Github

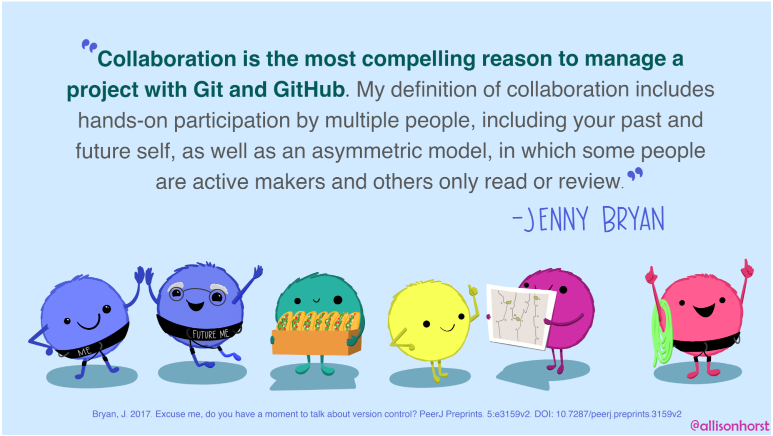
Sophie C. Schmidt, Florian Thiery

Github

What is Github?

- “a code hosting platform for version control and collaboration”
- a remote git repository, where you can
    - work collaboratively
    - in a version controlled way,
    - discuss changes “where they are made”,
    - interact with other users
- or simply back up your stuff remotely

Collaboration



"**Collaboration is the most compelling reason to manage a project with Git and GitHub**. My definition of collaboration includes hands-on participation by multiple people, including your past and future self, as well as an asymmetric model, in which some people are active makers and others only read or review."

-JENNY BRYAN

Bryan, J. 2017. Excuse me, do you have a moment to talk about version control? PeerJ Preprints. 5:e3159v2. DOI: 10.7287/peerj.preprints.3159v2

@allisonhorst

Github repository

Your Github repository

- has a name (github.com/yourname/yourrepo)
- settings:
  - private or public
    - private: you decide who can see it
    - public: everyone can see it
    - both: YOU decide who commits
  - initialize with a README?
  - add a gitignore?
  - choose a license

Short excursion: **README**

What is a README.md?

- md = markdown:
  - very simple markup language
  - hashtags (#) for headers
  - minus (-) for bullet points
  - asterisk (*) to make things italic: *italic*: *italic*
  - ...
  - git can track changes, because it's simple text
- README
  - a file that gets displayed first in the repo
  - project title
  - description
  - how to use the code (from data, workflows,... to licenses)
  - how to contribute
  - how to cite

Let me show you

Follow if you are fast enough, otherwise there will be time later!

Connecting to Github

Authentification with SSH

1. Testing your SSH connection [GitHub Docu]
2. Generate a new SSH Key [GitHub Docu]
3. Adding a new SSH key to your GitHub account [GitHub Docu]
4. Testing your SSH connection (step 1)

Testing your SSH connection

```
ssh -T git@github.com # Attempts to ssh to GitHub

# it worked!
> The authenticity of host 'github.com (IP ADDRESS)'
> can\'t be established. RSA key fingerprint is
> SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
> Are you sure you want to continue connecting (yes/no)?
> Hi username! You\'ve successfully authenticated,
> but GitHub does not provide shell access.

# SSH key is missing
> Permission denied (publickey).
```

Generating a new SSH key

```
ssh-keygen -t ed25519 -C "your_email@example.com"
# use your GitHub Email
> Generating public/private algorithm key pair.
> Enter a file in which to save the key
> (/c/Users/you/.ssh/id_algorithm): # press enter
> Enter passphrase (empty for no passphrase): # type a pw
> Enter same passphrase again: # type pw again
```

Adding your SSH key to the SSH Agent

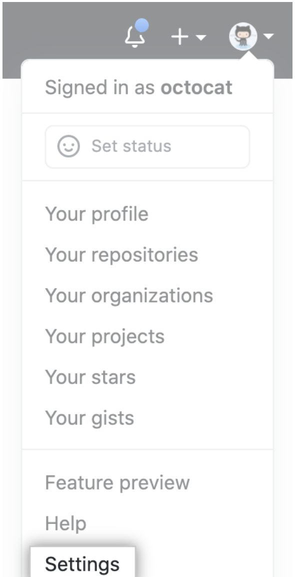
```
eval "$(ssh-agent -s)"
# start the ssh-agent in the background
ssh-add ~/.ssh/id_ed25519
# add your SSH private key to the ssh-agent
```

