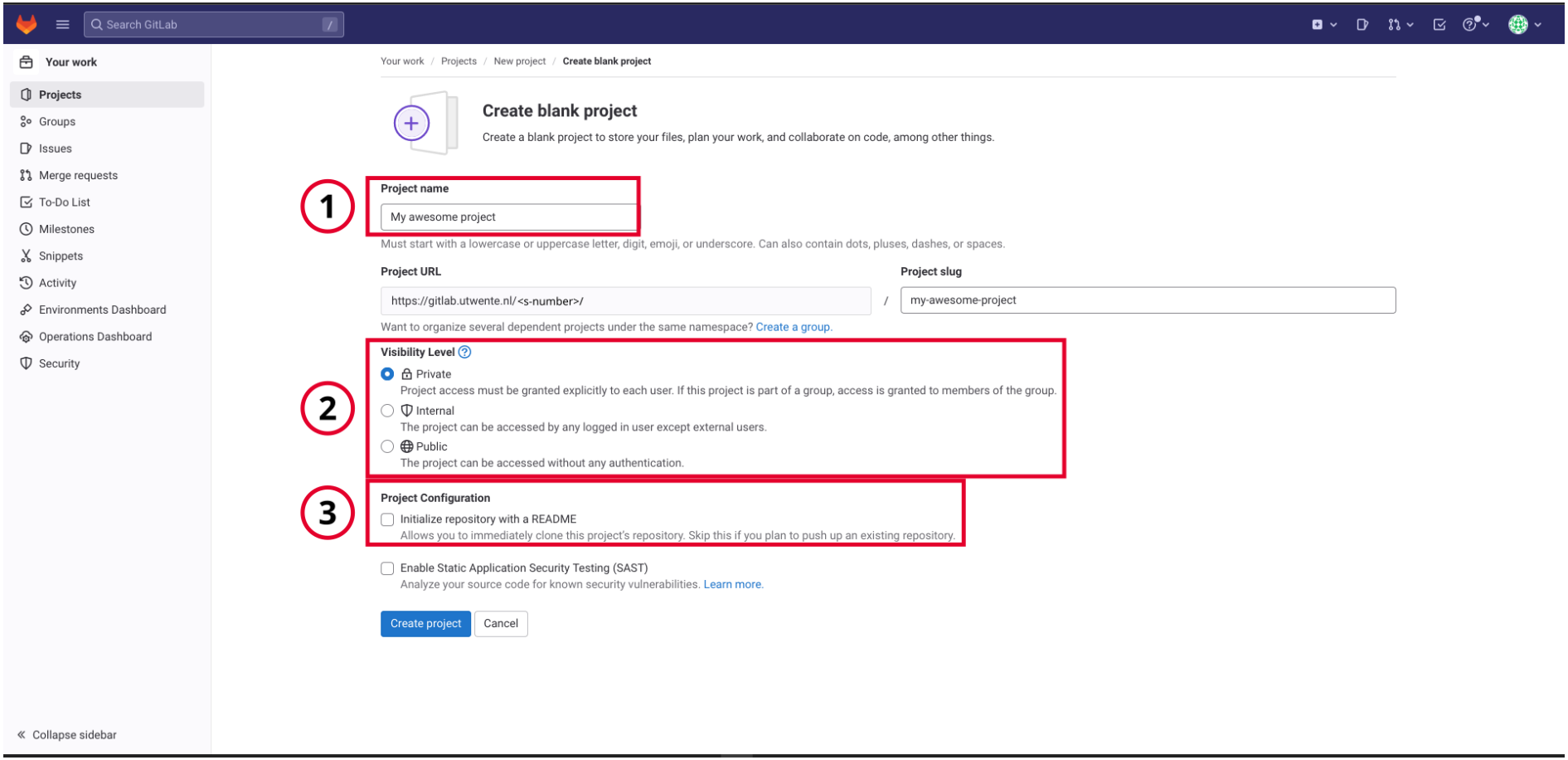
20 - Create repository on Gitlab



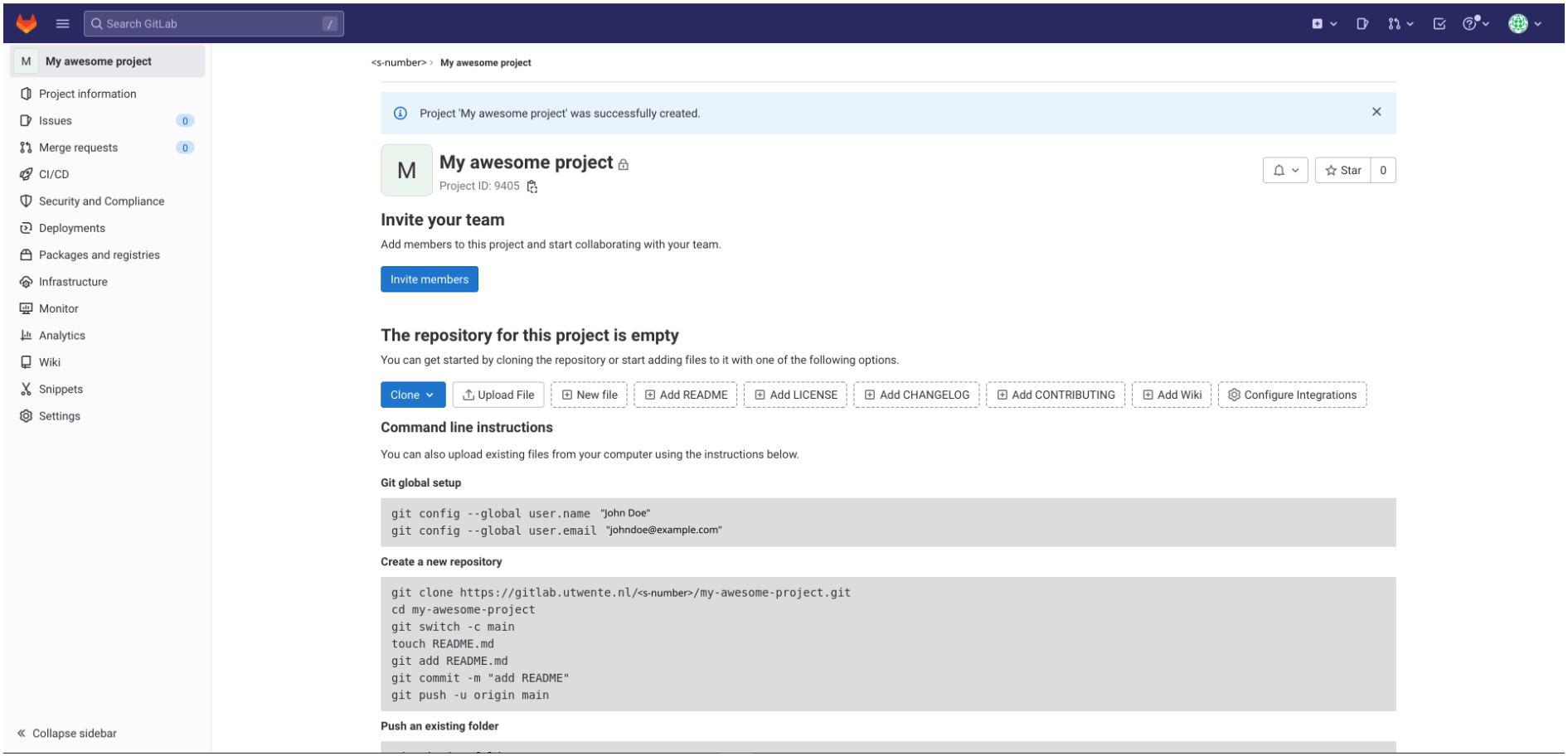
Create repository on Gitlab



Create repository on Gitlab



