Versioning solutions

Git Basics

Thomas Domingues

Summary

Introduction

- What is versioning? Why versioning?
- Which versioning solution?
- Git story (not a command)
- Useful git tools

Git basics

- Init / Clone / Remote
- Status / Add / Reset
- Commit / Log
- Push / Pull

Tutorials

What is versioning?

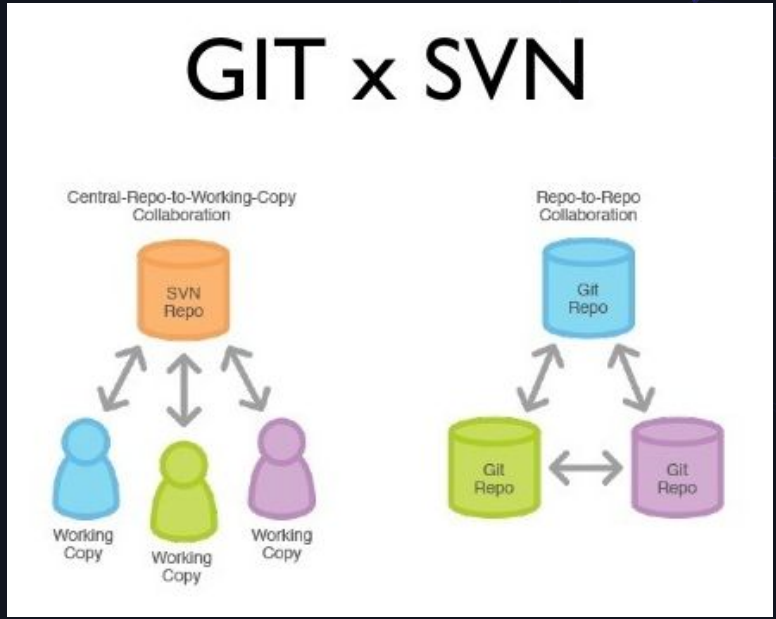
Version control, also known as source control, is the practice of tracking and managing changes to application code

Why versioning?

- Save project changes / history
- Work simultaneously with other developers from different computers (and Macs)
- Switch easily between multiple developments
- Being able to revert project to a stable state

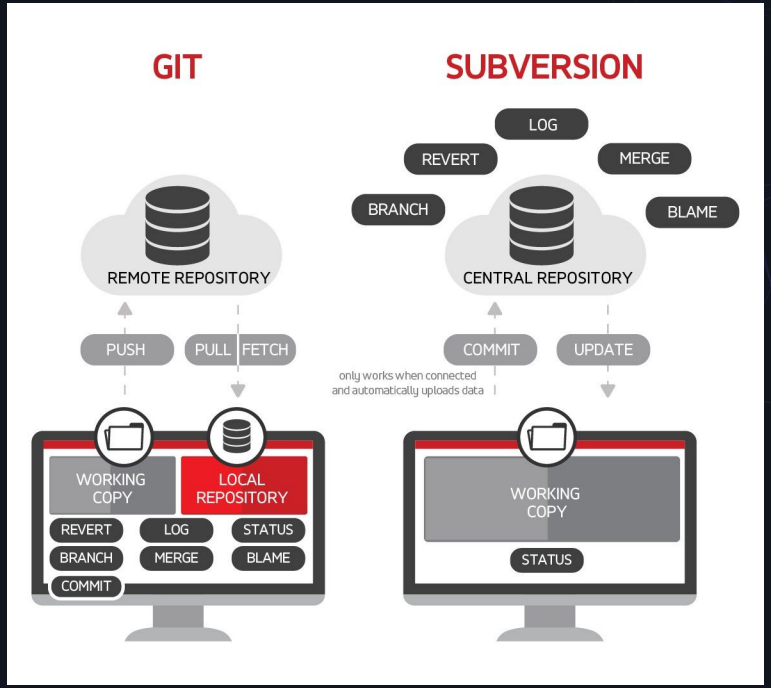
Which versioning solution?

- Two main solutions: Git and SVN
- Main difference: SVN behave as a simple communication between one server and multiple clients, in contrary of Git where all machines are servers and clients



Which versioning solution?

Main strength of Git: Big part of your actions doesn't depend on the server



Git story

- This is a picture of an happy Linus Torvalds, creator of Linux kernel.
- Linus is managing the Linux project by receiving community patches by emails, because he doesn’t like other versioning solutions (CVS, SVN, ...). But everyone is tired of this work method.
- In 2002, Linus managed to get a free licence of Bitkeeper, a versioning tool, for the Linux project

Git story

- This is a picture of unhappy Linux kernel contributors, open source evangelists who doesn’t want to use a proprietary solution.

