Git

RESEARCH INSTITUTE
NATURE AND FOREST

Peter Desmet, Stijn Van Hoey

Who are we?

Stijn Van Hoey

Twitter

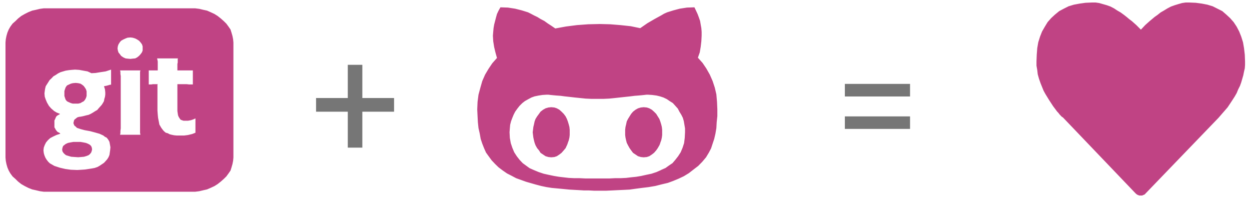
GitHub

Peter Desmet

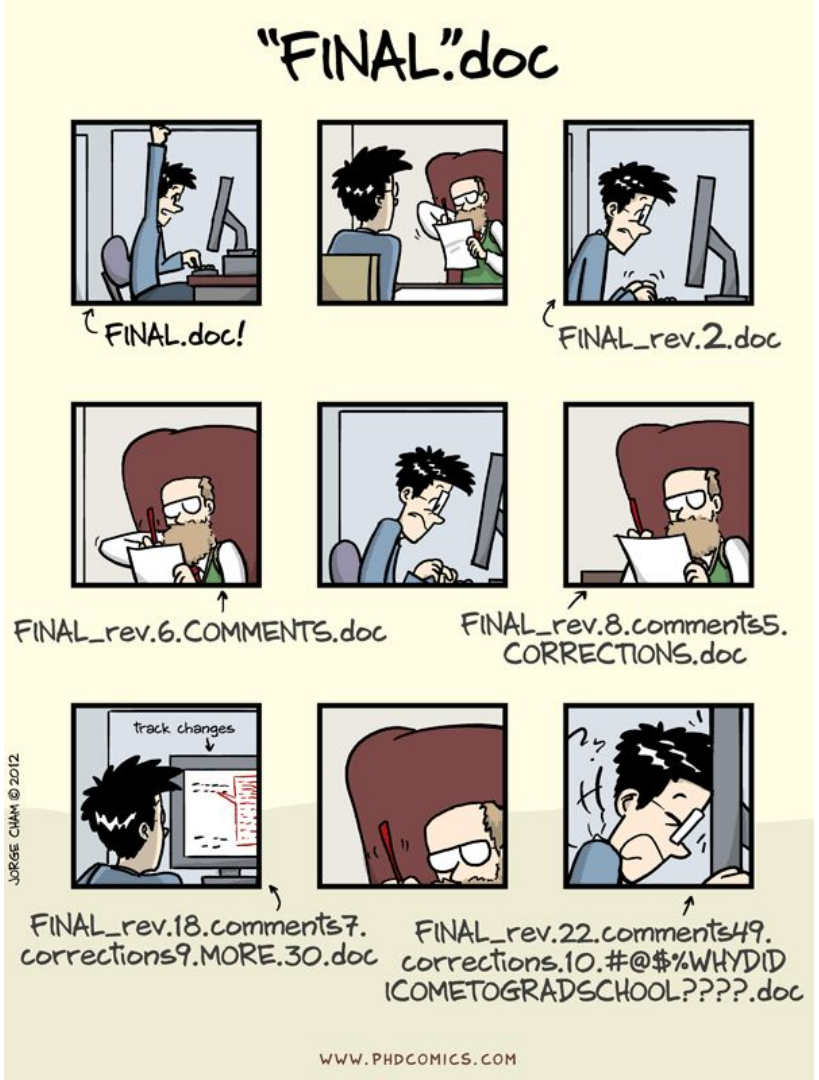
Twitter

GitHub

What is your current backup strategy?

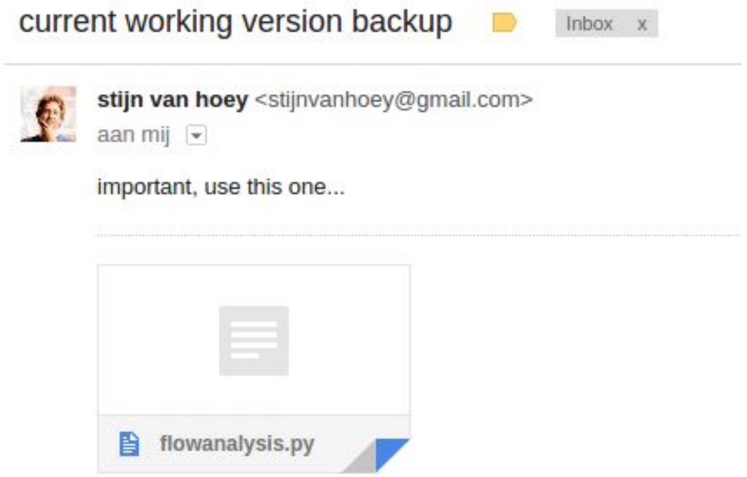

git
+
=

SO WHAT?



http://swcarpentry.github.io/git-novice/01-basics/

"Piled Higher and Deeper" by Jorge Cham,
http://www.phdcomics.com

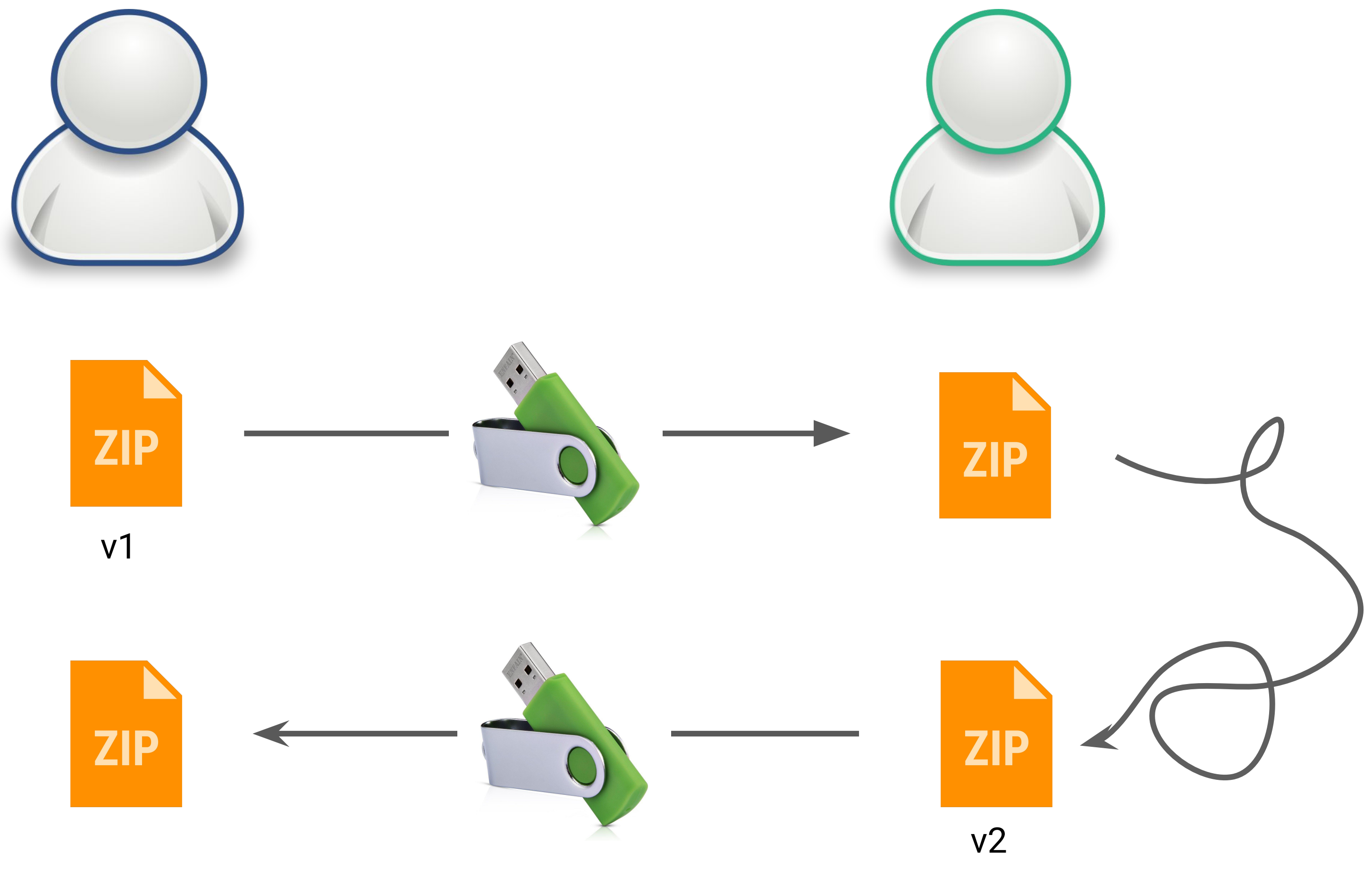

current working version backup Inbox x

stijn van hoey <stijnvanhoey@gmail.com>
aan mij

important, use this one...

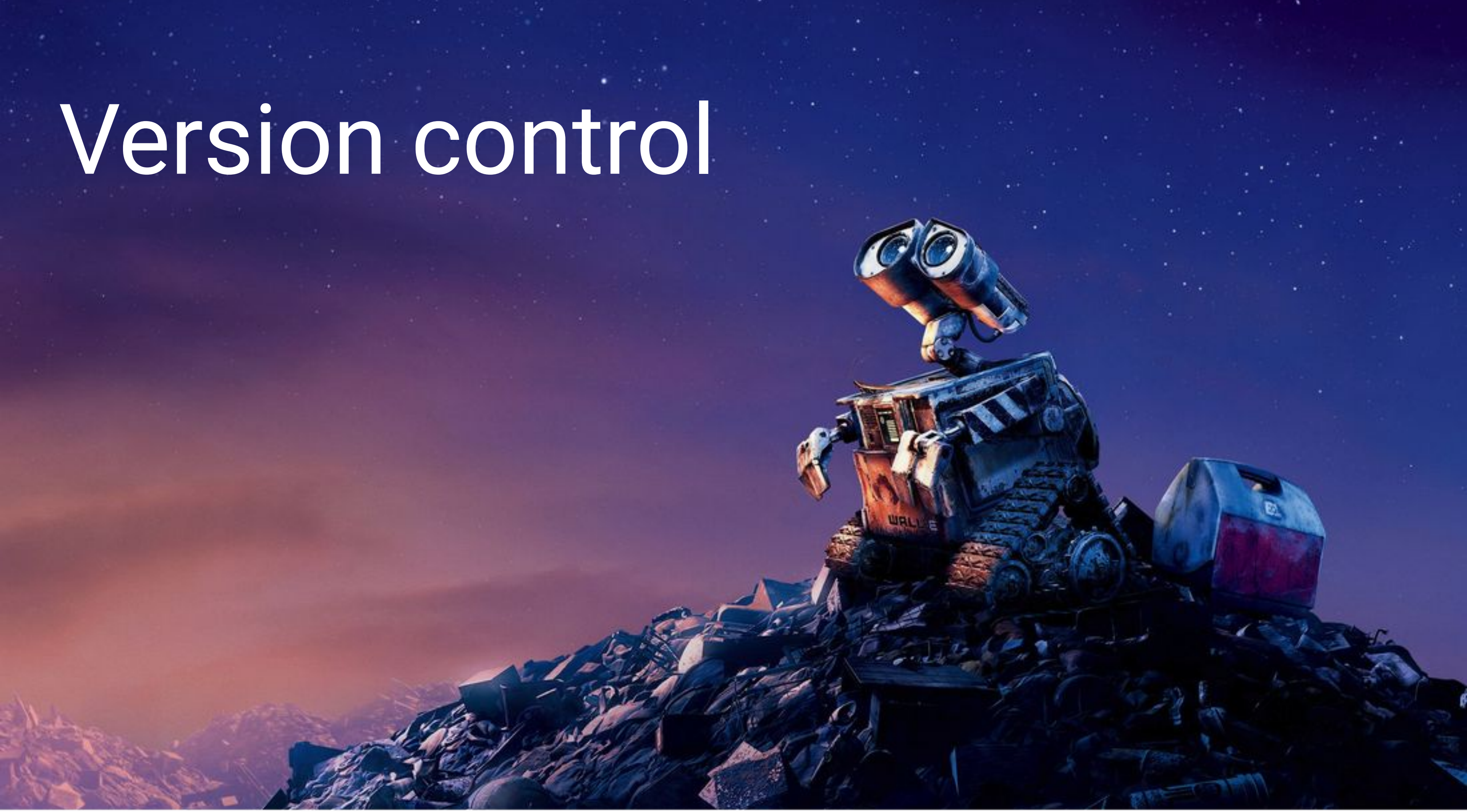
flowanalysis.py

I KNOW
WHAT YOU
DID LAST
SUMMER


ZIP
v1
ZIP
ZIP
ZIP
v2

WALL·E

Version control



Version control...

1. Tell the story of your project
2. Travel back in time
3. Experiment with changes
4. Backup your work
5. Collaborate on projects

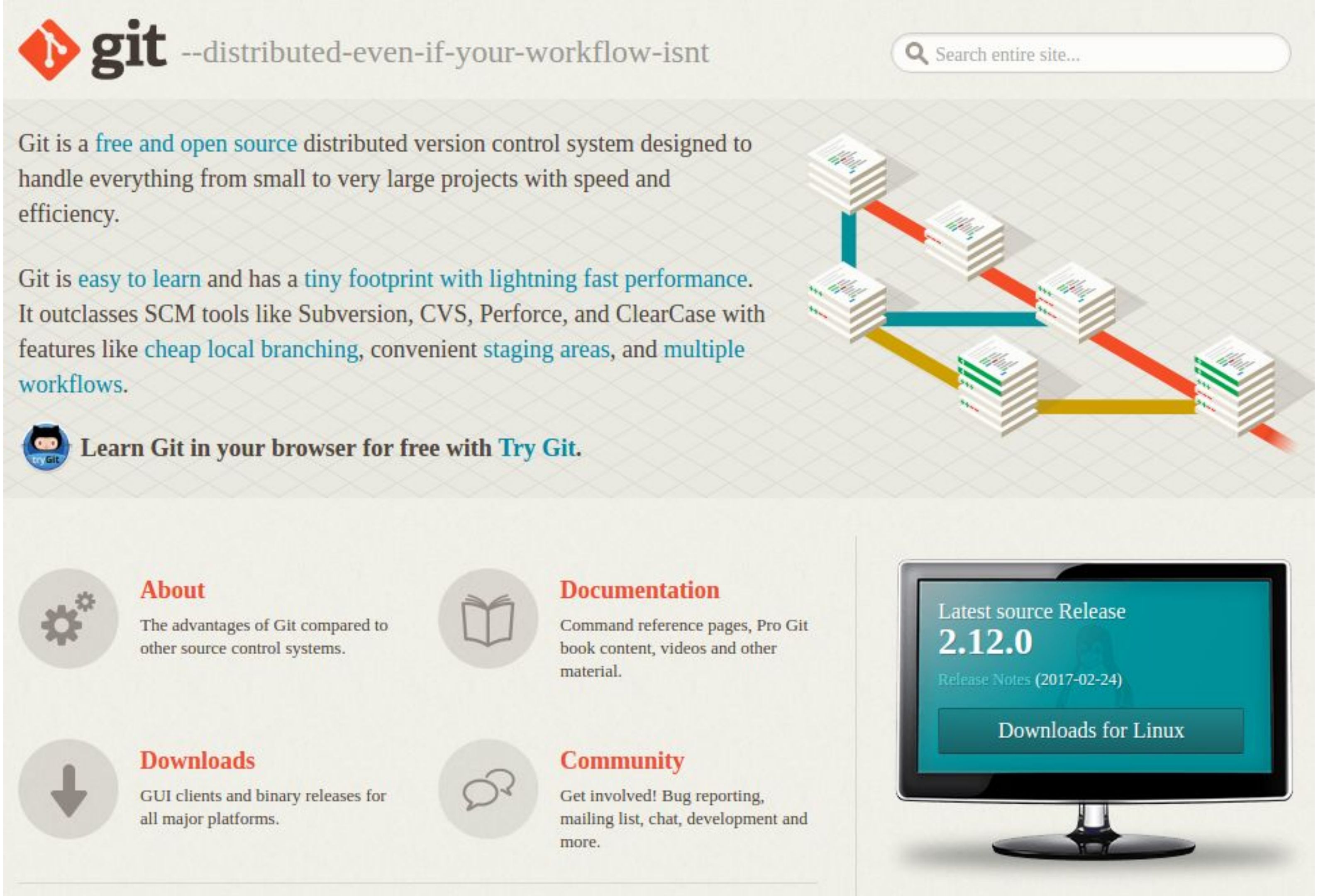
Version control...