Create repository on Gitlab



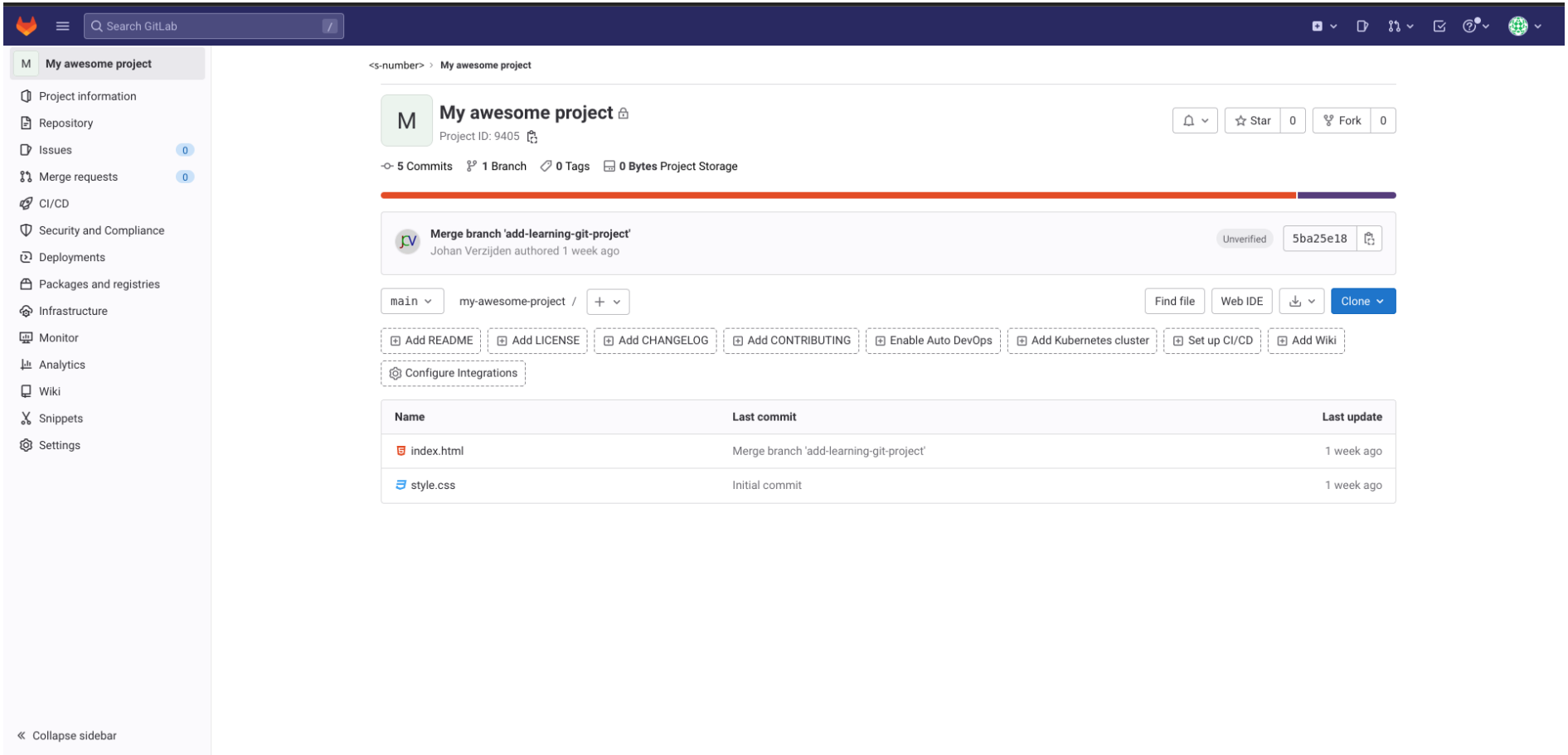
21 - Set up remote tracking locally

```
~/my-portfolio$ git remote add origin https
