When Software Enters the Stage

Roel Aaij

November 23rd, 2022

Nikhef

Version Control

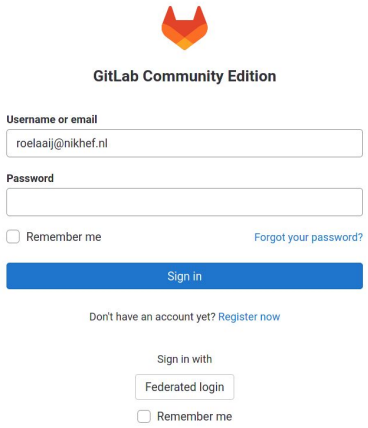
- It's very easy to forget what you did even a few days ago
- When working on multiple projects, it's worse
- When writing your thesis, you'll need to remake plots

- You want to keep track of the history of what you're doing
  - Data
  - Code
  - Documentation

- The answer is: version control
- A very popular tool: git
- https://swcarpentry.github.io/git-novice/

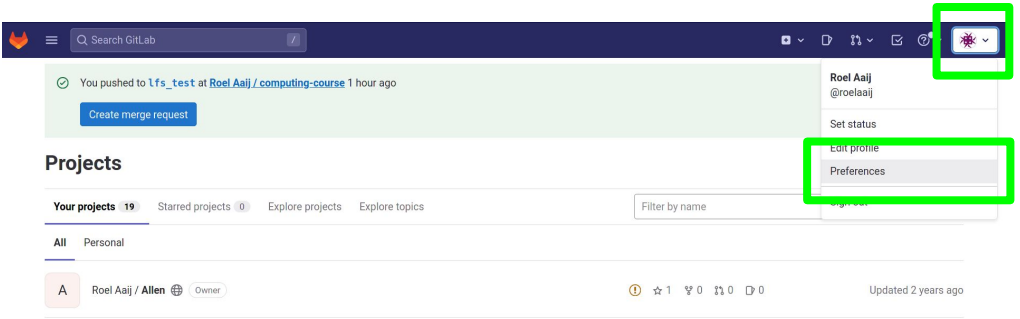
GitLab

- We now have a repository somewhere on stoomboot, let's make that more useful
- go to https://gitlab.nikhef.nl and hit "Federated login"



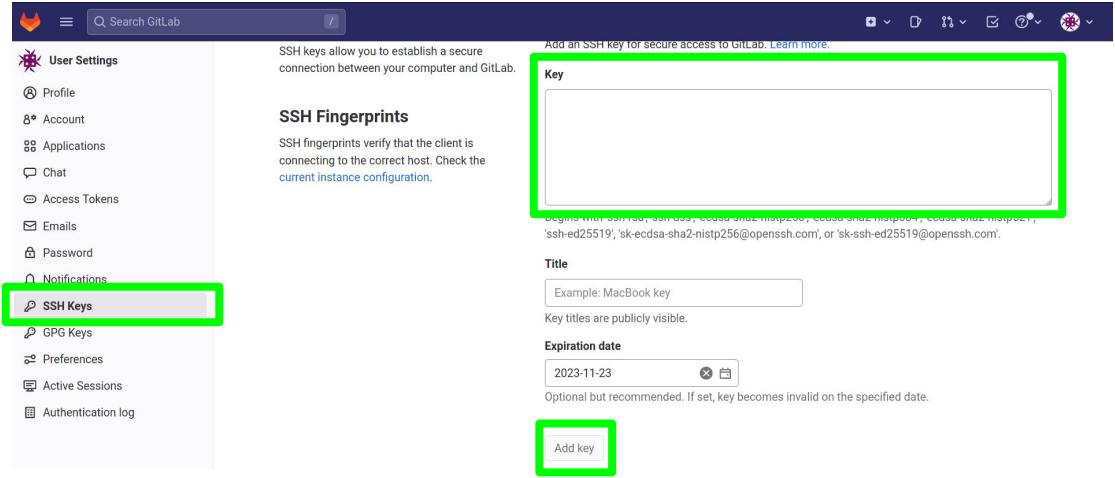
GitLab SSH Keys

- Intermezzo: add your SSH key to gitlab
- https://www.nikhef.nl/pdp/computing-course/work/ssh.html
- `cat ~/.ssh/id_rsa.pub`
- Click on your profile and select “Preferences”

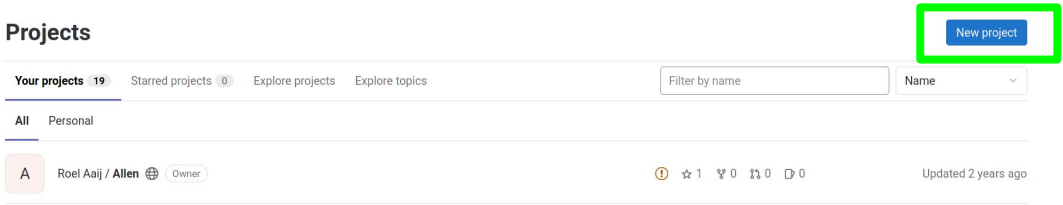


GitLab SSH Keys

- Select “SSH Keys” from the menu on the left
- Copy paste your public key into the box and add it with a title



New GitLab project



New GitLab project

New project > **Create blank project**

Project name

TestProject

Project URL

https://gitlab.nikhef.nl/ roelaaij /

Project slug

testproject

Want to organize several dependent projects under the same namespace? Create a group.