1. Tell the story of your project git
2. Travel back in time git
3. Experiment with changes git
4. Backup your work git
5. Collaborate on projects git

What is Git?

"Git is an application that runs on your computer, like a web browser or a word processor"

(Tom stuart)



git --distributed-even-if-your-workflow-isnt

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Learn Git in your browser for free with Try Git.

About
The advantages of Git compared to other source control systems.

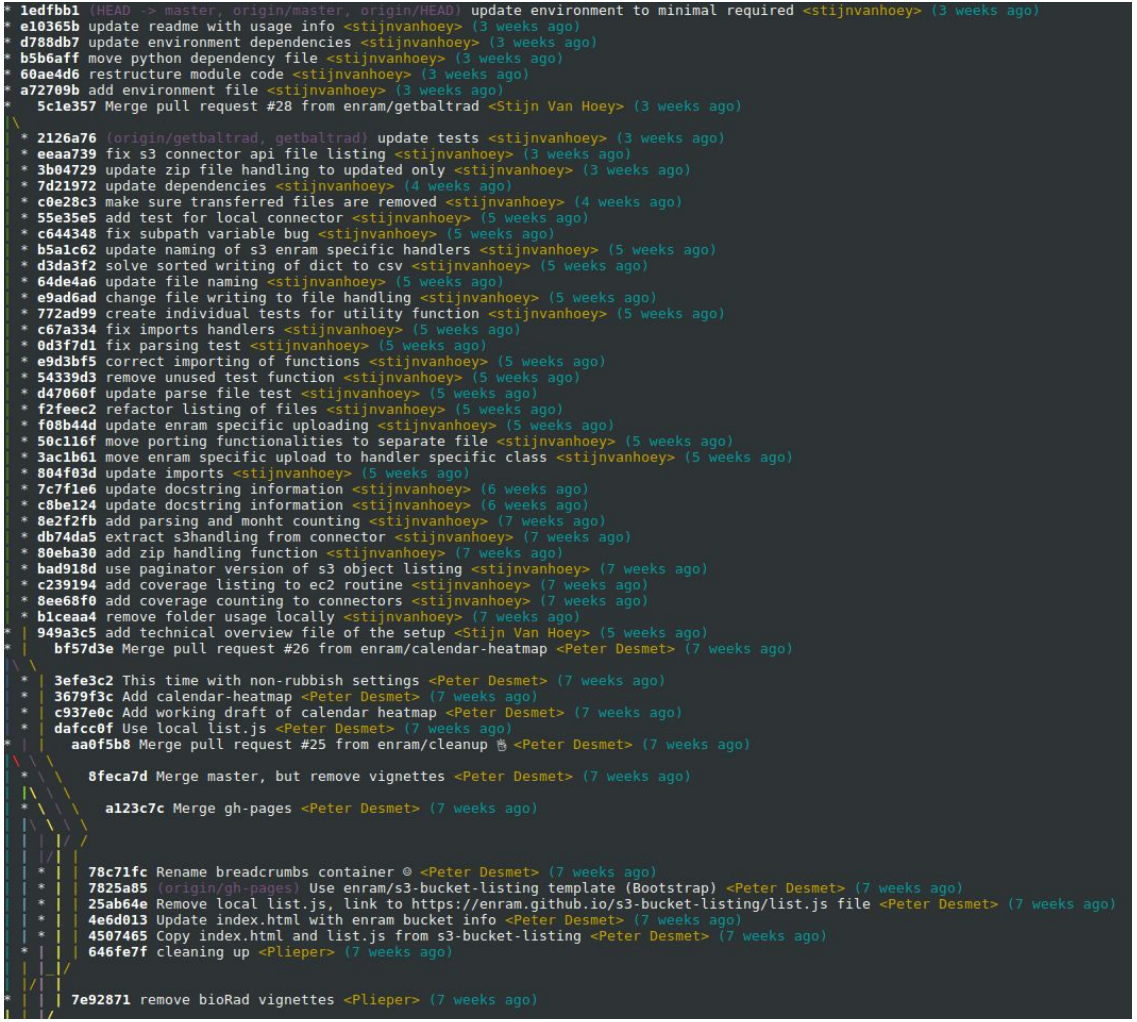
Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.12.0
Release Notes (2017-02-24)
Downloads for Linux

Git has a bad reputation...



```
* 1edfbb1 (HEAD -> master, origin/master, origin/HEAD) update environment to minimal required <stijnvanhoey> (3 weeks ago)
* e10365b update readme with usage info <stijnvanhoey> (3 weeks ago)
* d788db7 update environment dependencies <stijnvanhoey> (3 weeks ago)
* b5b6aff move python dependency file <stijnvanhoey> (3 weeks ago)
* 60ae4d6 restructure module code <stijnvanhoey> (3 weeks ago)
* a72709b add environment file <stijnvanhoey> (3 weeks ago)
*   5c1e357 Merge pull request #28 from enram/getbaltrad <Stijn Van Hoey> (3 weeks ago)
|\
| * 2126a76 (origin/getbaltrad, getbaltrad) update tests <stijnvanhoey> (3 weeks ago)
| * eeaa739 fix s3 connector api file listing <stijnvanhoey> (3 weeks ago)
| * 3b04729 update zip file handling to updated only <stijnvanhoey> (3 weeks ago)
| * 7d21972 update dependencies <stijnvanhoey> (4 weeks ago)
| * c0e28c3 make sure transferred files are removed <stijnvanhoey> (4 weeks ago)
| * 55e35e5 add test for local connector <stijnvanhoey> (5 weeks ago)
| * c644348 fix subpath variable bug <stijnvanhoey> (5 weeks ago)
| * b5a1c62 update naming of s3 enram specific handlers <stijnvanhoey> (5 weeks ago)
| * d3da3f2 solve sorted writing of dict to csv <stijnvanhoey> (5 weeks ago)
| * 64de4a6 update file naming <stijnvanhoey> (5 weeks ago)
| * e9ad6ad change file writing to file handling <stijnvanhoey> (5 weeks ago)
| * 772ad99 create individual tests for utility function <stijnvanhoey> (5 weeks ago)
| * c67a334 fix imports handlers <stijnvanhoey> (5 weeks ago)
| * 0d3f7d1 fix parsing test <stijnvanhoey> (5 weeks ago)
| * e9d3bf5 correct importing of functions <stijnvanhoey> (5 weeks ago)
| * 54339d3 remove unused test function <stijnvanhoey> (5 weeks ago)
| * d47060f update parse file test <stijnvanhoey> (5 weeks ago)
| * f2feec2 refactor listing of files <stijnvanhoey> (5 weeks ago)
| * f08b44d update enram specific uploading <stijnvanhoey> (5 weeks ago)
| * 50c116f move porting functionalities to separate file <stijnvanhoey> (5 weeks ago)
| * 3ac1b61 move enram specific upload to handler specific class <stijnvanhoey> (5 weeks ago)
| * 804f03d update imports <stijnvanhoey> (5 weeks ago)
| * 7c7f1e6 update docstring information <stijnvanhoey> (6 weeks ago)
| * c8be124 update docstring information <stijnvanhoey> (6 weeks ago)
| * 8e2f2fb add parsing and monht counting <stijnvanhoey> (7 weeks ago)
| * db74da5 extract s3handling from connector <stijnvanhoey> (7 weeks ago)
| * 80eba30 add zip handling function <stijnvanhoey> (7 weeks ago)
| * bad918d use paginator version of s3 object listing <stijnvanhoey> (7 weeks ago)
| * c239194 add coverage listing to ec2 routine <stijnvanhoey> (7 weeks ago)
| * 8ee68f0 add coverage counting to connectors <stijnvanhoey> (7 weeks ago)
| * b1ceaa4 remove folder usage locally <stijnvanhoey> (7 weeks ago)
* | 949a3c5 add technical overview file of the setup <Stijn Van Hoey> (5 weeks ago)
* |   bf57d3e Merge pull request #26 from enram/calendar-heatmap <Peter Desmet> (7 weeks ago)
|\ \
| * | 3efe3c2 This time with non-rubbish settings <Peter Desmet> (7 weeks ago)
| * | 3679f3c Add calendar-heatmap <Peter Desmet> (7 weeks ago)
| * | c937e0c Add working draft of calendar heatmap <Peter Desmet> (7 weeks ago)
| * | dafcc0f Use local list.js <Peter Desmet> (7 weeks ago)
* | |   aa0f5b8 Merge pull request #25 from enram/cleanup ✋ <Peter Desmet> (7 weeks ago)
|\ \ \
| * \ \   8feca7d Merge master, but remove vignettes <Peter Desmet> (7 weeks ago)
| |\ \ \
| * \ \ \   a123c7c Merge gh-pages <Peter Desmet> (7 weeks ago)
| |\ \ \ \
| | | |/ /
| | |/| |
| | * | | 78c71fc Rename breadcrumbs container ☺ <Peter Desmet> (7 weeks ago)
| | * | | 7825a85 (origin/gh-pages) Use enram/s3-bucket-listing template (Bootstrap) <Peter Desmet> (7 weeks ago)
| | * | | 25ab64e Remove local list.js, link to https://enram.github.io/s3-bucket-listing/list.js file <Peter Desmet> (7 weeks ago)
| | * | | 4e6d013 Update index.html with enram bucket info <Peter Desmet> (7 weeks ago)
| | * | | 4507465 Copy index.html and list.js from s3-bucket-listing <Peter Desmet> (7 weeks ago)
| * | | | 646fe7f cleaning up <Plieper> (7 weeks ago)
| | |_|/
| |/| |
* | | | 7e92871 remove bioRad vignettes <Plieper> (7 weeks ago)
| | |/
```

There are other applications you can use to use Git*

* Github desktop, GitKraken, integration in IDE environments (Rstudio, Eclipse,...),...

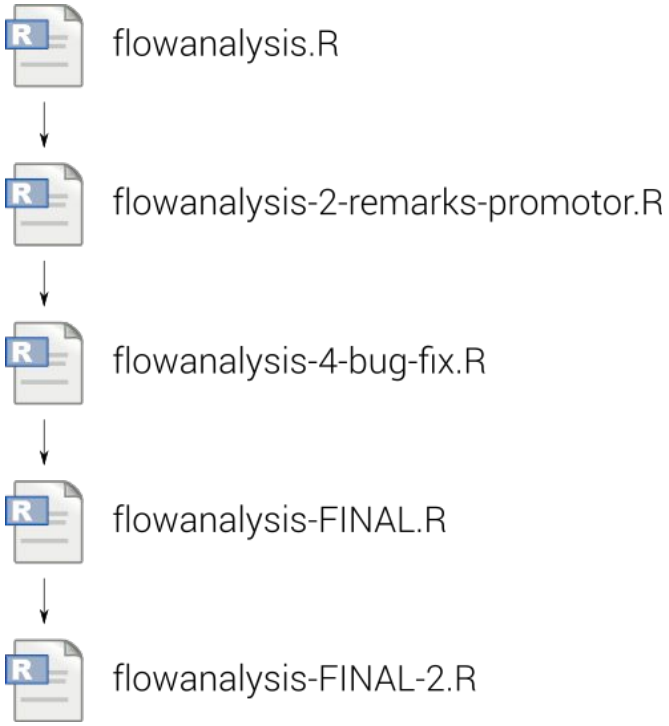
First,

TERMINOLOGY

1. Tell the story of your project

You use Git to take snapshots of all the files in a folder. This folder is called a repository or repo.