    ://gitlab.utwente.nl/<s-number>/<project-
    name>.git
```

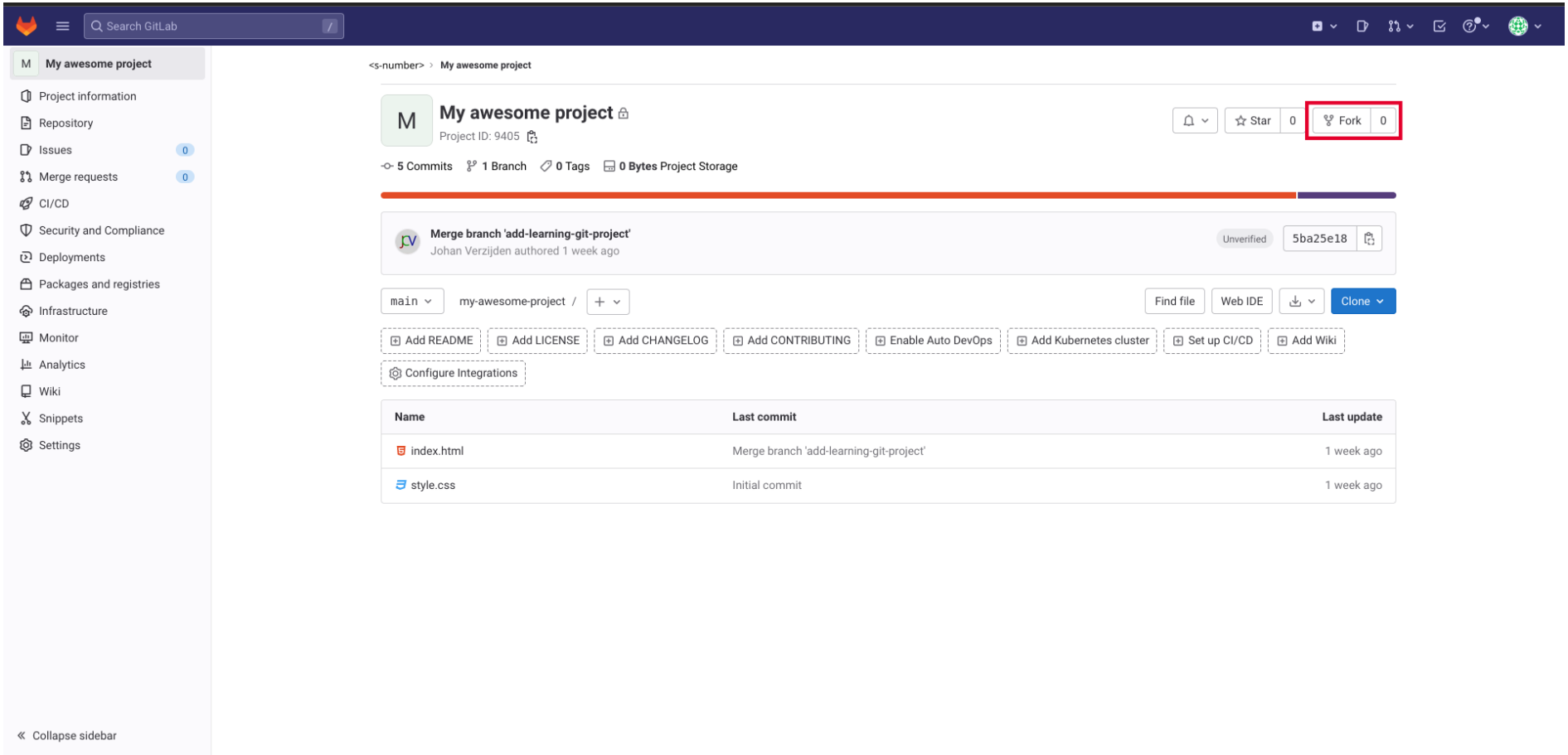
22 - Push to remote

```
~/my-portfolio$ git push -u origin main
```

