Distributed version control with git — a brief introduction

Andrei Chis

based on slides by Oscar Nierstrasz
Why version control?
Why version control?

Bob
Why version control?
Why version control?
Why version control?
Why version control?
Why version control?

A recipe for disaster!
Why version control?

Cope with the confusion that happens when multiple people edit the same files
Controlled evolution

Can still lead to disaster!
git
Tracks the history of a collection of files
git

Tracks the history of a collection of files

Can revert the collection of files to another version
git

distributed
version control
system
What is a distributed version control system?
What is a centralized version control system?
Bob

checkout

Central repository

Carol
Bob

commit

Central repository

update

Carol
Bob

commit

Central repository

Carol
you must update before every commit
What is a distributed version control system?
Remote repository (groupXY)
Bob

local repository

clone

Remote repository (groupXY)

Carol
Bob

local repository

remote repository (groupXY)

local repository

clone

clone

Carol
Bob

Remote repository (groupXY)

local repository

local repository

Carol
Bob

commit

local repository

Remote repository (groupXY)

local repository

Carol
Bob commit local repository

Remote repository (groupXY)

Carol

local repository

push

push
Bob -> local repository
commit -> local repository

push -> Remote repository (groupXY)
pull -> Remote repository

local repository
local repository

Carol
Bob commits to the local repository, which is then pushed to the remote repository (groupXY).
Bob commit local repository

Remote repository (groupXY)

local repository

push conflict

Carol commit local repository

commit
you must pull before every push
Remote repository (p2exercises)
Remote repository (p2exercises)

git remote add p2exercises ...
git remote add p2exercises ...

Remote repository (p2exercises)

Person X
do not commit after the deadline; it leads to merge conflicts
Basic git
A “commit” is “a set of changes” to a “set of files”
Most commits modify (or merge) earlier commits
A graph of commits may belong to a \textit{branch}
master
is the main branch
“HEAD “is the current branch
Create a git repo

```
mkdir repo
cd repo
git init
```
Tell git to “stage” changes

git add ...

HEAD
→ master
→ C0
Commit your changes

git commit ...

HEAD
master
C1
C0
Collaborating
John

Local repo

Public repo

master

C1

C0

Jane

Local repo
John

Local repo

- master
  - C2
    - C1
      - C0

- master
  - C1
    - C0

- master
  - C1
    - C0

Jane

Public repo

- master
  - C1
    - C0

- master
  - C1
    - C0

- master
  - C1
    - C0

Local repo

- master
  - C3
    - C1
      - C0

- master
  - C1
    - C0

- master
  - C1
    - C0

```
git add ...
git commit ...
```
(nothing new to pull)
John

Local repo

git push

master

C2

C1

C0

Public repo

master

C2

C1

C0

Jane

Local repo

master

C3

C1

C0
John

Local repo

- master
  - C2
    - C1
      - C0

Public repo

- master
  - C2
    - C1
      - C0

Jane

Local repo

- master
  - C3
    - C2
      - C1
      - C0

git pull
John

Local repo

- master
  - C2
  - C1
  - C0

Public repo

- master
  - C2
  - C1
  - C0

Jane

Local repo

- master
  - C4
    - C3
    - C2
  - C1
  - C0

NB: `git pull` = fetch + merge
John

Local repo

Public repo

Jane

Local repo

```
C0
  ↓
C1
  ↓
C2
  ↓
master
```

```
C0
  ↓
C1
  ↓
C2
  ↓
C3
  ↓
C4
  ↓
master
```

```
C0
  ↓
C1
  ↓
C2
  ↓
C3
  ↓
C4
  ↓
master
```

"git push"
John

Local repo

`git pull`

master

- C4
- C3
- C2
- C1
- C0

Public repo

master

- C4
- C3
- C2
- C1
- C0

Jane

Local repo

master

- C4
- C3
- C2
- C1
- C0
to be continued
Resources

http://git-scm.com/


https://github.com/

http://www.slideshare.net/chacon/getting-git

http://oreilly.com/
Attribution-ShareAlike 3.0

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:

**Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

http://creativecommons.org/licenses/by-sa/3.0/