Adding a new SSH key to your GitHub account

```
clip < ~/.ssh/id_ed25519.pub #copies key to your clipboard
```

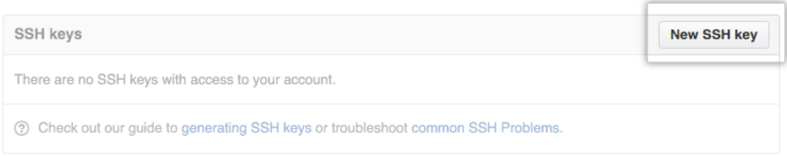
Go to GitHub online (Part 1)

In the upper-right corner of any page, click your profile photo, then



Go to GitHub online (Part 2)

In the “Access” section of the sidebar, click “SSH and GPG keys”. Click New SSH key or Add SSH key.


SSH keys

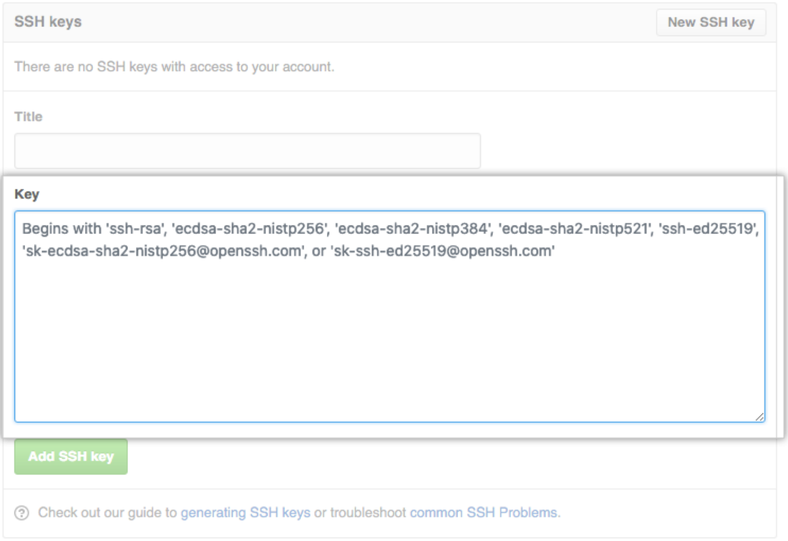
There are no SSH keys with access to your account.

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

In the “Title” field, add a descriptive label for the new key. For example, if you’re using a personal Mac, you might call this key “Personal MacBook Air”.

Go to GitHub online (Part 3)

Paste your key into the “Key” field, then click “Add SSH key”.


SSH keys

New SSH key

There are no SSH keys with access to your account.

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

Testing your SSH connection

```
ssh -T git@github.com # Attempts to ssh to GitHub

# it worked!
> The authenticity of host 'github.com (IP ADDRESS)'
> can\'t be established. RSA key fingerprint is
> SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
> Are you sure you want to continue connecting (yes/no)?
> Hi username! You\'ve successfully authenticated,
> but GitHub does not provide shell access.
```

Let's check: Does this work for all of you?

Workflow

Connecting remote and local repos

- easiest way: *clone* the github repository to your own computer
- = download the folder that is your repository
- use :

```
cd /path/to/where/you/want/it
git clone git@github.com:yourname/yourrepo.git
```

−> creates new git-repository
/path/to/where/you/want/it/yourrepo

- now change / add / commit as you want
- your local repo then has an "origin", which is the remote github repo from where you cloned

Synchronise your local and remote repos

- we want to synchronise them, so they have the same commit history
  - **push**: uploades changes **to** remote
  - **pull**: downloads (`git fetch`) and merges (`git merge`) changes **from** the remote in one go
  - ONLY commited changes get copied
- `git clone`: needs to be done once
- `git pull`: anytime something in the remote has been changed