When you want to take a snapshot
of a file or files, you create a commit


flowanalysis.R
flowanalysis-2-remarks-promotor.R
flowanalysis-4-bug-fix.R
flowanalysis-FINAL.R
flowanalysis-FINAL-2.R

by saving copies

by making commits

by saving copies

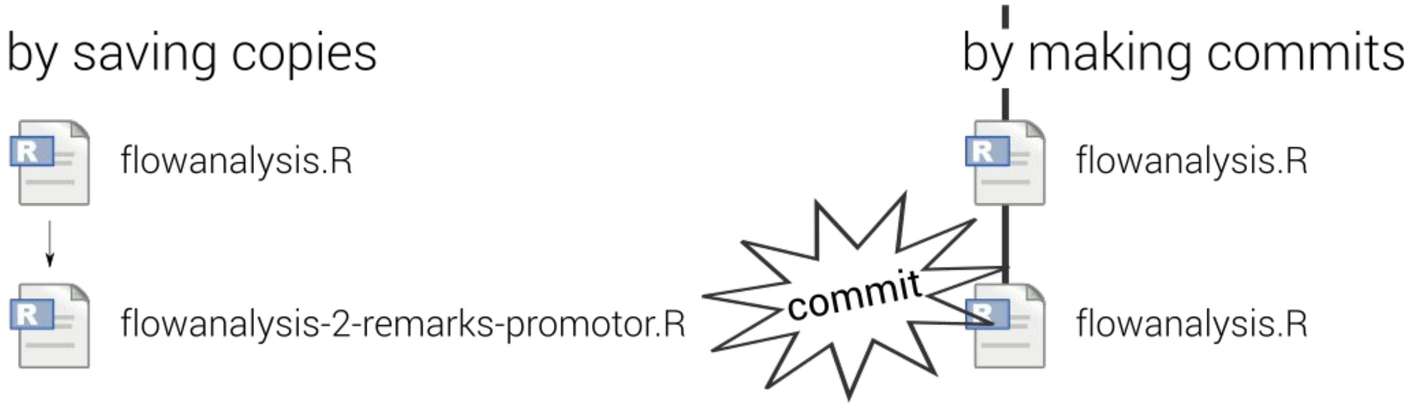
flowanalysis.R

by making commits

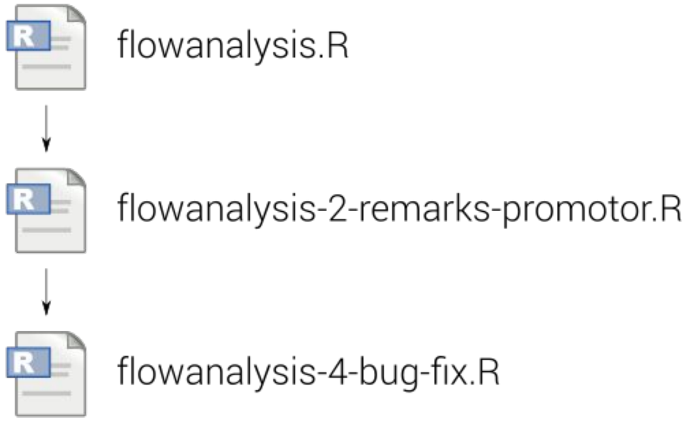
flowanalysis.R

by saving copies

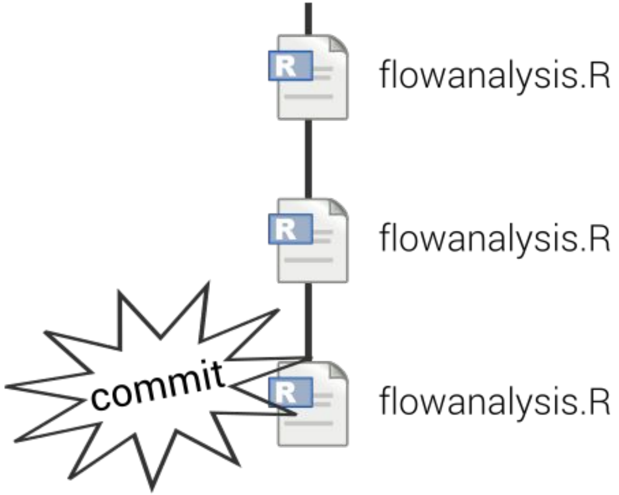
by making commits



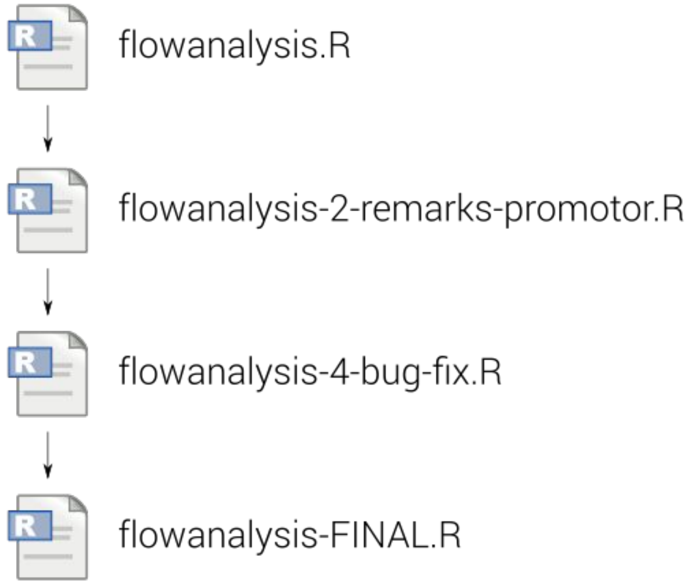
by saving copies



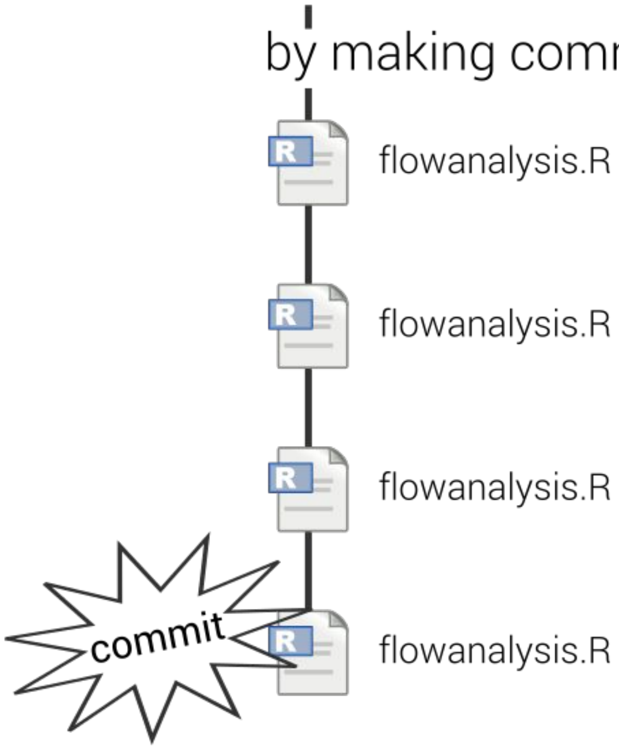
by making commits

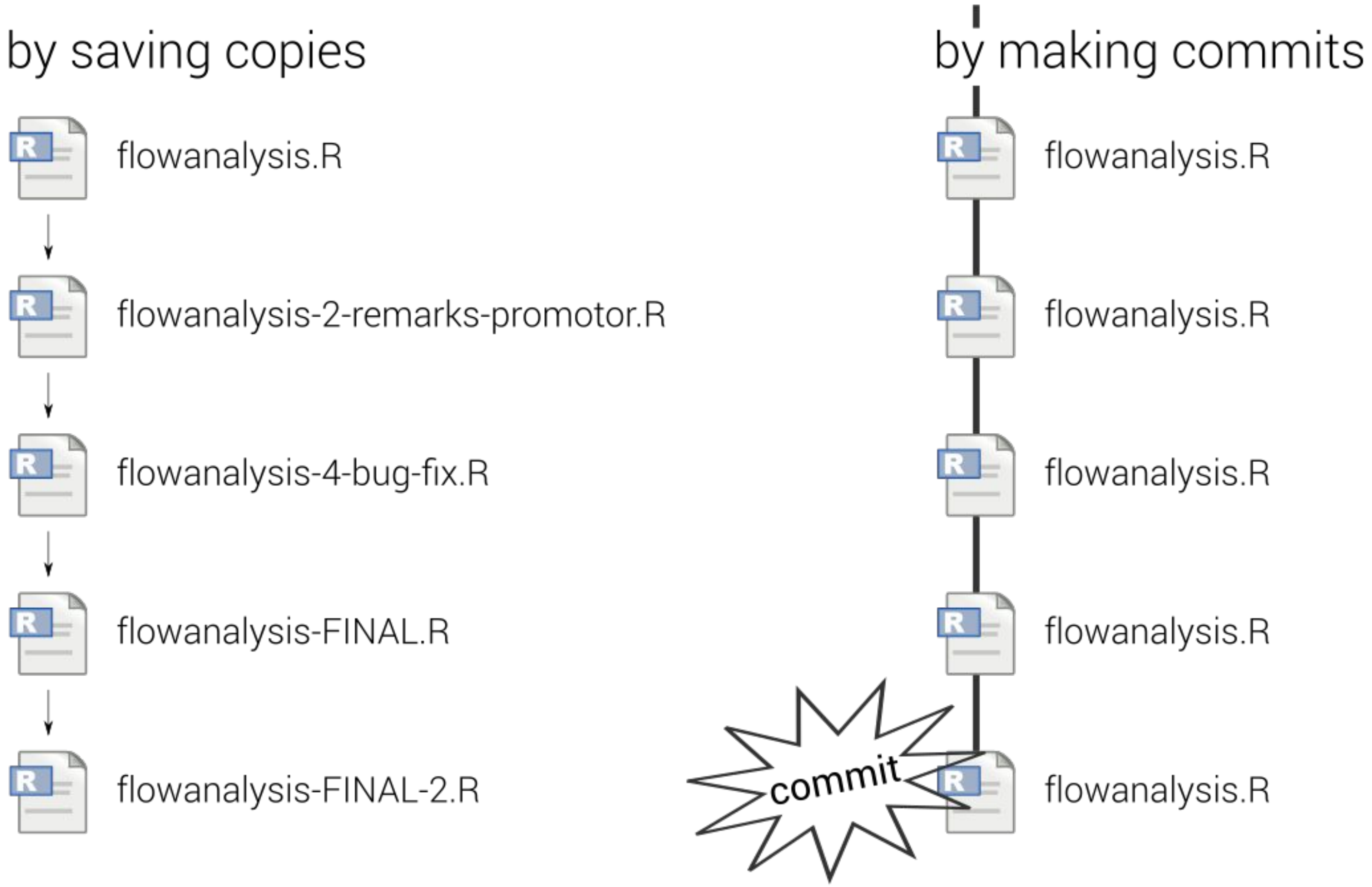


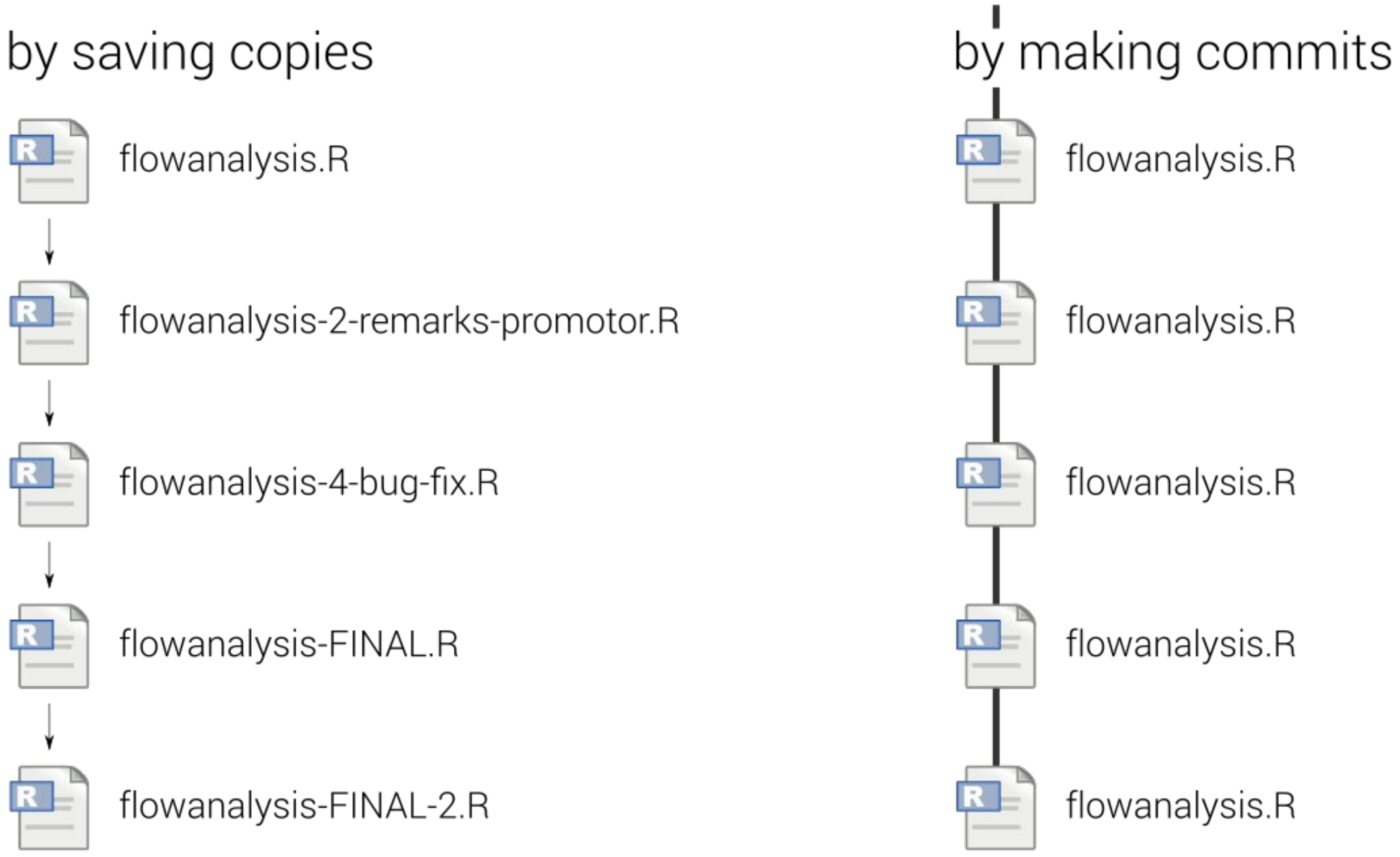
by saving copies



by making commits




by saving copies
flowanalysis.R
flowanalysis-2-remarks-promotor.R
flowanalysis-4-bug-fix.R
flowanalysis-FINAL.R
flowanalysis-FINAL-2.R
by making commits
flowanalysis.R
flowanalysis.R
flowanalysis.R
flowanalysis.R
commit
flowanalysis.R


by saving copies
flowanalysis.R
flowanalysis-2-remarks-promotor.R
flowanalysis-4-bug-fix.R
flowanalysis-FINAL.R
flowanalysis-FINAL-2.R
by making commits
flowanalysis.R
flowanalysis.R
flowanalysis.R
flowanalysis.R
flowanalysis.R

When you commit a file or files, some information is saved along with the changes of the file

1. Who
2. When

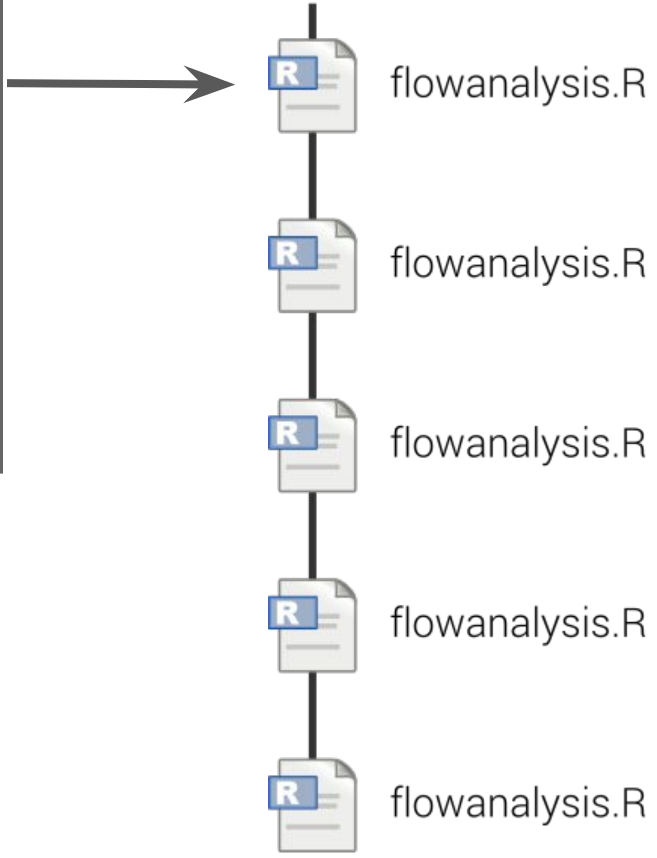
You can add more information about the changes you’ve made in a commit message

A good commit message:

```
Extend season handling

Dependent of the use-case, the
meteorological or astrological
handling of season start and end
dates is required. This commit
provides support for both use
cases as class property
```