Visibility Level

- Private
  Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
- Internal
  The project can be accessed by any logged in user except external users.

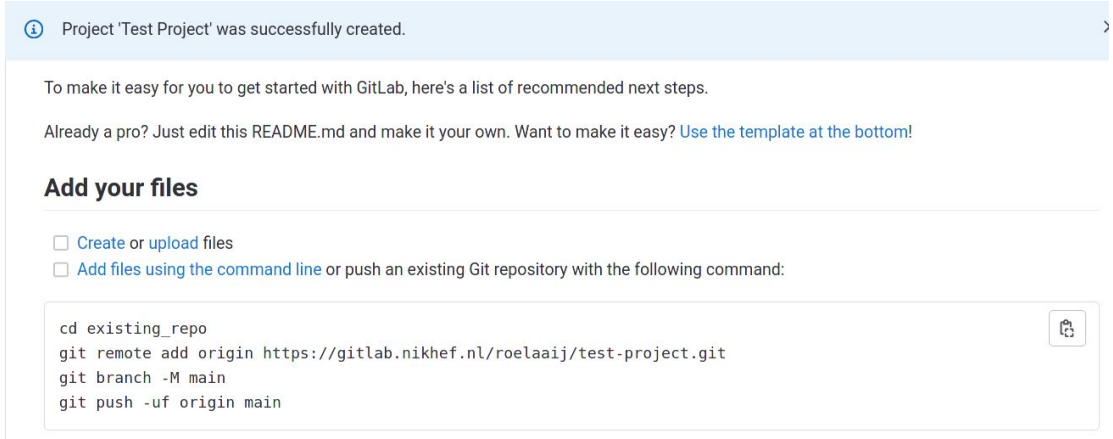
Project Configuration

- [x] Initialize repository with a README
  Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
- [ ] Enable Static Application Security Testing (SAST)
  Analyze your source code for known security vulnerabilities. Learn more.

Create project Cancel

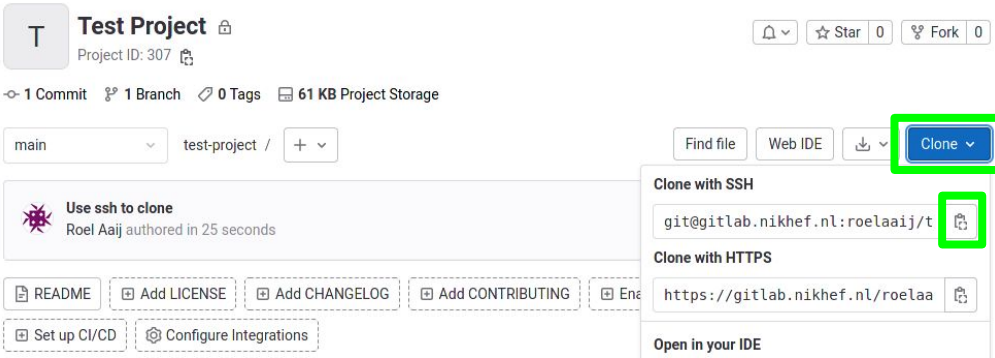
New GitLab project

- Copy the commands, but instead of https, we need ssh



New GitLab project

- Copy the ssh URL



What about data?

- You should keep track of your data too
- Git is great for ASCII encoded files, not so much for binary
- Do it in the same repository, with symlinks
- Create folders for various types of files
  - Work in progress
  - Plot created from this data
  - Data used for thesis or publication
  - etc.
- Use the symlinks in your code and notebooks
- Commit them as part of the rest

What about software?

- Create a conda environment
  https://www.nikhef.nl/pdp/computing-course/software/where-to-get.html
- Create conda spec file, and store it in your repository:
  `$> conda list --explicit > spec-file.txt`
- Create environment from spec file:
  `$> conda create --name myenv --file spec-file.txt`
- This relies on the availability of packages in conda
- For experiment software: include the version(s) of the release
- For software you develop:
  - Create a repository for your software
  - Create a release
  - Upload it to zenodo
  - cite the DOI
- For those who want more: build containers

Zenodo

- Zenodo is a general purpose repository for scientific documents and data
- Hosted by CERN
- For each submission, a digital object identifier (DOI) is created, which allow citation of the submission
- Let's go and upload something:
  - `$> tar jcf my_repo.tar.bz2 my_repo`
  - https://sandbox.zenodo.org/
  - sign up (easiest with ORCID)
  - log in
  - upload

arXiv

- arXiv is a preprint server
- The majority of papers in our fields are put on arXiv
- No peer-review, also many “interesting” papers
- A DOI is generated for each submission
- arXiv publications can be cited
- Don’t forget to update the citation once a journal publication is available
- Many categories available:
  - https://arxiv.org
  - https://arxiv.org/archive/hep-ex
  - https://arxiv.org/archive/hep-ph
  - https://arxiv.org/archive/hep-th
  - https://arxiv.org/archive/cs.LG

Journal

- In addition to what you create, you may also want to keep track of what you do
- A research journal
- You can do it on old-fashioned paper
- Or use a digital one and store it using git (with a remote on gitlab.nikhef.nl)

- Emacs + org-mode: https://orgmode.org/worg/org-tutorials/org4beginners.html
- Markdown in your editor of choice
- LaTeX
- Jupyter notebooks: https://nextjournal.com/schmudde/how-to-version-control-jupyter