Synchronise your local and remote branches

- pull and push can be done with specific branches

```
git push origin main # pushes main branch to remote's main
```

- same structure for other branches:

```
git pull <branch_remote> <branch_local>
```

Push doesn't work?

```
(base) sophie@idhrenisle:
~/R/caa2022_GitGitHub_workshop/slides
$ git push origin main
To github.com:sslarch/caa2022_GitGitHub_workshop.git
! [rejected]    main -> main (fetch first)
```

–> solution: pull!

```
(base) sophie@idhrenisle:
~/R/caa2022_GitGitHub_workshop/slides $ git pull
origin main...
```

```
Von github.com:sslarch/caa2022_GitGitHub_workshop
 * branch            main       -> FETCH_HEAD
   89c51f9..05066e5  main       -> origin/main
Merge made by the 'recursive' strategy.
```

- and now push again. :-)

If Merge doesn't work?

- if you pull sth that can't be merged automatically
  - DON'T PANIC

In the file you will find “conflict markers”:

- `<<<<<<<HEAD.`: beginning of the conflict, now your version of the text is shown
- =======: divides your version from the changes in the other branch
- >>>>>>> `BRANCH-NAME` : end of the problem with the name of the “rival” branch

Merge conflict example

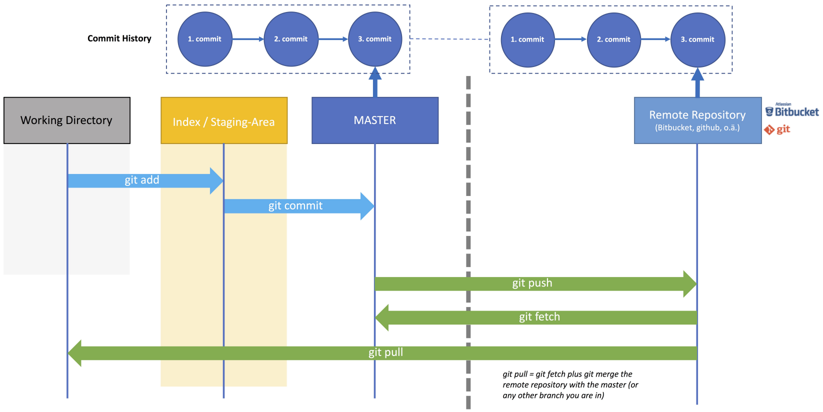
- one person wrote “open an issue” in the base or HEAD branch
- another person wrote “ask your question in IRC” in the compare branch or branch-a.
- both the same line –> git doesn’t know which to save

```
If you have questions, please
<<<<<<< HEAD
open an issue
=======
ask your question in IRC.
>>>>>>> branch-a
```

Merge conflict solution

- easy way:
  - amend the affected lines by hand
  - delete the conflict markers
  - save
  - git add
  - git commit
- nothing gets overwritten by accident!

Visualisation of Workflow



Excercise

Create and use your Github repository

- open Github and log in
- create a new repository
    - make it public
    - initialize with a README
    - add a gitignore
- now:

```
cd /path/to/where/you/want/it
git clone git@github.com:yourname/yourrepo.git
```

Well done!

- open the README.md
- write a couple of words (use `echo`, nano or your own text editor)
- git add
- git commit
- git push

Collaborating

Issues: Github's space for discussions

- can be created by anyone to suggest additions, changes, . . .
- discussion thread of comments
- to dos
- assign people to tasks
- link comments to commits / code lines (?)
- *close* them when done discussing!

Pull requests: Github's infrastructure for collaboration

- public repositories can be *forked* by anybody
  - = make a copy to your own profile
  - there make all the changes and commits you want
  - want to suggest changes to the owner of the original repo?
    –> create a *pull request* (PR)
- owner looks at your commits and decides whether they want to integrate the changes or not
- you can *comment* the PR & discuss changes

Excercise

Collaborating with a partner

- take a partner
- exchange URLs of your repositories
- fork the repo of your partner
- clone it to your computer
- add a sentence to his README
- git add, commit, push
- create a pull request
- partner merges

And now?

- your partner change something remotely in your repository!
- you need to pull the change

```
cd /your/repo
git pull
```