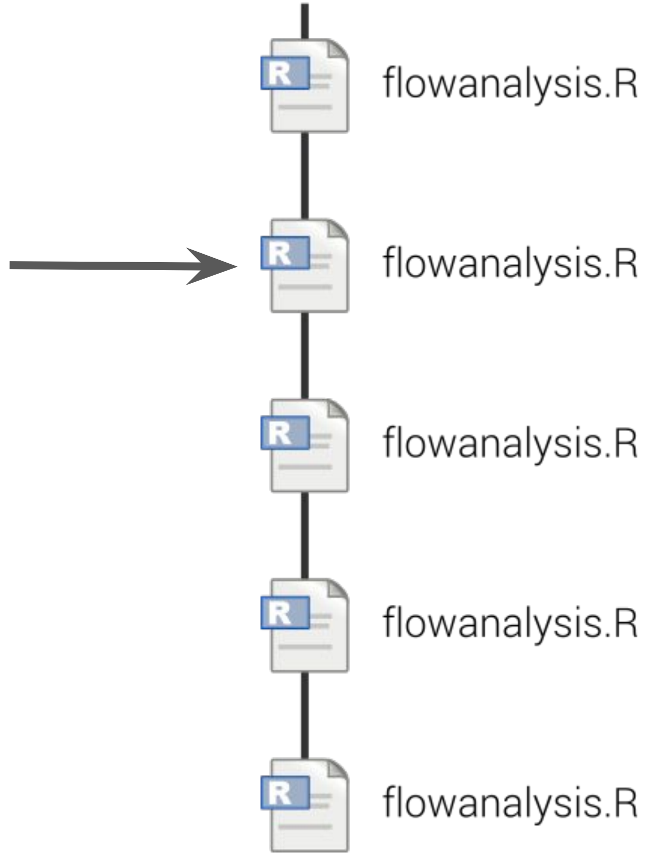
by making commits

```
Stijn Van Hoey
11:08am September 2nd 2016

Adapt plot date label handling

Change the default handling of
xlabel date values vertical
printing into multiline date
labels
```



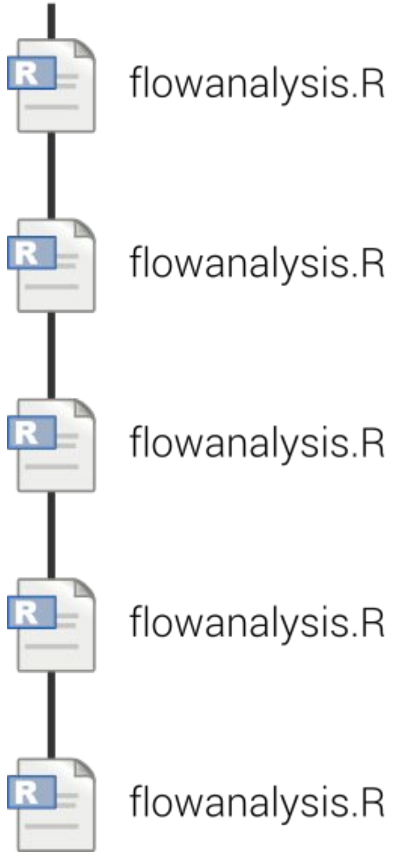
by making commits

```
Stijn Van Hoey
09:20am September 10th 2016

Extend season handling

Dependent of the use-case, the
meteorological or astrological
handling of season start and end
dates is required. This commit
provides support for both use
cases as class property
```



by making commits

```
Stijn Van Hoey
3:06pm December 15th 2016

Add horizon plot

The horizon plot enables to
quickly compare the flow during
time for multiple stations at
once. By overlapping, color
intensity represents peak values
and low values are more clearly
visualised
```



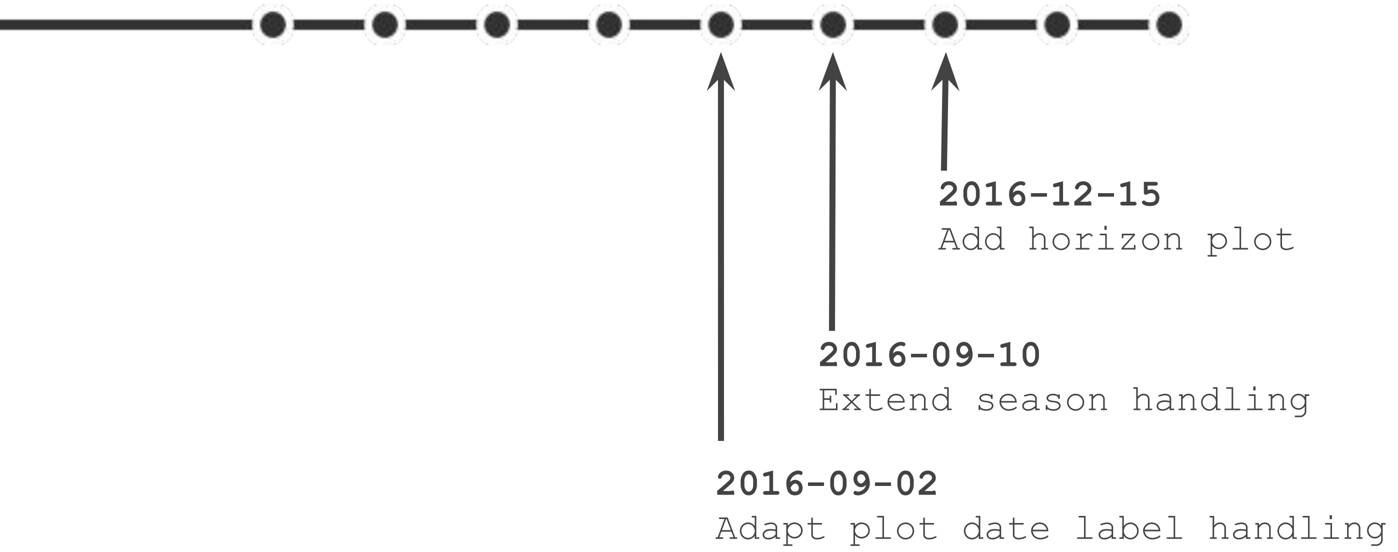
A good commit message:

GUIDELINES:

1. Separate subject from body with a blank line
2. Limit the subject line to 50 characters
3. Capitalize the subject line
4. Do not end the subject line with a period
5. Use the imperative mood in the subject line
6. Wrap the body at 72 characters
7. Use the body to explain what and why vs. how

When you want to include a previously nonexistent file to the commit, you need to add the file before committing

Git stores the whole history of your project

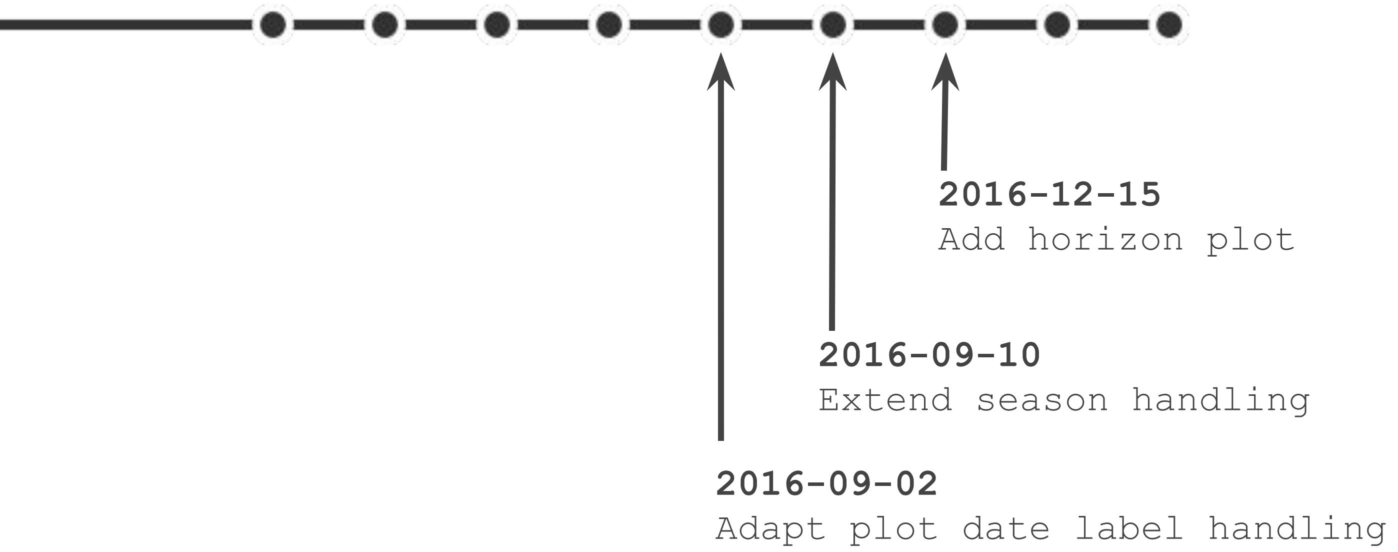


repository - your project folder
add - add a new file history
commit - save a snapshot

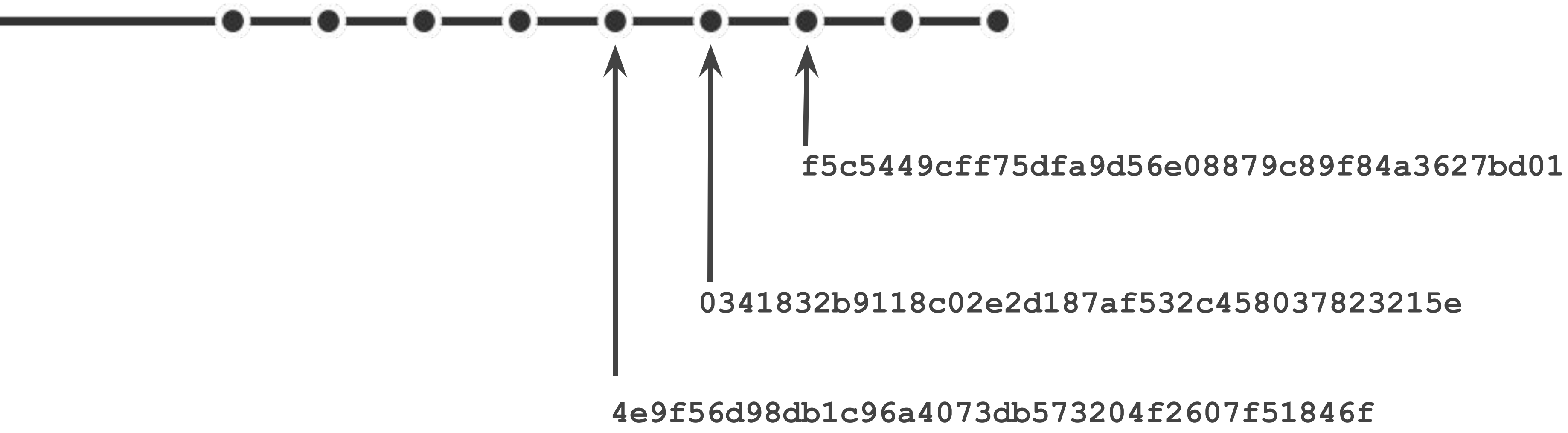
2. Travel back in time

Once you’ve saved some snapshots,
Git lets you move through them

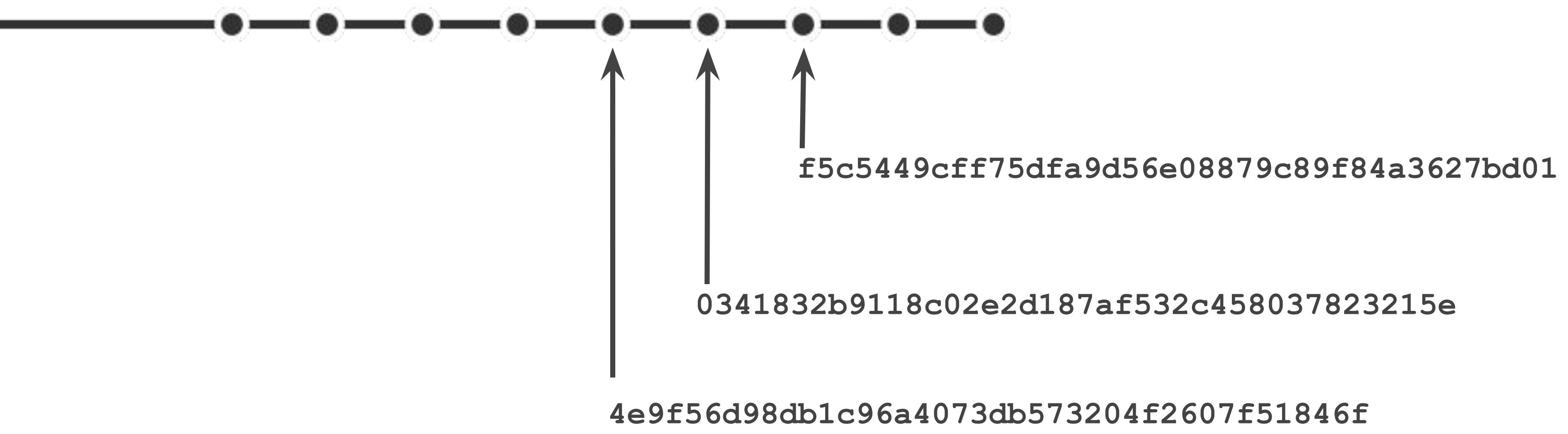
Git stores the whole history of your project



Each of these commits has an id called a hash

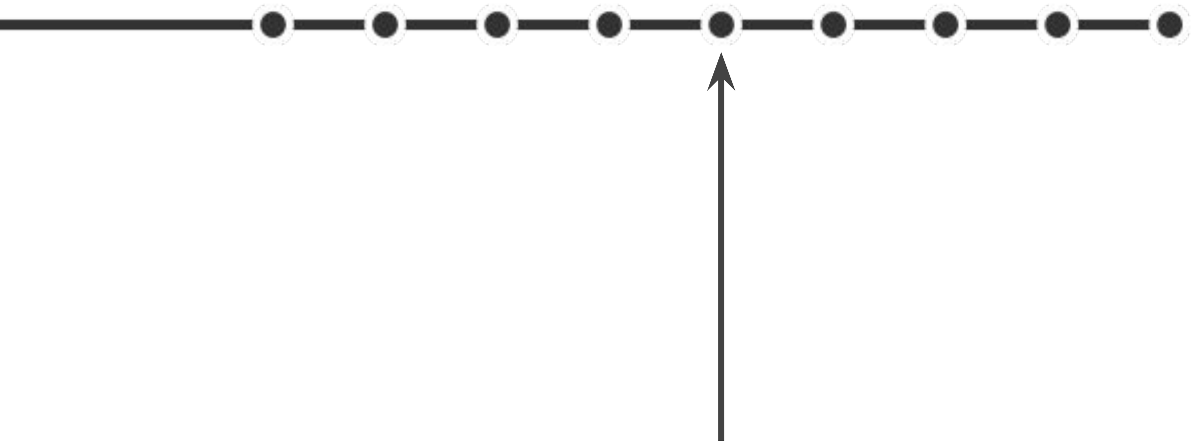

f5c5449cff75dfa9d56e08879c89f84a3627bd01
0341832b9118c02e2d187af532c458037823215e
4e9f56d98db1c96a4073db573204f2607f51846f

