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?

Bob

Carol
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
Bob

Repository

Version 1

Version 2

Carol
Controlled evolution

Can still lead to disaster!
git
git

Tracks the history of a collection of files
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 centralized version control system?
Bob → checkout → Central repository → Carol
Bob

commit

Central repository

update

Carol
Bob -> Central repository -> Carol
commit
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

Bob's local repository

Remote repository (groupXY)

Clone

Carol's local repository

Clone
Bob commits to his local repository. He then pushes his changes to the remote repository (groupXY). Carol has a local repository that is also connected to the remote repository.
Bob commits to the local repository. The local repository has a remote repository (groupXY) that allows push and pull operations.
Bob commits to his local repository. The changes are then pushed to the remote repository (groupXY), and Carol can pull these changes into her local repository.
Bob commits to a local repository, which is then pushed to a remote repository. Carol also commits to a local repository, with their changes eventually pushed to the same remote repository.
you must pull before every push
Remote repository (p2exercises)
git remote add p2exercises ...
git remote add p2exercises ...
git pull p2exercises master
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 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 ...
```
Commit your changes

git commit ...
Collaborating
John

Local repo

git clone ...

master

C1

C0

Public repo

master

C1

C0

Jane

Local repo

git clone ...

master

C1

C0
John

Local repo

- master
  - C2
  - C1
  - C0

Jane

Public repo

- master
  - C1
  - C0

Local repo

- master
  - C3
  - C1
  - C0

git add ...
git commit ...
git add ...
git commit ...
John

Local repo

Public repo

Jane

Local repo

---

```
git pull
```

(nothing new to pull)
John's Local repo

```
master
 C2
 C1
 C0
```

Jane's Local repo

```
master
 C3
 C1
 C0
```

Public repo

```
master
 C2
 C1
 C0
```

John commits C2, C1, and C0 to his local repository.

John pushes these commits to the public repository using `git push master`. The public repository now reflects the changes made by John.

Jane receives the updated public repository and can pull the changes to her local repository using `git pull`. This updates her local repository with the latest changes from the public repository.
NB: `git pull` = fetch + merge
John

Local repo

- master
  - C2
  - C1
  - C0

Public repo

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

Jane

Local repo

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

**git push**
John

Local repo

Local repo

Public repo

Local repo

git pull
In case of fire

1. git commit
2. git push
3. leave building
to be continued
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:

**BY:** 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/