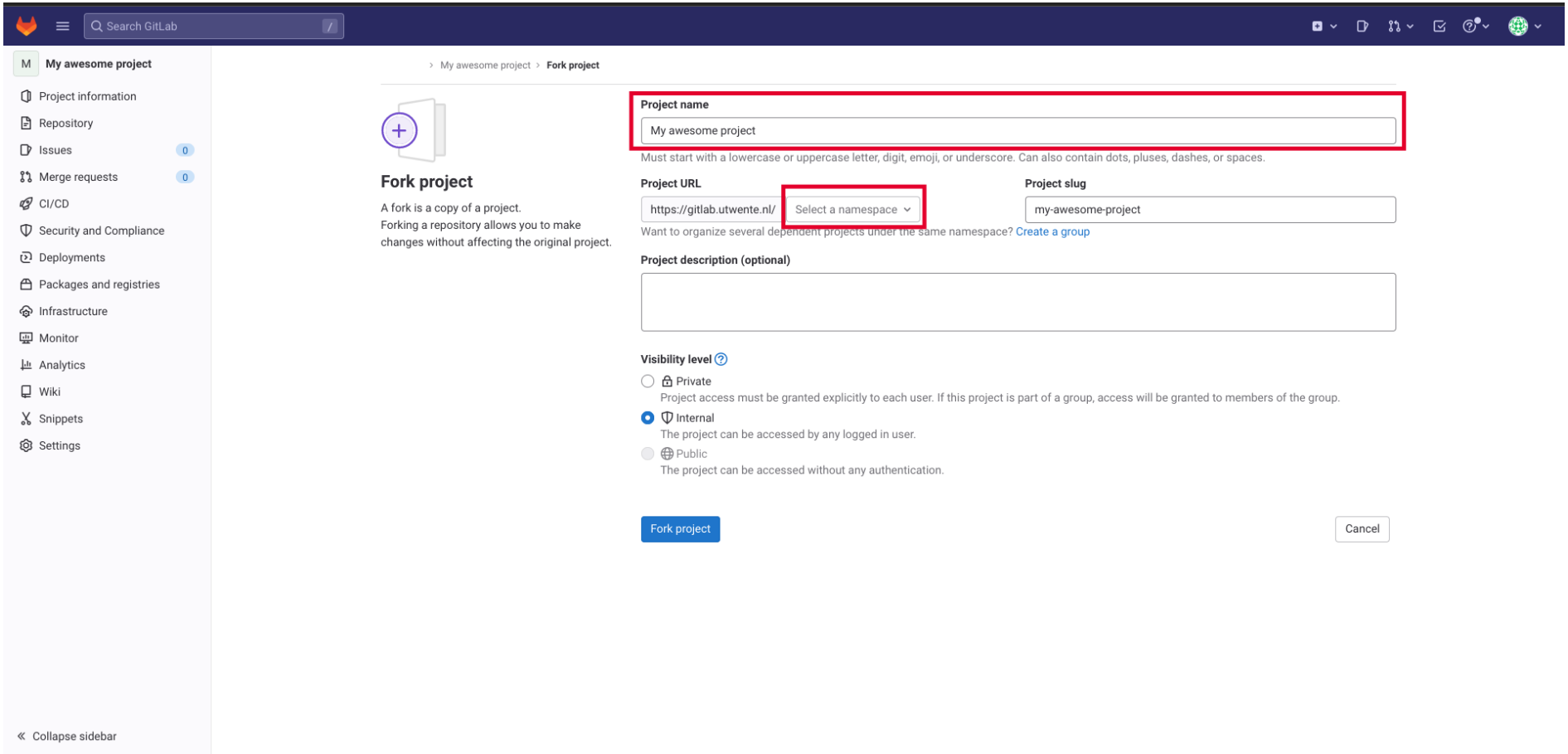
23 - Look at repository on Github/Gitlab