You can tell Git what commit you want to check out using the commit hash


f5c5449cff75dfa9d56e08879c89f84a3627bd01
0341832b9118c02e2d187af532c458037823215e
4e9f56d98db1c96a4073db573204f2607f51846f

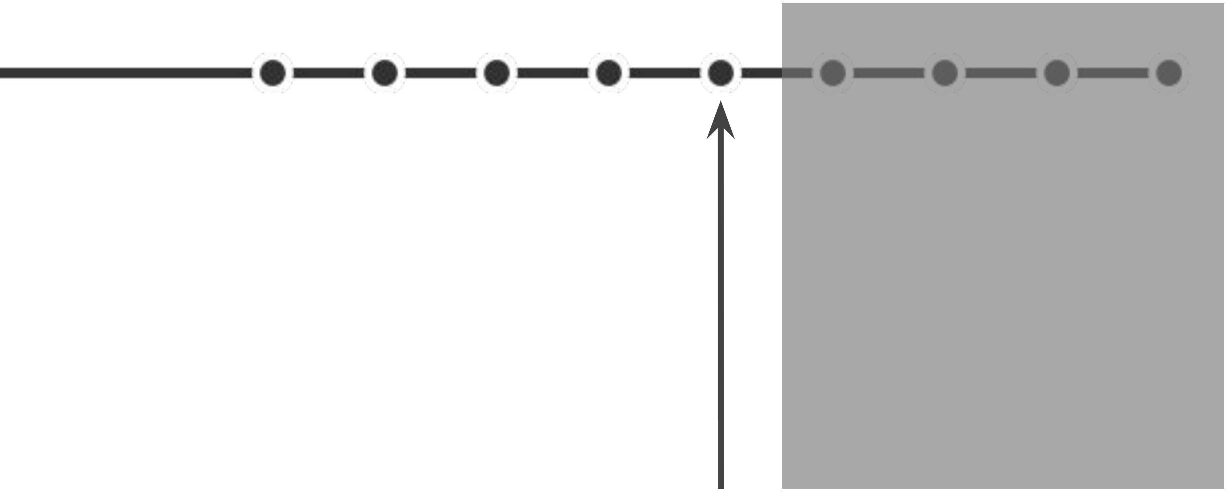
Getting the files from a commit in the past is known as doing a check out

You can tell Git what commit you want to check out using the commit hash



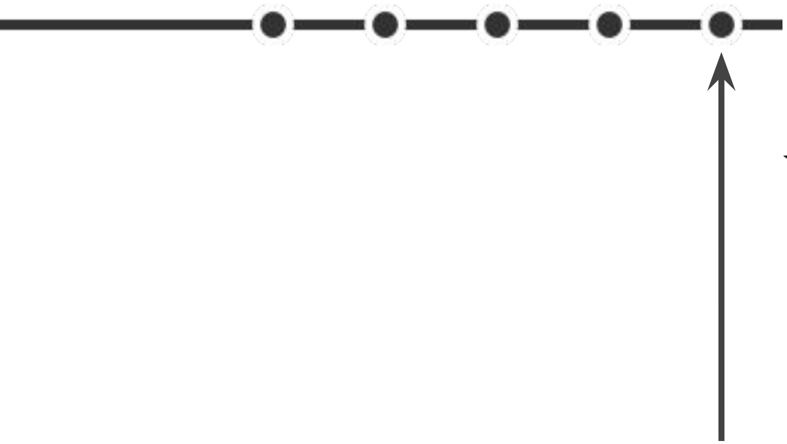
2016-09-02: `Adapt plot date label handling`
`4e9f56d98db1c96a4073db573204f2607f51846f`

Your other commits still exist, but when you look into your repo, it’s as if they never happened



2016-09-02: Adapt plot date label handling
4e9f56d98db1c96a4073db573204f2607f51846f

Your other commits still exist, but when you look into your repo, it’s as if they never happened



2016-09-02: Adapt plot date label handling
4e9f56d98db1c96a4073db573204f2607f51846f

No worries, everything still exists!

hash - a computer generated id
checkout - time travel to a specific commit

3. Experiment with changes

So far, everything has been very linear and ordered…

So far, everything has been very linear and ordered…

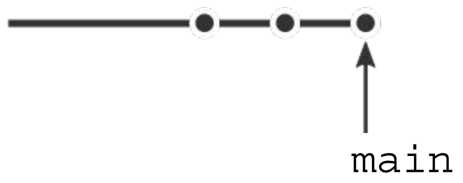
… science is not linear!

Within research projects, you want to make easily discardable experiments

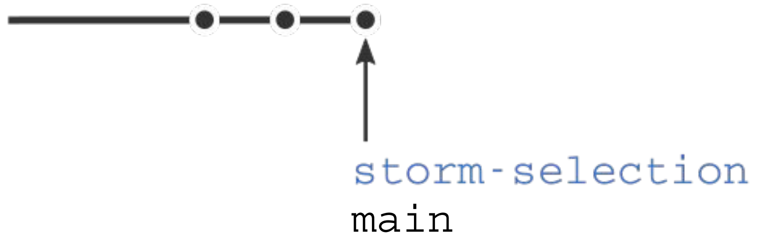
The way you do this in Git is with branches

A branch is a moveable label attached to a commit

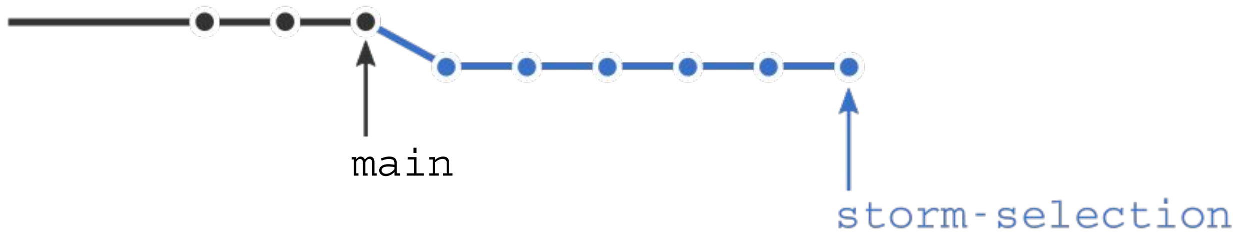
The default branch name in Git is main



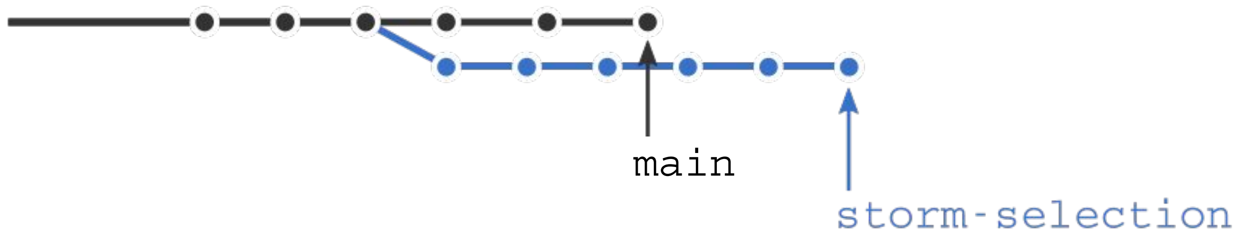
You can add your own branches too, with a chosen name



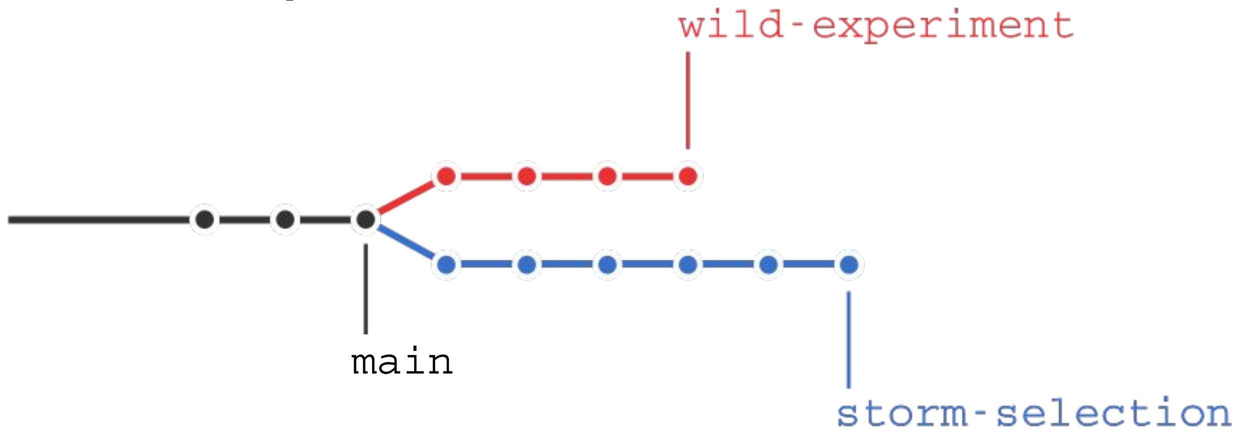
You will mostly do multiple commits on a branch while working



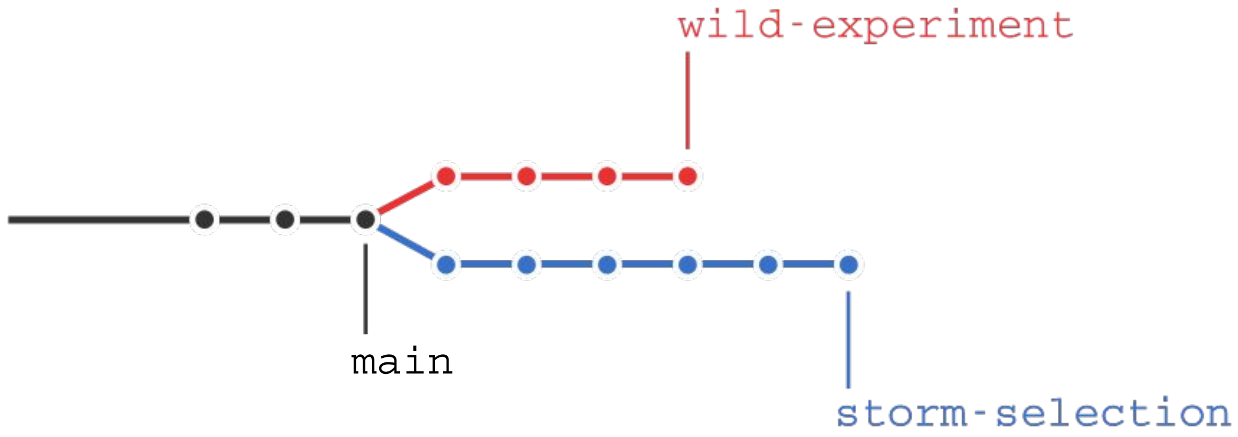
You can have multiple branches at the same time...



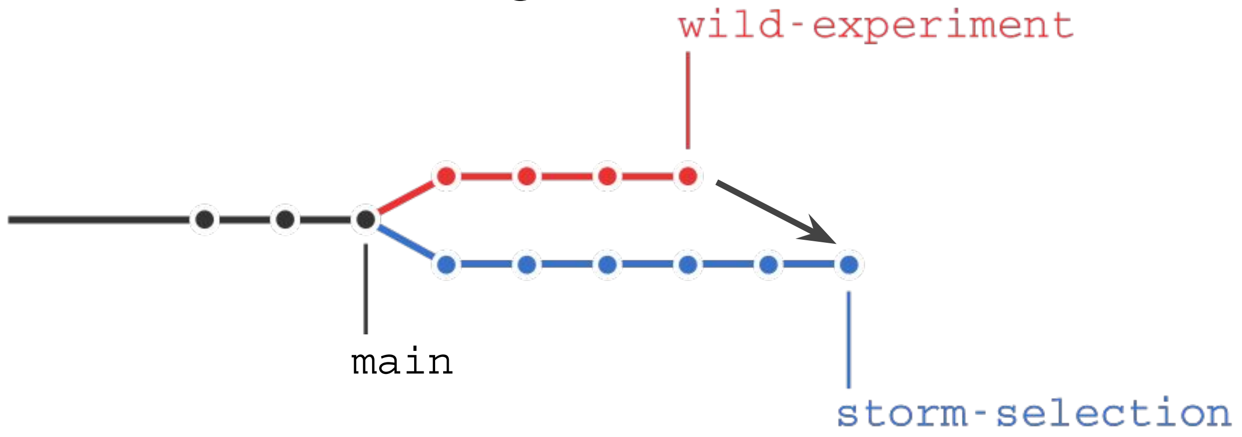
But it is good practice to treat your main branch *special*



Branches are *cheap* and ideal for trying out stuff



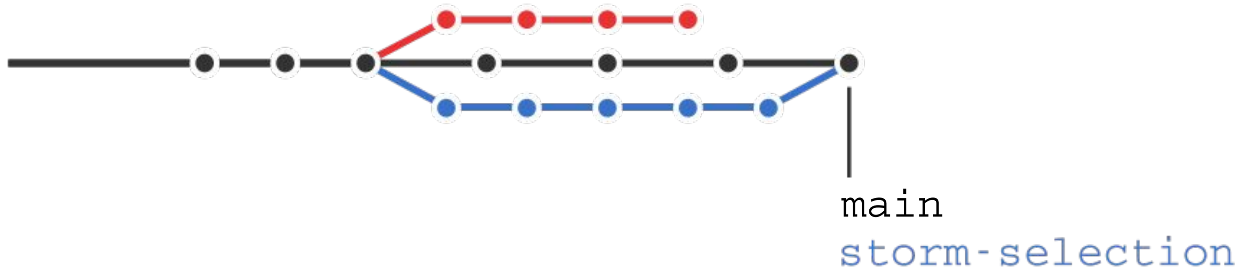
You can switch between branches using check out using the branch label