Git story

- This is a picture of unhappy Linux kernel contributors, open source evangelists who doesn’t want to use a proprietary solution.
- This is a picture of the Linus answer

Git story

- In 2005, Andrew Tridgell, a main contributor of Linux kernel, mostly known for being the creator of the Samba protocol, want to continue contributing to Linux project without accepting the Bitkeeper licence. So he reverse-engineers Bitkeeper client to create his own open-source one
- In response, Bitkeeper revoke the free licence for Linux kernel development

Git story

End of the story: Linus take a 2 weeks vacation, come back with the Git project, open-source it and everyone is happy

Useful git tools

- GitKraken, a git GUI (obviously forbidden for this week)
- (Windows users) Git Bash, a better terminal for Windows bundled in Git For Windows

Git basics - Init

Init a Git Project

```
user@/git-project$ git init
Initialized empty Git repository in /git-project/.git/
user@/git-project$ ls -la
total 12
drwxr-xr-x 3 user user 4096 Dec 17 14:35 .
drwxr-xr-x 7 user user 4096 Dec 17 14:35 ..
drwxr-xr-x 7 user user 4096 Dec 17 14:35 .git
```

https://git-scm.com/docs/git-init

Git basics - Clone

- Clone a distant project: `git clone [path] [local folder]`
- Examples :
  - `git clone ../another/path/in/my/computer mynewfolder`
  - `git clone /another/path/in/my/computer mynewfolder`
  - `git clone https://my-git.server/user/repo.git mynewfolder`
  - ⭐ `git clone git@my-git.server:user/repo.git mynewfolder`

Git basics - Remote

Manage the set of repositories ("remotes") whose branches you track.

```
user@/git-project$ git remote -v
origin ../another/path/in/my/computer (fetch)
origin ../another/path/in/my/computer (push)
user@/git-project$ git remote add upstream git@my-git.server:user/repo.git
user@/git-project$ git remote -v
origin ../another/path/in/my/computer (fetch)
origin ../another/path/in/my/computer (push)
upstream   git@my-git.server:user/repo.git (fetch)
upstream   git@my-git.server:user/repo.git (push)
```

Git basics - Remote

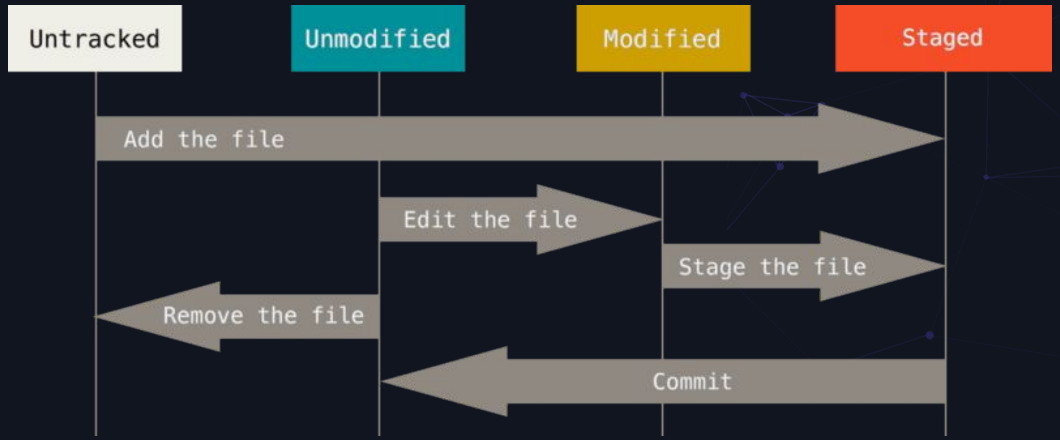
Other useful `git remote` commands:

- `rename`
- `remove`

https://git-scm.com/docs/git-remote

Git basics - Status

Manage file status:



From https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Git basics - Status

```
user@/git-project$ git status
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
user@/git-project$ touch README.md
user@/git-project$ git status
On branch master
Your branch is up to date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)
      README.md
nothing added to commit but untracked files present (use "git add" to track)
```

https://git-scm.com/docs/git-status

Git basics - Add / Reset

```
user@/git-project$ git add README.md
user@/git-project$ git status
[...]
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
      new file:   README.md
user@/git-project$ git reset HEAD README.md
user@/git-project$ git status
[...]
Untracked files:
  (use "git add <file>..." to include in what will be committed)
      README.md
[...]
```

https://git-scm.com/docs/git-add / https://git-scm.com/docs/git-reset

Git basics - Commit

- Initial configuration:
  - `git config --global user.email "tdomingues@myges.fr"`
  - `git config --global user.name "Thomas Domingues"`

Git basics - Commit

Record changes to the repository

```
user@/git-project$ git status
[...]
      new file:   README.md
user@/git-project$ git commit -m “Init project”
[master ef11547] Init project
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```

https://git-scm.com/docs/git-commit

Git basics - Commit

- What’s a good commit message? No fixed standards.
- https://greg0ire.fr/git-gud/ proposition:
  - First line: Main topic, up to 50 chars
  - Second line: empty <= mandatory
  - Third line and more: long description of changes
- Gitmoji proposition: Add emojis to your commit messages 🤩

Git basics - Log

Show commit logs

```
user@/git-project$ git log
commit ef115470ebdb1d0c97820ab0f5af6f3040470798 (HEAD -> master)
Author: Thomas Domingues <tdomingues@myges.fr>
Date:   Thu Dec 17 14:35:35 2020 +0100
    Init project
```

https://git-scm.com/docs/git-log

Git basics - Push

Update remote refs

```
user@/git-project$ git push upstream master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 873 bytes | 873.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To my-git.server:user/repo.git
   ef11547..0470798  master -> master
```

https://git-scm.com/docs/git-push

Git basics - Pull

Fetch from and integrate with another repository

```
user@/git-project$ git pull upstream master
Already up to date.
```

https://git-scm.com/docs/git-pull

Tutorials

Your turn! Thanks to the previous slides, make your first commit with only command line:

- Install and configure Git on your machine
- Init a git project
- Create a README.md file
- Make a commit
- Log the commit
- Send me in private the answer of the log