24 - Fork from neighbour



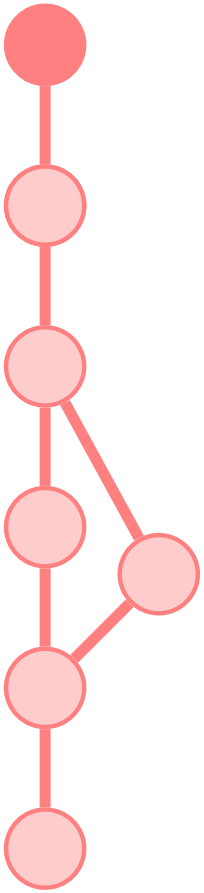
Fork from neighbour



Clone fork to local

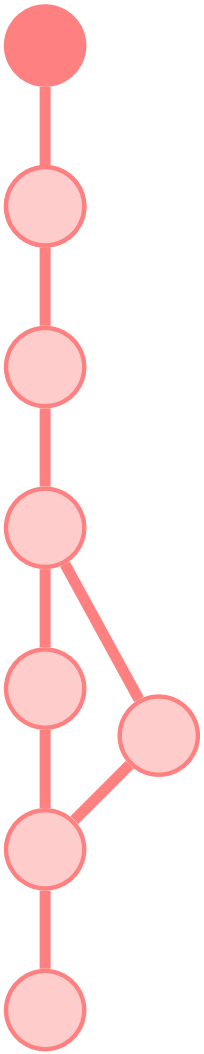
```
~$ git clone https://gitlab.utwente.nl/<s-
   number>/<forked-project-name>
~$ cd <forked-project-name>
```

25 - Revert last commit to fix the text colour



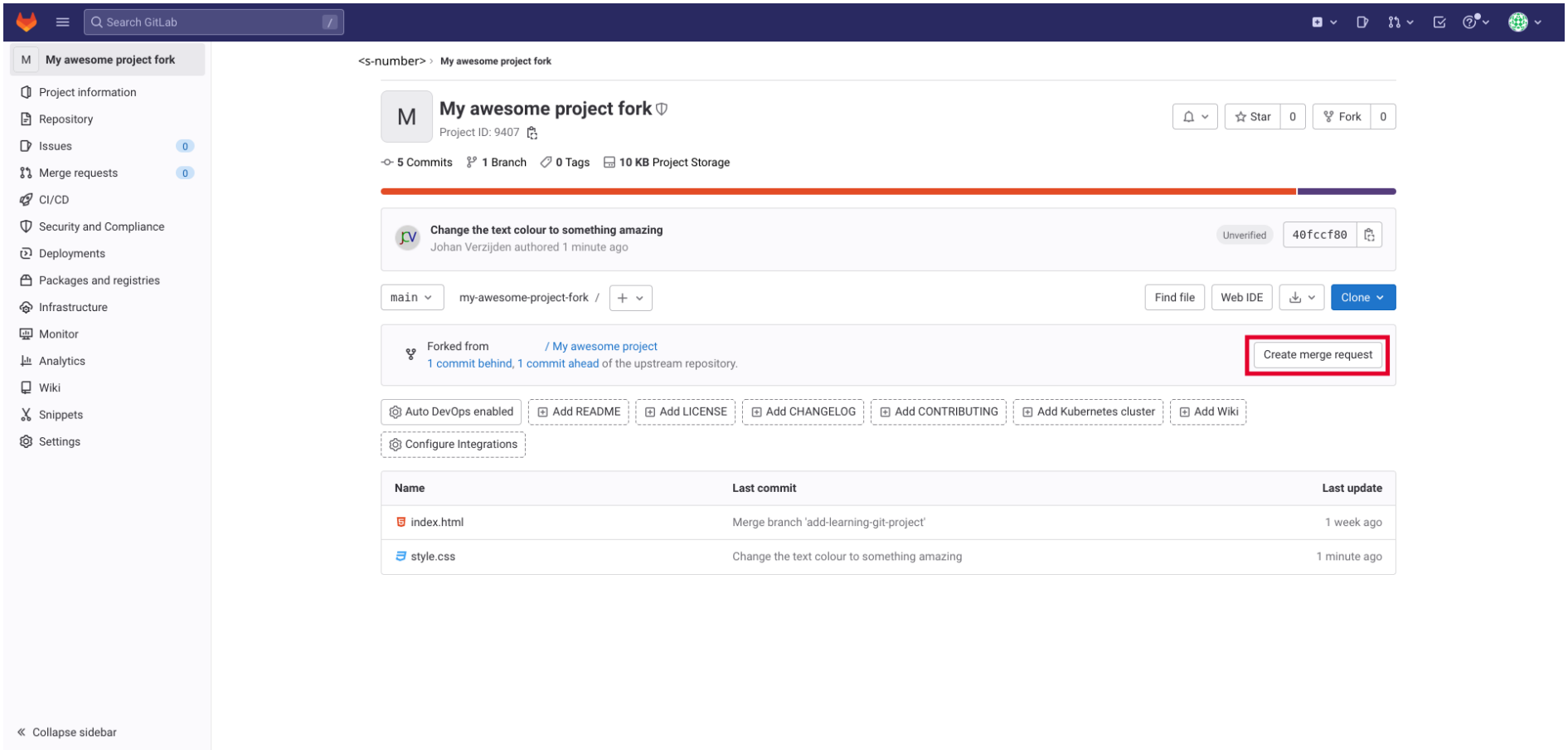
```
~/<forked-project-name>$ git revert <
    commit hash>
```

