Introduction to git

Peggy Lindner

Center for Advanced Computing and Data Systems

Introduction to git

Discuss

What is your current version control system?

1. How do you manage different file versions?
2. How do you work with collaborators on the same files?
3. How much would your science suffer if your workstation exploded right now? (scale from 1-10)

What is git

Version control system

- manage different versions of files
- collaborate with yourself
- collaborate with other people

Version control software keeps track of every modification to the code in a special kind of database.

- If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.
- other version control systems are svn, cvs, mercurial

Why use git

"Always remember your first collaborator is your future self, and your past self doesn't answer emails"

Christie Bahlai

What is git good for?

- backup
- reproducibility
- collaboration
- organization
- transparency

What you get

Tour of a git repository

Overview

1. Git on the command line
2. Git in SourceTree
3. Github vs. GitLab vs. Bitbucket for remote mirroring

Command line git

Make a directory with a small bash script, e.g.

```
mkdir test
cd test
vi example.sh
```

```
#!/bin/bash
#Declare array with 4 elements
ARRAY=( 'Debian Linux' 'Redhat Linux' Ubuntu Linux )
# get number of elements in the array
ELEMENTS=${#ARRAY[@]}

# echo each element in array
# for loop
for (( i=0;i<$ELEMENTS;i++)); do
    echo ${ARRAY[${i}]}
```

Initializing a repository

Prerequisites:

- git installed (check with `which git`)
- git configured (check with `git config --list`)

```
git config --global user.name ”Vlad Dracula”
git config --global user.email ”vlad@tran.sylvan.ia”
git config --global color.ui ”auto”
git config --global core.editor ”nano”
```

Initializing a repository

```
git init
```

Notice the `.git/` directory

Checking repository status

```
git status
```

Adding your file

```
git add example.sh
```

or, to add everything

```
git add --all
```

Your changes are now “staged””

(Image from Software Carpentry)

Committing

Changes aren’t final until they’re committed

```
git status
```

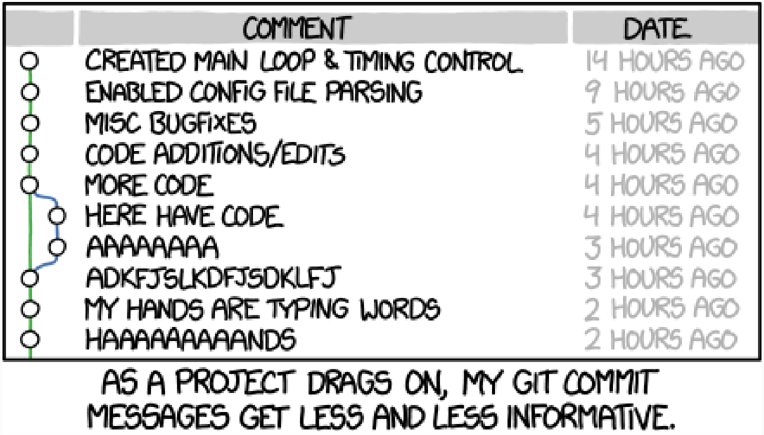
Committing

Once you’re sure that you’re changes are worth saving

(THIS WILL GO ON YOUR PERMANENT RECORD)

```
git commit -m ’changed x, y, and z’
```

Commit messages

- Describe why and the what “in a nutshell”
- Note to your future self (and to anyone else who you’re collaborating with)
- 

What did we do?

```
git status
git log
```

Make another change

1. Change file
2. Add changes
3. Commit changes
4. View updated log

Now, do something really stupid

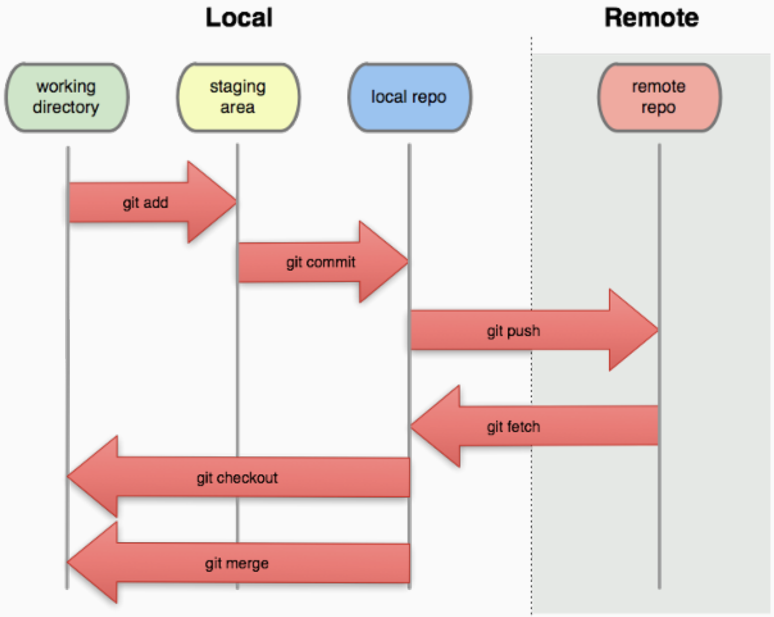
“Accidentally” introduce some errors to your file

Woops

Not that this ever happens…

```
git diff
git checkout HEAD your_file.R
```

What happened?