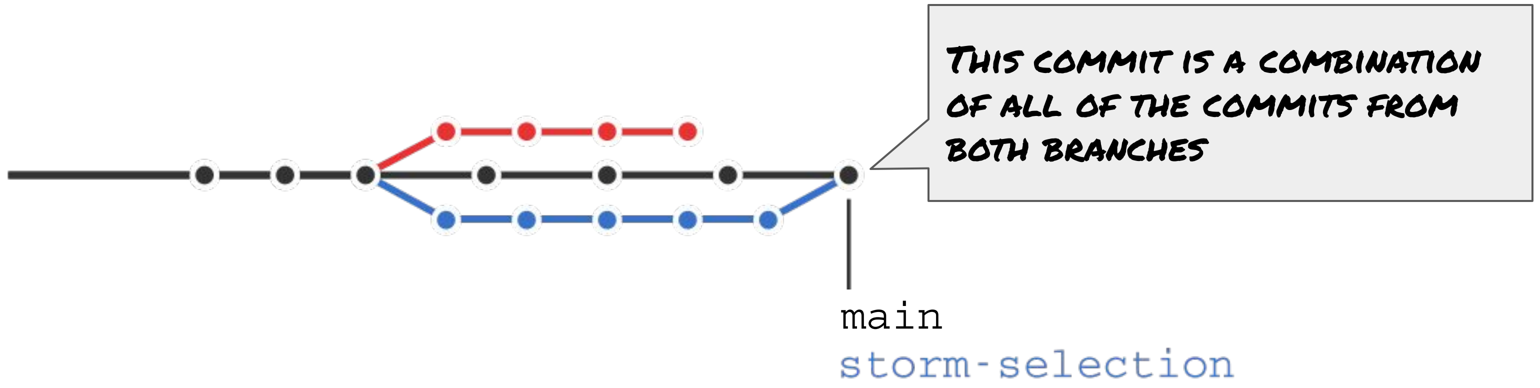
Branches contain work in progress

Once you're happy with some work, you need a way to get it back into main

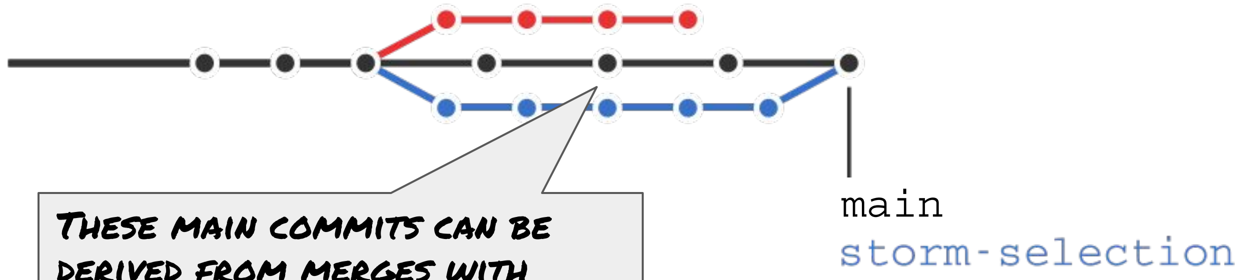
To get changes from one branch into another, you merge them



To get changes from one branch into another, you merge them


THIS COMMIT IS A COMBINATION OF ALL OF THE COMMITS FROM BOTH BRANCHES
main
storm-selection

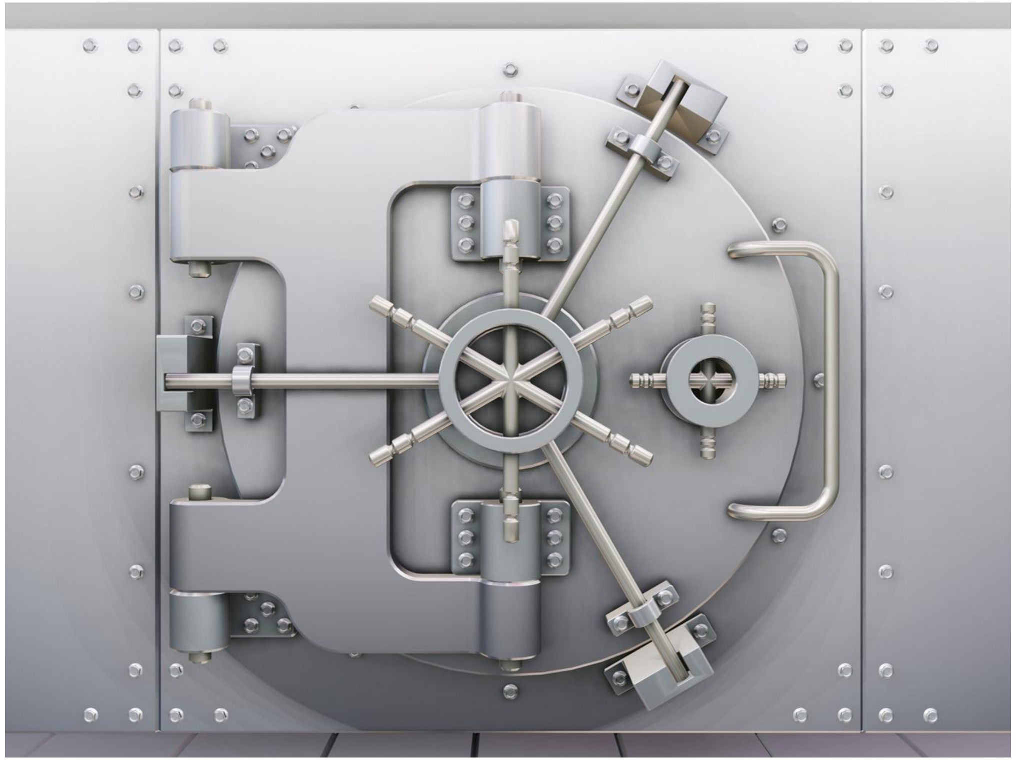
To get changes from one branch into another, you merge them



THESE MAIN COMMITS CAN BE DERIVED FROM MERGES WITH OTHER BRANCHES OR QUICK/CRUCIAL FIXES!

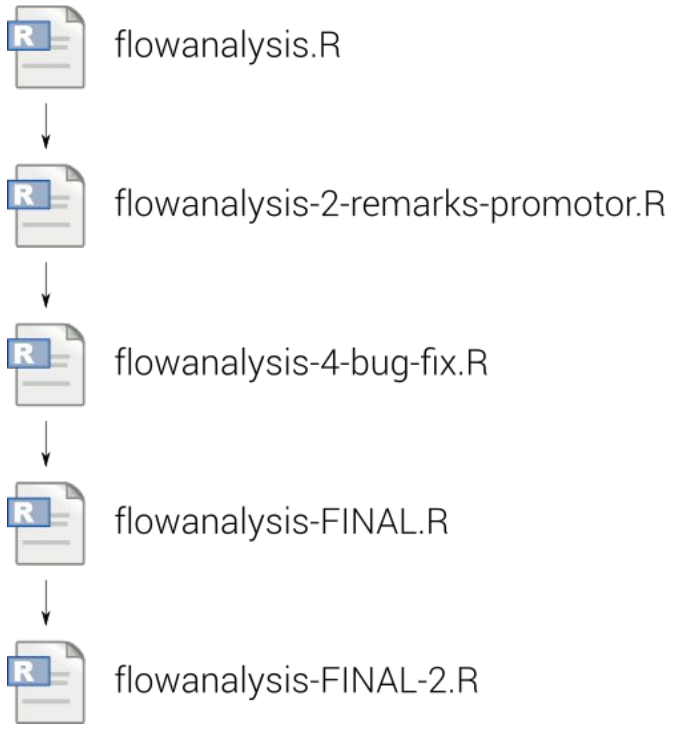
branch - a label that points to a commit
merge - the combination of two or more branches

4. Backup your work



Everyone knows that you should back up your work regularly

Ideally to somewhere that is geographically distinct from your computer

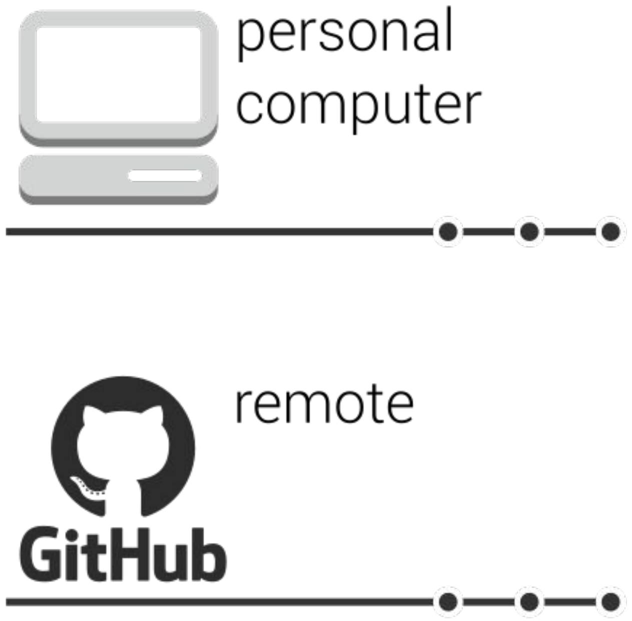


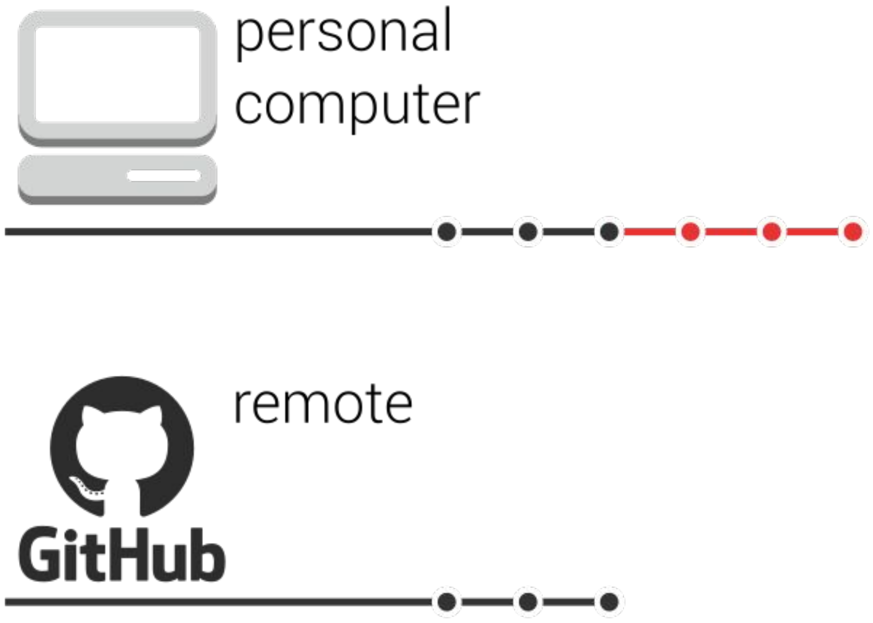
WHAT IS YOUR CURRENT BACKUP STRATEGY?

- Safer
- Access from different places
- Shared access

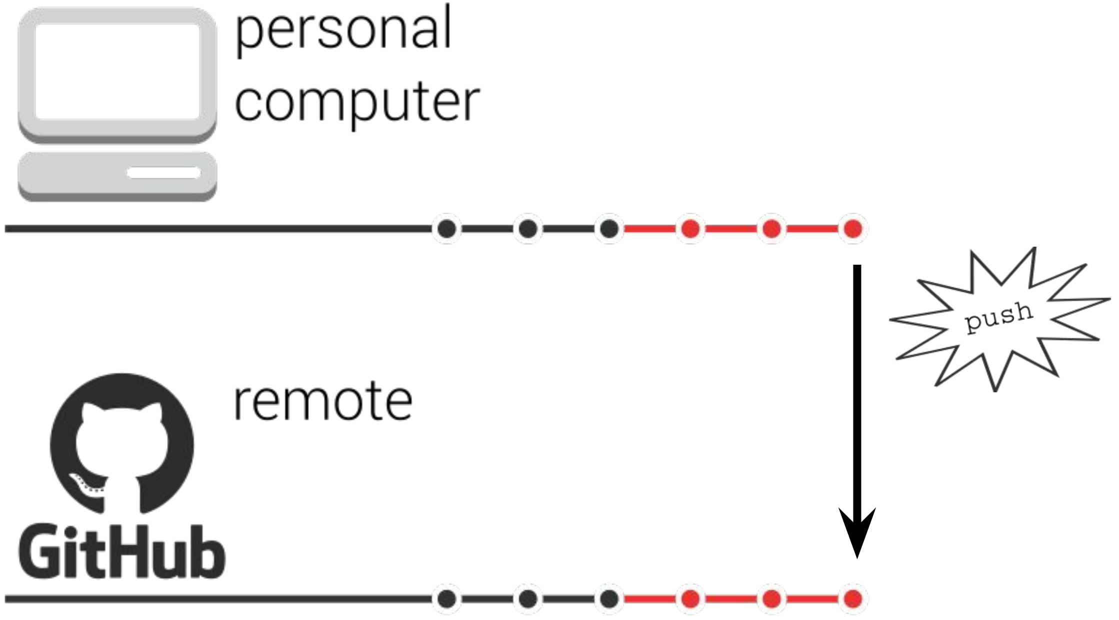
In Git this place is called a remote

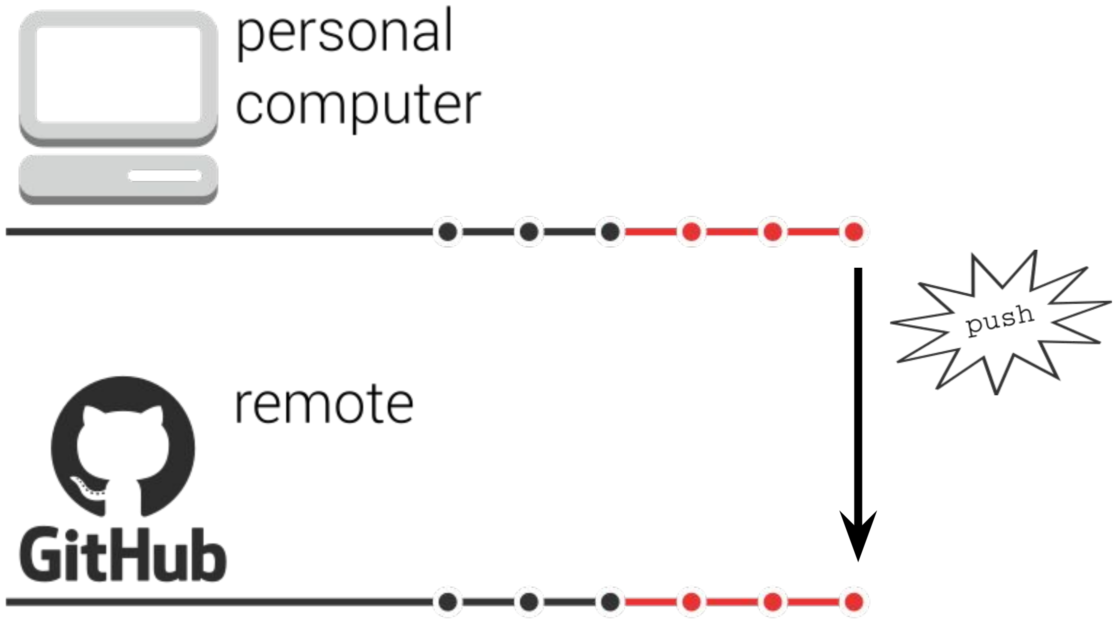
A very popular remote is GitHub


personal
computer
remote
GitHub

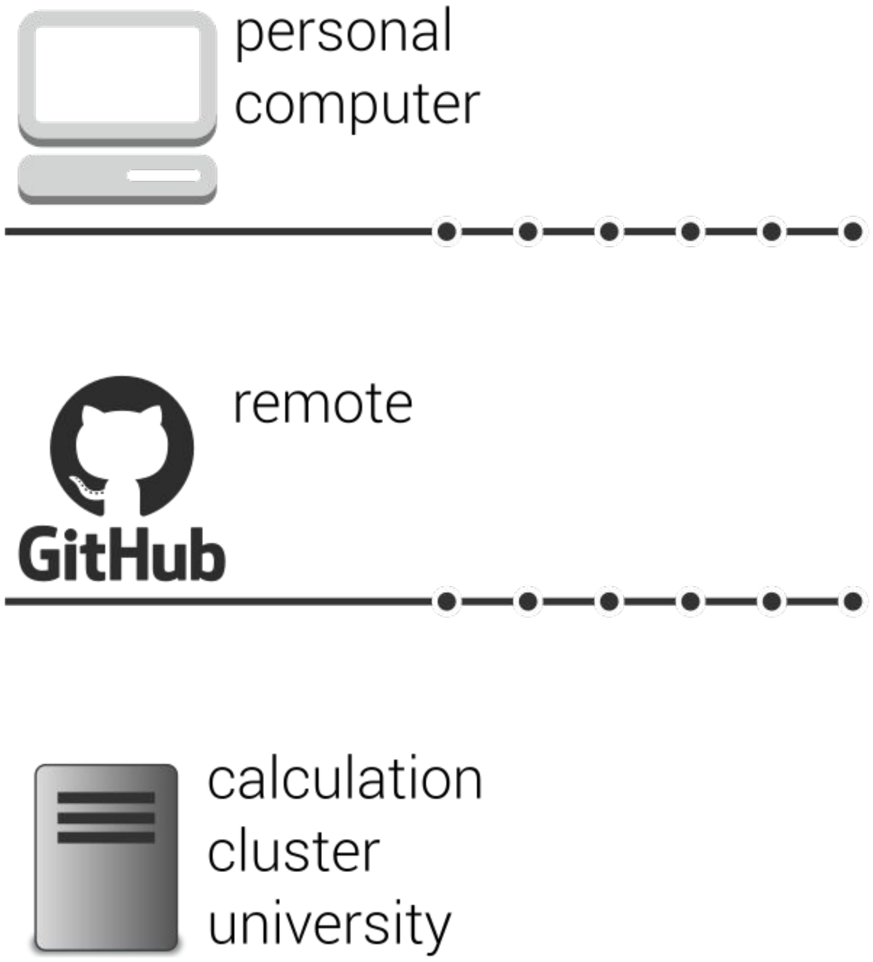


Whenever you developed code on your computer, you can send the changes to the remote