26 - Do a different commit, push to remote again

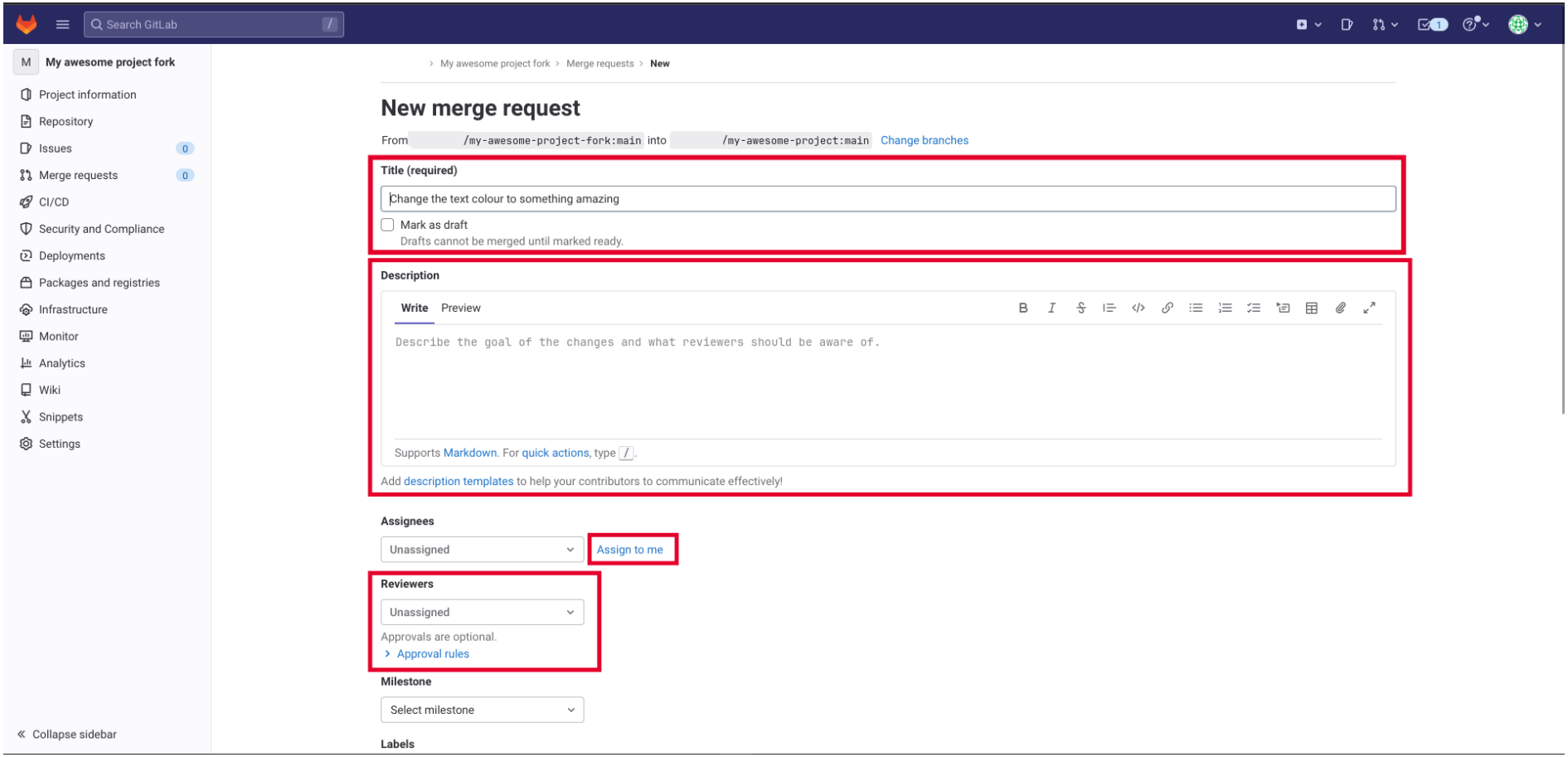


```
# Change text to a different colour
~/<forked-project-name>$ git add .
~/<forked-project-name>$ git commit -m '
    Change the text colour to something
    amazing'
~/<forked-project-name>$ git push
```

