Version Control II

Joachim Vandekerckhove

Winter 2025

Version control

- Allow tracking of changes
- Allow branching and merging
- Highly secure

git

- Lots of functionality
- High performace
- Industry standard
- Not too hard to get started
- Quite hard to master

git commands

- git status
- git pull
- git push
- git add
- git commit
- git clone
- git checkout
- git diff
- git fetch
- git merge

git habits

- commit often
- always pull latest versions
- use branches
- comment on commits a lot
- test changes before committing

git first time

1. Create an ssh key pair and copy the public key to GitHub
2. Make a new repo on github.com
3. Give git your identity locally

```
git config --global user.email "jv@class-docker"
git config --global user.name "joachim"
git config --global --list
```

4. Clone the repo locally **not inside the workspace**

```
mkdir /repo
cd /repo
git clone git@github.com:<you>/<repo>.git
```

Example workflow

1. Create a new branch

```
git checkout -b small-edits
```

2. Make changes, commit frequently

... (make new file)

```
git add my.new.file.name
git commit -m "New file added!"
```

... (make small edit)

```
git commit -a -m "I made a small change"
```

... (make tiny edit)

```
git commit -a -m "I made a tiny change"
```

Different example workflow

1. Make sure you have all the latest

```
git checkout main
git fetch --all --prune
git rebase
git checkout -b bugfix
```

2. *(make small edit)*

```
git commit -a -m "Fixed bug"
git push
```

3. *(open pull request)*

Resources

Git tutorial

Git cheat sheet

Ubuntu CLI cheat sheet

Linux-fu

Version Control II

Joachim Vandekerckhove

Winter 2025