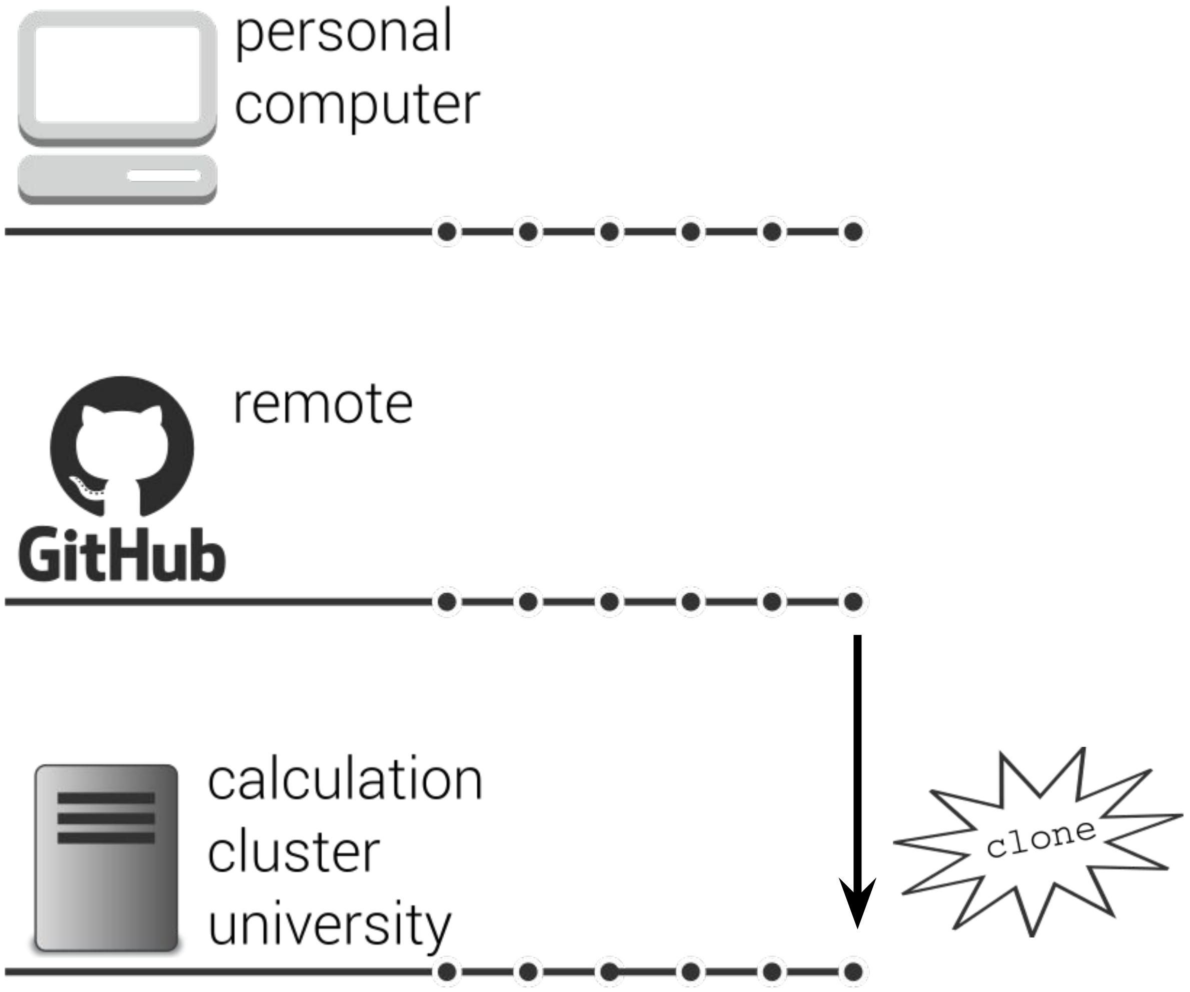

personal
computer
push
remote
GitHub

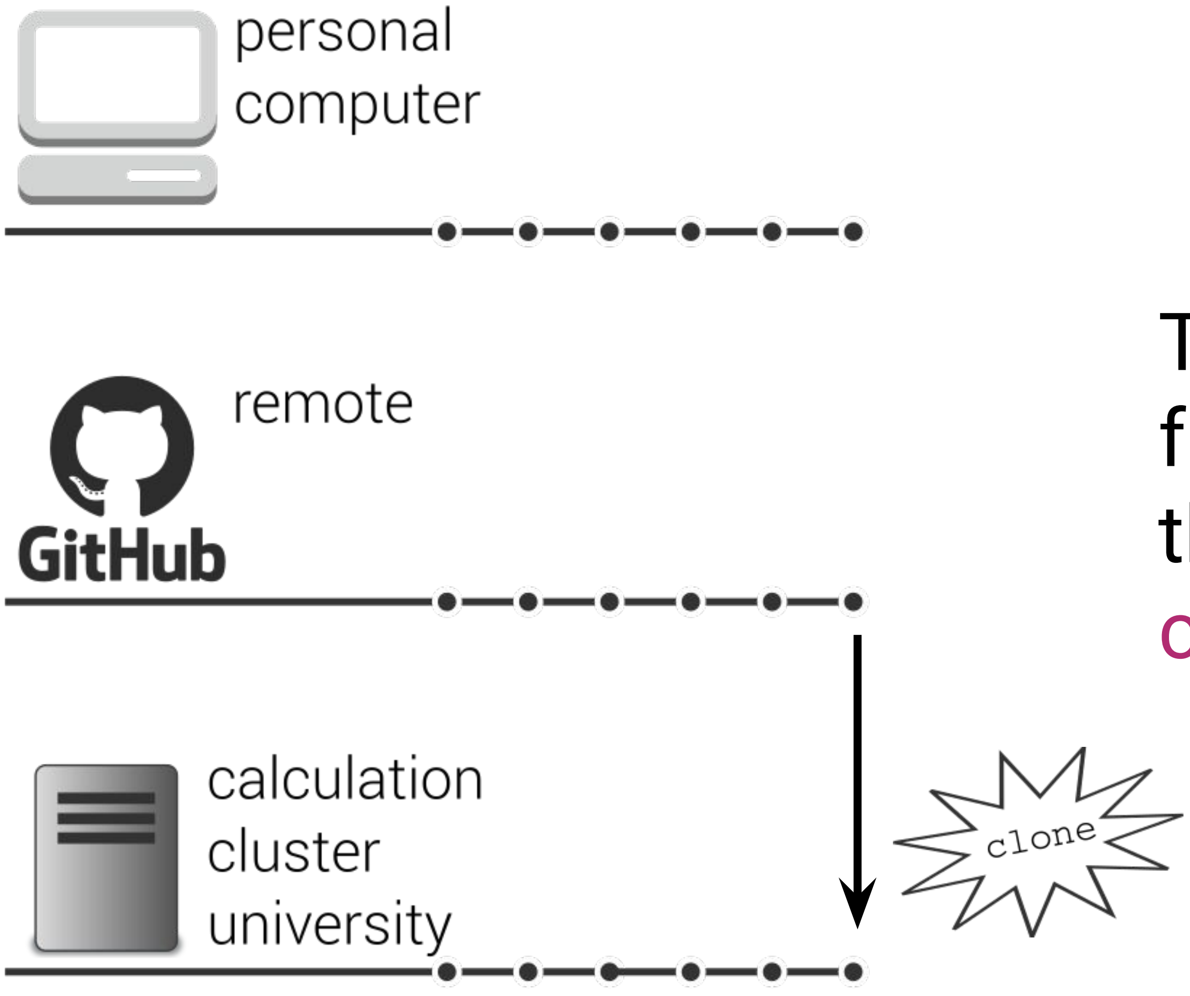


This is known as a push

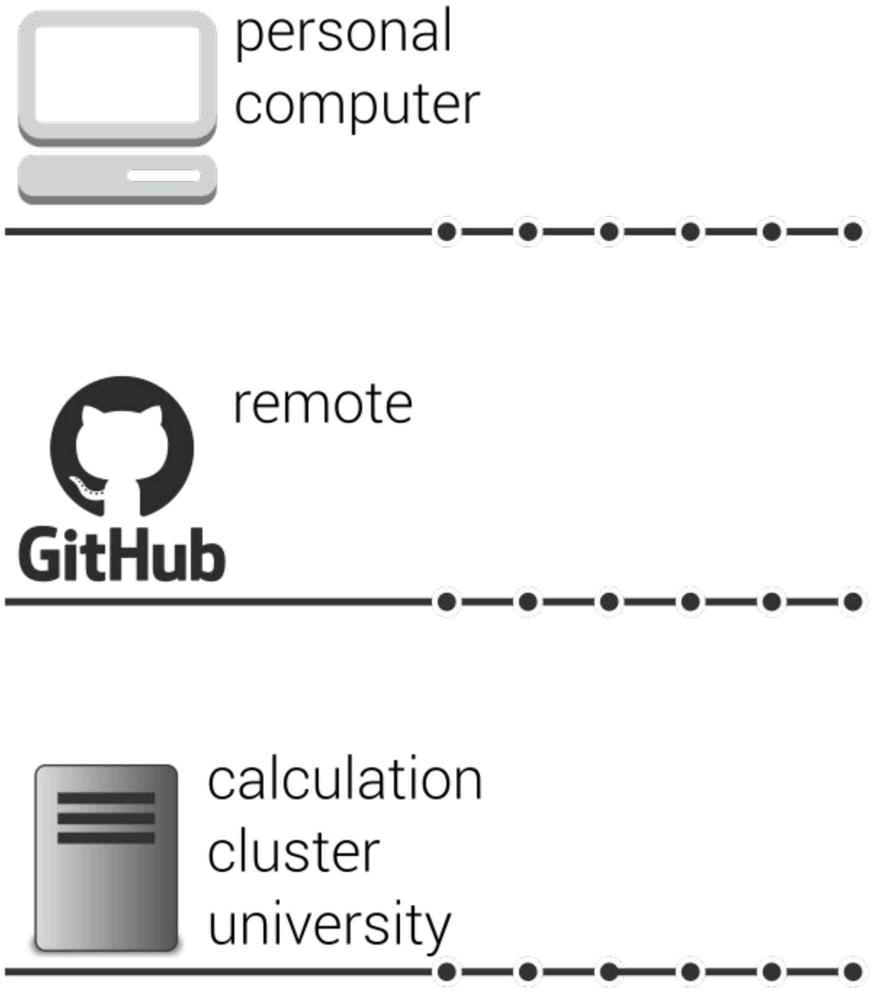


Consider the situation where you want to run your code on a remote server

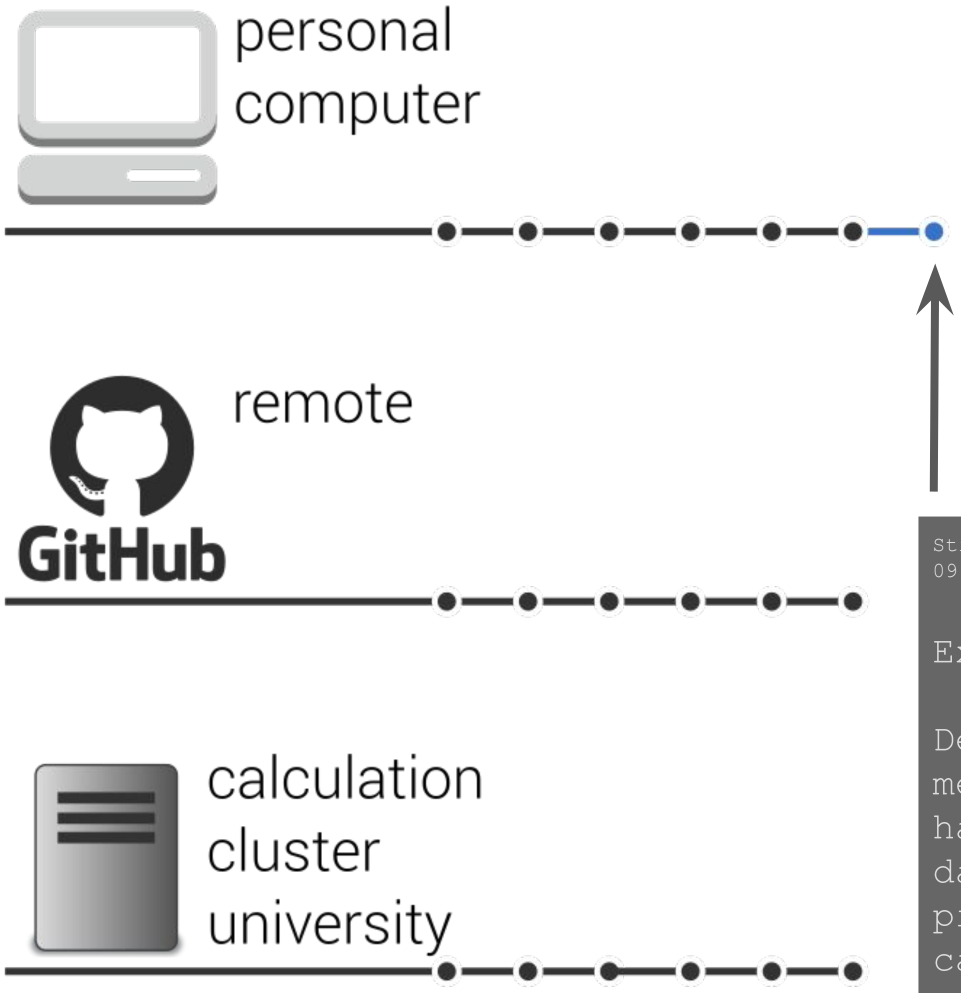

personal
computer
remote
GitHub
calculation
cluster
university
clone



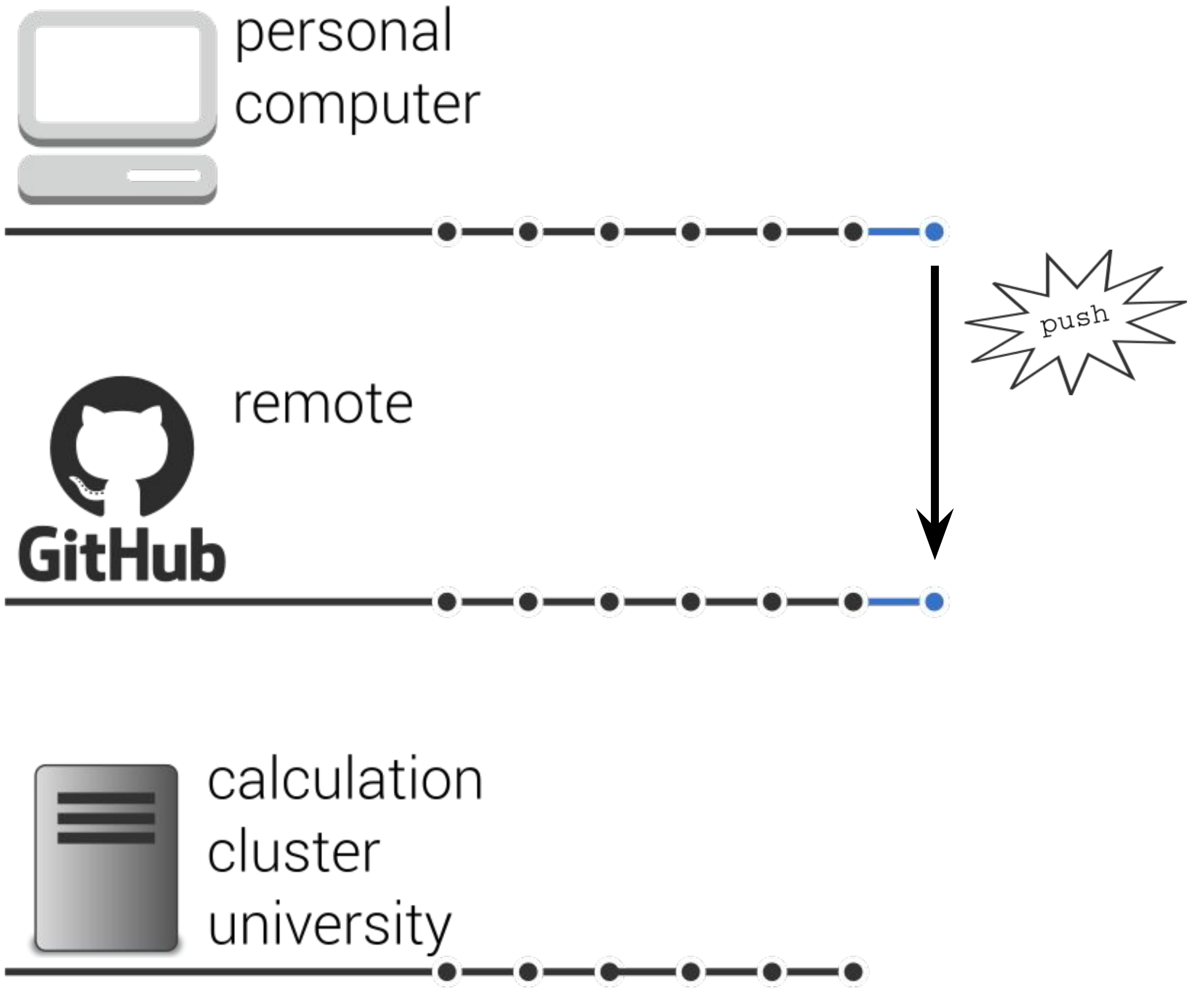
To get some work from a remote for the first time you clone it

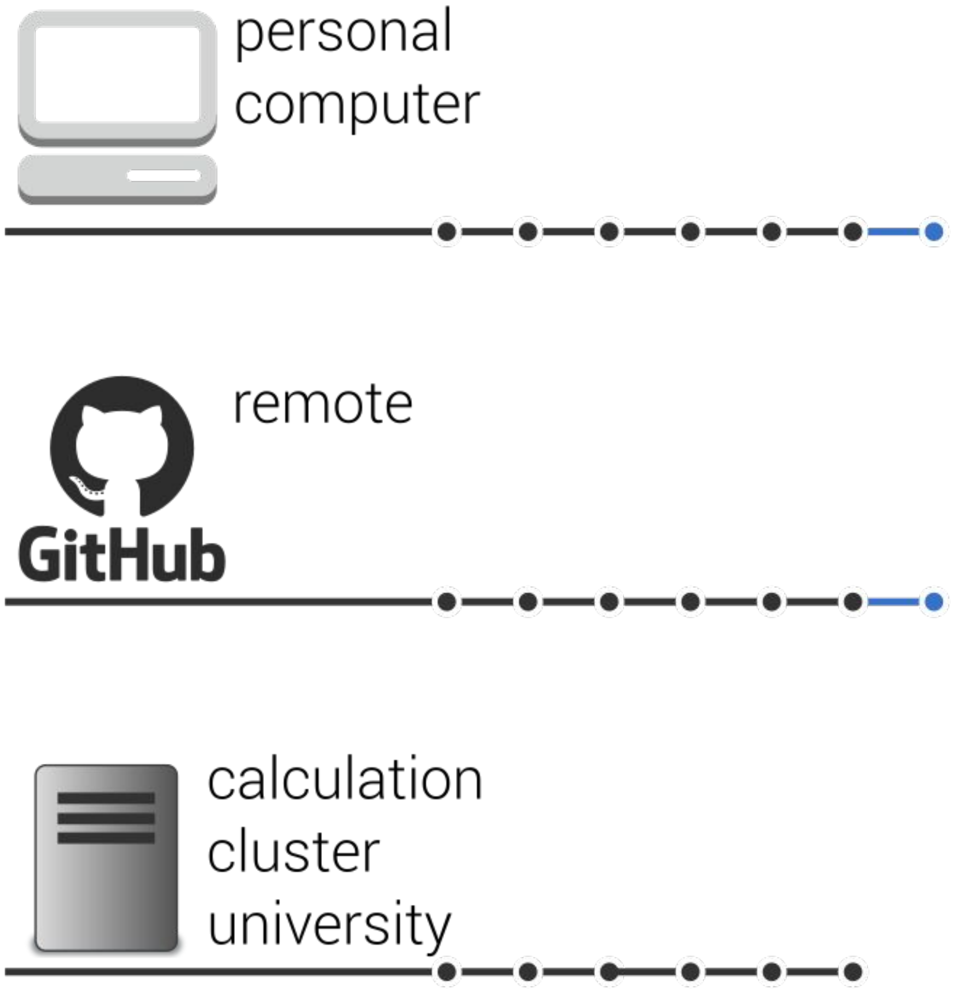


Now, both your computer and calculation cluster have the same repo

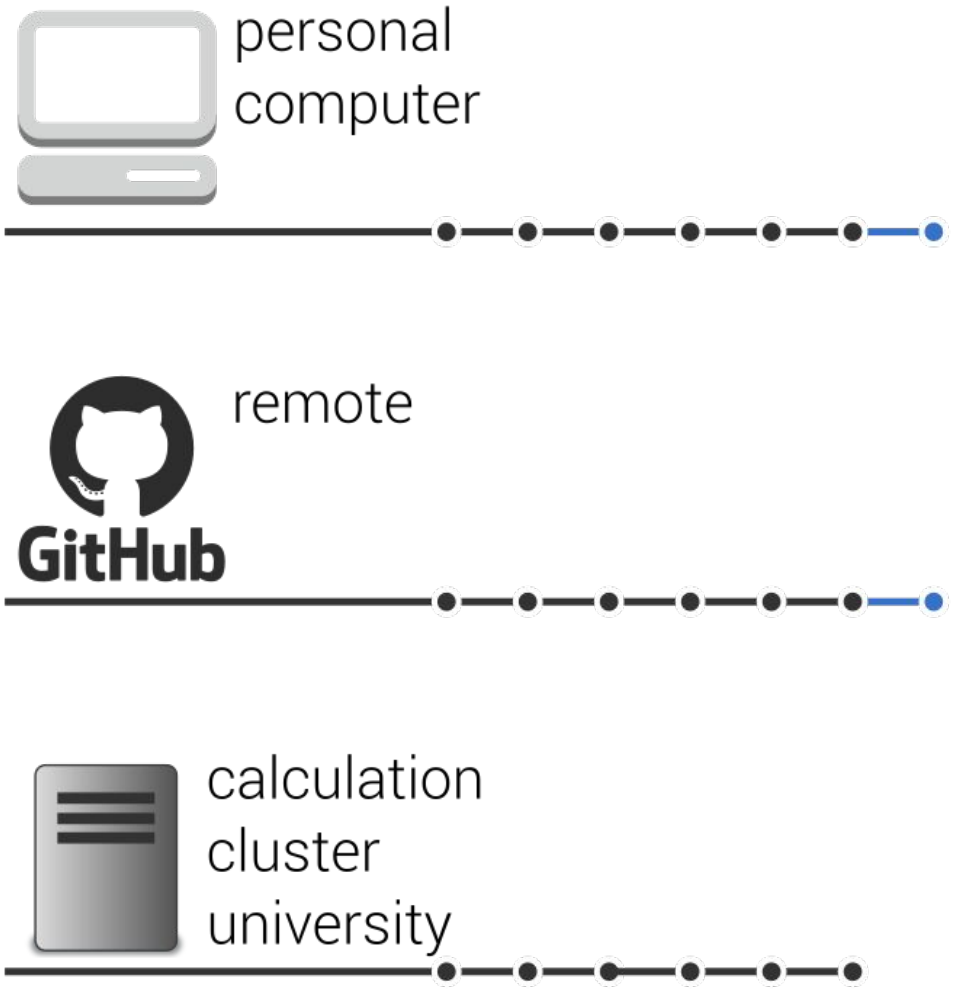


```
Stijn Van Hoey
09:20am September 10th 2016

Extend season handling

Dependent of the use-case, the
meteorological or astrological
handling of season start and end
dates is required. This commit
provides support for both use
cases as class property
```

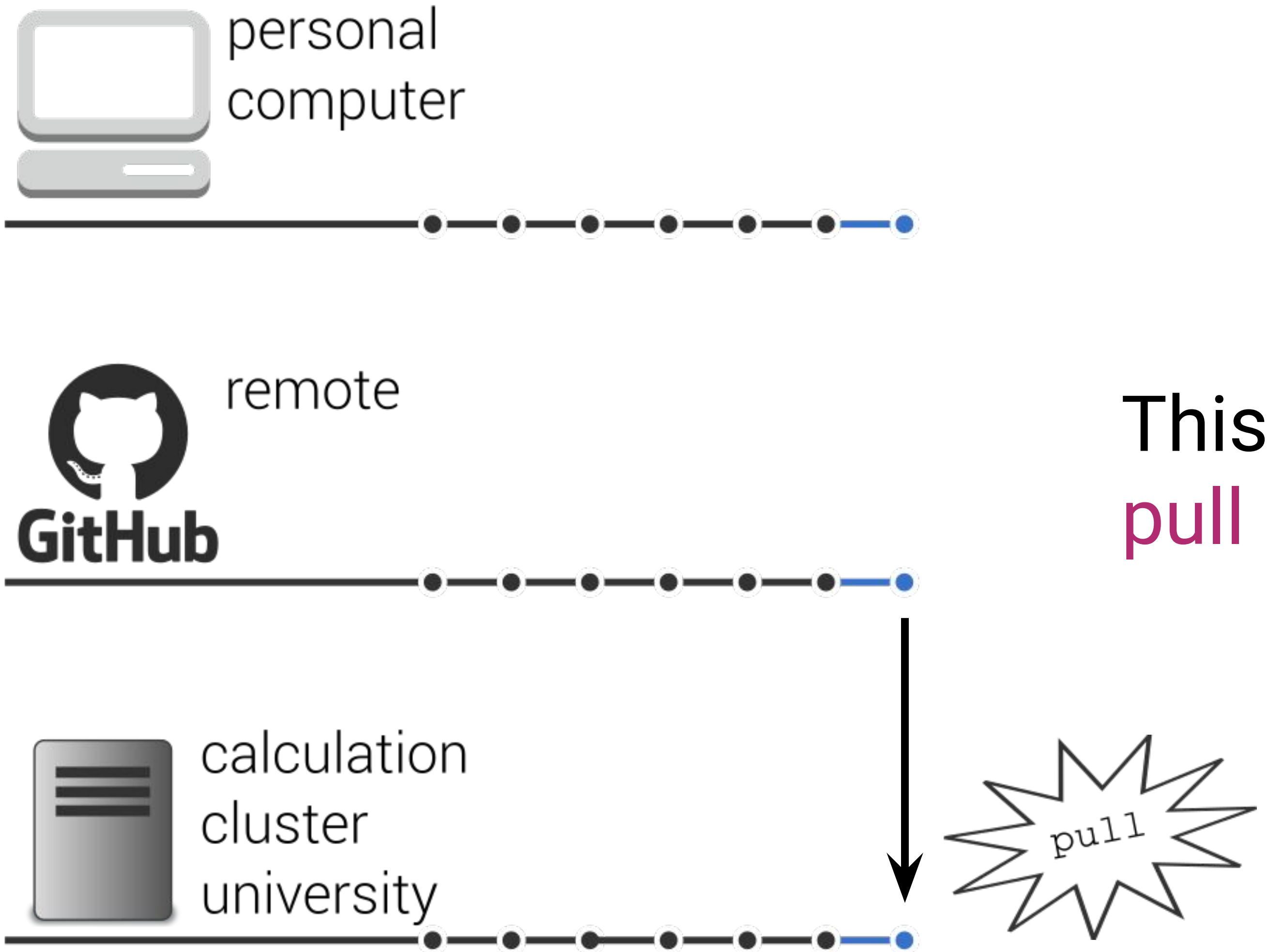

personal
computer
push
remote
GitHub
calculation
cluster
university



The code on the cluster does not contain the new feature



To get these changes, you need to download them on the cluster



This is known as a pull

remote - a server with a repo on it
clone - get the repo from the remote for the first time
pull - receive new commits from the remote
push - send your new commits to the remote

So far...

repository	your project folder
add	add a new file to the snapshot
commit	save a snapshot
hash	a computer generated id
checkout	time travel to a specific commit
branch	a moveable label that points to a commit
merge	the combination of two or more branches
remote	server with a repo on it
clone	get the repo from the remote for the first time
pull	receive new commits from the remote
push	send your new commits to the remote

Git is great with
text files
(.R, .py, .md,.tex,...)

We're not only working with text files...

Git is not suitable for...

- Large files (100s of MBs - GBs)
- Binary files
  (Pictures, Videos, Music, Office Documents, output binaries)

...for the moment, remember .gitignore

Version control...

1. Tell the story of your project git
2. Travel back in time git
3. Experiment with changes git
4. Backup your work git
5. Collaborate on projects git

What is your current backup strategy?

Git will not be your backup strategy, but will be part of your backup strategy

Acknowledgements:

- https://speakerdeck.com/alicebartlett/git-for-humans
- Benny Colyn 2016