Wait, what does HEAD refer to?

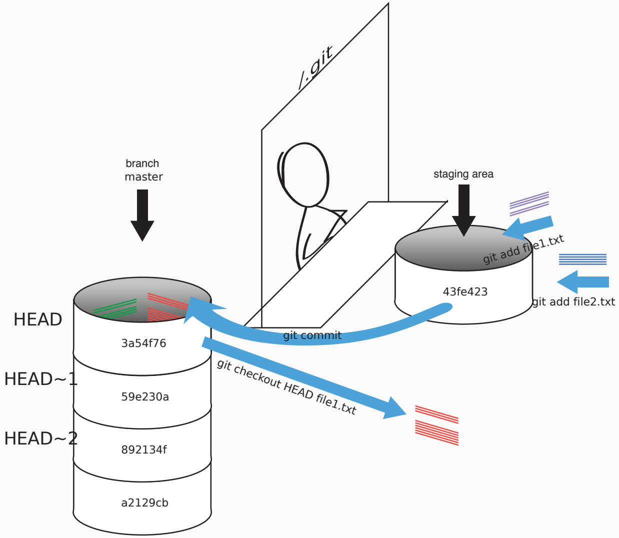


Figure 3: Commits ≈ a stack of heads

What if you really screw up?

A git choose your own adventure

`http://sethrobertson.github.io/GitFixUm/fixup.html`

Mirroring your repository on the internet

Setting up a “remote”

1. Create repository on Githubwith no .gitignore, no README, and no license
2. Add that as a remote

```
git remote add origin https://www.github.com/user/test.git
```

How to check:

```
git remote -v
```

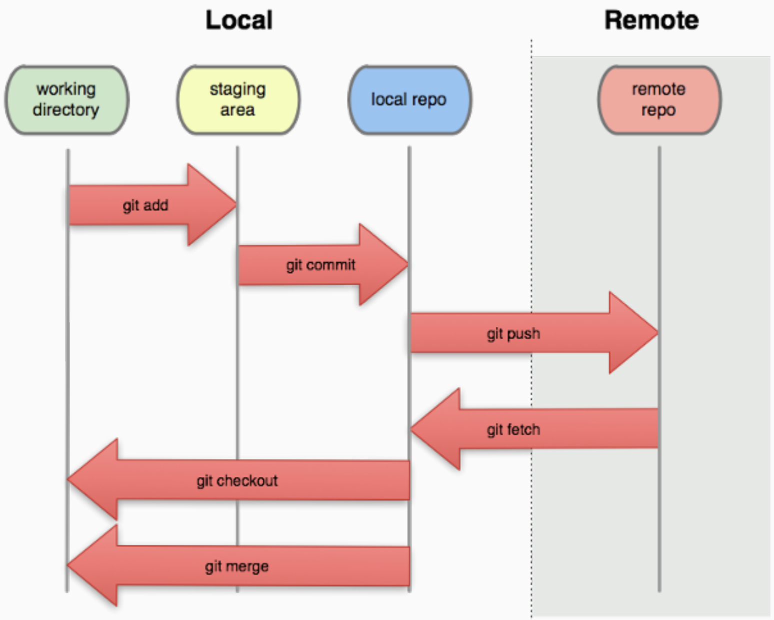
Once your repository has been linked to remote

Push your changes

```
git push -u origin master
```

Check the remote (Github or Bitbucket) to see new changes

Overview


Local
Remote
working directory
staging area
local repo
remote repo
git add
git commit
git push
git fetch
git checkout
git merge

Clone

- command copies an existing Git repository to your local machine

```
git clone git://cmake.org/cmake.git cmake.git
```

SourceTree’s interface

- Check out from your remote git repository

Demo of adding, committing, pushing

Github vs. GitLab vs. Bitbucket

Private repos:

- free on Bitbucket (w/ < 6 collaborators)
- free on GitLab (unlimited collaborators)
- not free on Github

Github vs. GitLab vs. Bitbucket

- all very similar
- Popularity & user base (4 vs. ?? vs. 1 million)
- free vs. pay
- open source vs. closed source

You can use all three if you want!

Additional resources

Motivation

- Ram K. 2013. Git can facilitate greater reproducibility and increased transparency in science.

Instruction

- Pro Git
- Software carpentry
- Git for scientists

Alternative interfaces

- GUIs for the command line averse