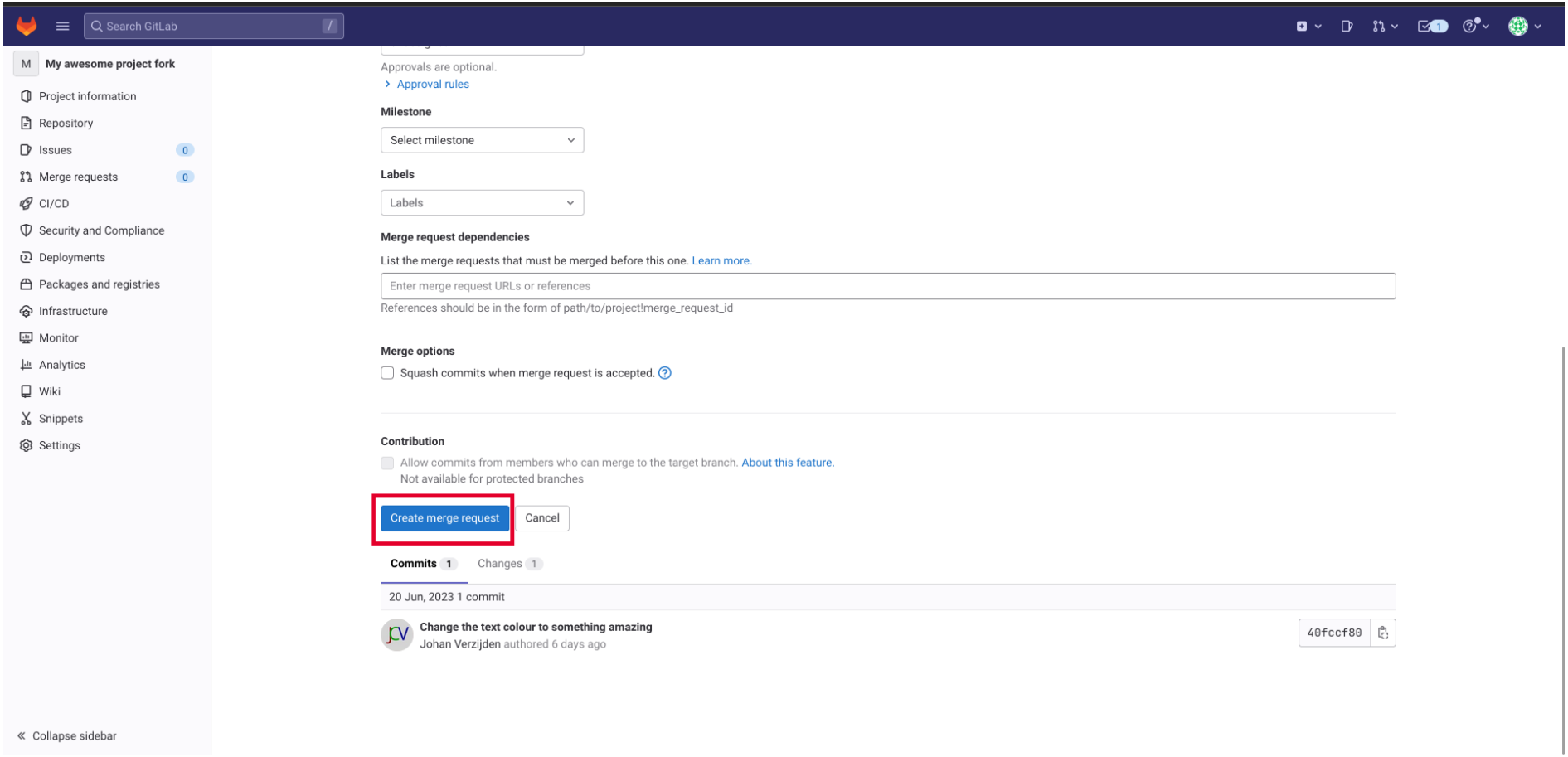
27 - Open merge request



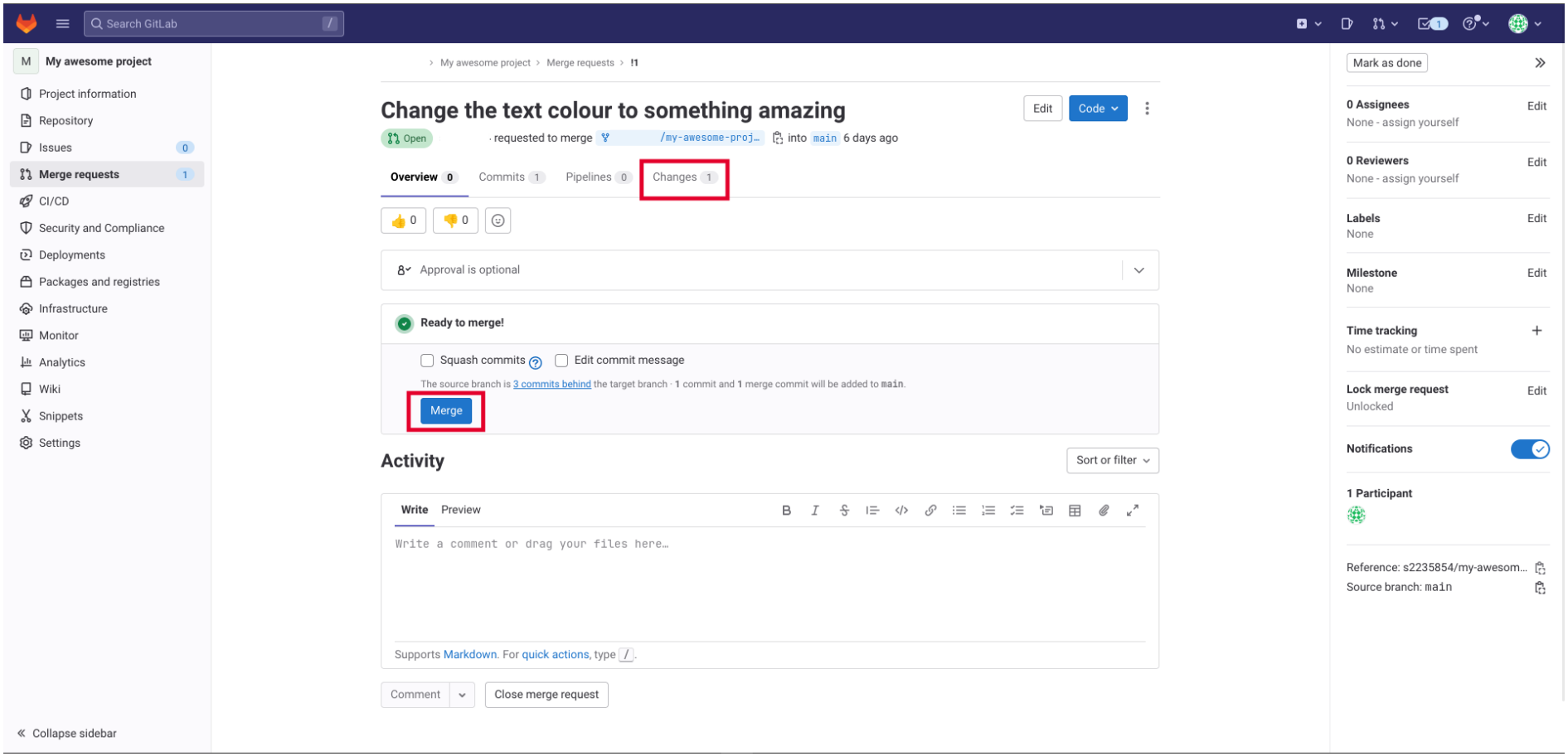
Open merge request



Open merge request



28 - Review and accept merge request



29 - Look at Blame

```
~/my-repo $ git blame index.html
```

Advanced topics

- Stash
- Rebase
- Cherry-pick

30 - Common Git Commands - Cheat sheet

git ...

- init
- add [<filenames>]
- commit [-m '<message>']
- switch [-c] <branch name>
- log [--graph --oneline]
- push
- pull
- fetch
- clone <path/URL>
- blame <file> [--color-by-age]
- revert <commit>
- reset [--hard] <commit>

Sources for future reference

- `git help <command>`
- `man gittutorial`
- gitimmersion.com

Contact details

Think of a question later on? Feel free to reach out to us!

MasterCLASS
masterclass@scintilla.utwente.nl

Kasper Müller
kasperm@scintilla.utwente.nl

Johan Verzijden
johanv@scintilla.utwente.nl