



GIT

OSSS - 2

Mihai Maruseac mihai@rosedu.org

June 18, 2013

ROSEdu

Why?

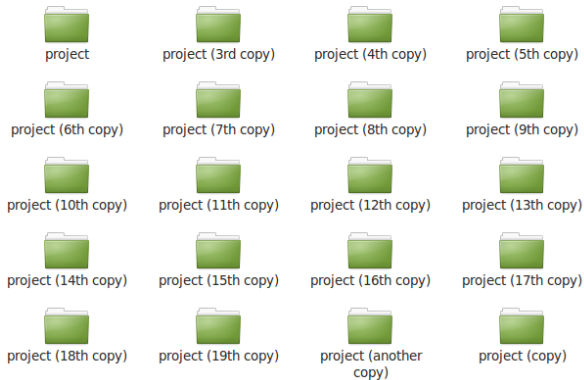


project



project (copy)

Why?



Why?

- change history

Why?

- change history
- ability to go back

Why?

- change history
- ability to go back
- query about certain changes

Why?

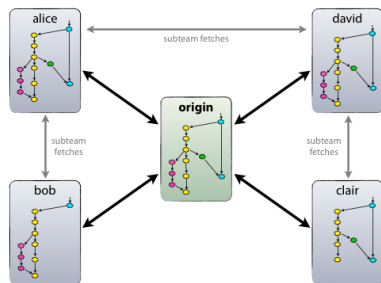
- change history
- ability to go back
- query about certain changes
- non-linear development

Versioning Solutions



Why Git?

- cool
- everybody uses it
- distributed
- no Internet connectivity needed
- well-documented
- GitHub, Gitorious, BitBucket, ...
- Linux, Mac, Windows, ...
- continuous development
- super-cool facilities for powerusers
- more power by integration with third-party tools



Git Design

- performance
- nonlinear development
- integrity

GitHub

- <https://github.com/>
- <http://osrc.dfm.io/>
- <http://song-of-github.herokuapp.com/>

Initial Config :: SSH key

- ssh-keygen
- In `~/ .ssh`:
 - `id_rsa`
 - `id_rsa.pub`

Initial Config :: SSH key on GitHub

The screenshot shows the GitHub user profile page for 'mihaimaruseac'. The top navigation bar includes the GitHub logo, a search bar, and links for 'Explore', 'Gist', 'Blog', and 'Help'. The user's name and profile picture are visible in the top right. A red arrow labeled '1' points to the 'Settings' icon (a gear) in the top right corner.

On the left side, there is a sidebar menu with the following items: Profile, Account Settings, Emails, Notification Center, Billing, Payment History, **SSH Keys 2** (highlighted), Security History, Applications, Repositories, and Organizations. At the bottom of the sidebar, it shows 'PRIVATE REPOS 0 OF 0' with a yellow progress bar.

The main content area is titled 'SSH Keys' and contains a table of existing keys:


SSH Key	Action
Keldon (31:91:cf:af:c7:7a:45:8d:15:89:26:15:c9:bb:d3:f1)	Delete
techblog (f1:6a:78:1c:74:96:c6:17:cb:f3:5b:e9:80:f9:7f:aa)	Delete
mithlond (4d:a2:a0:1d:15:35:5f:de:03:ea:67:cb:96:11:f8:73)	Delete

There is a red '3' next to the 'Add SSH key' button. Below the table is a form titled 'Add an SSH Key' with the following fields:

- Title**: A text input field containing the text 'title'.
- Key**: A text area containing the text 'id_rsa.pub'.

GitHub :: online editing


branch: master | git-demo / participants.md

 **mihaimaruseac** just now Update participants.md

1 contributor


file | 2 lines (1 sloc) | 0.021 kb

Edit Raw Blame History

Code Preview 

Spaces 2 No wrap


```
1 Mihai Maruseac, 2013
```

 Commit summary:

Update participants.md

Extended description: (optional)

Added myself to participants.md.

 **mihaimaruseac**
mihaimaruseac@gmail.com

Cancel Commit Changes

Initial Config

- `git config --global user.name "name surname"`
- `git config --global user.email "name@dom.com"`
- `git config --global color.ui auto`
- `git config --global color.pager true`
- `git config --global core.editor editor`

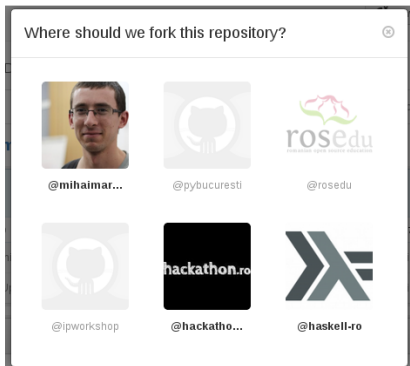
Initial Config

- `git config --global user.name "name surname"`
- `git config --global user.email "name@dom.com"`
- `git config --global color.ui auto`
- `git config --global color.pager true`
- `git config --global core.editor editor`

- `cat ~/.gitconfig`

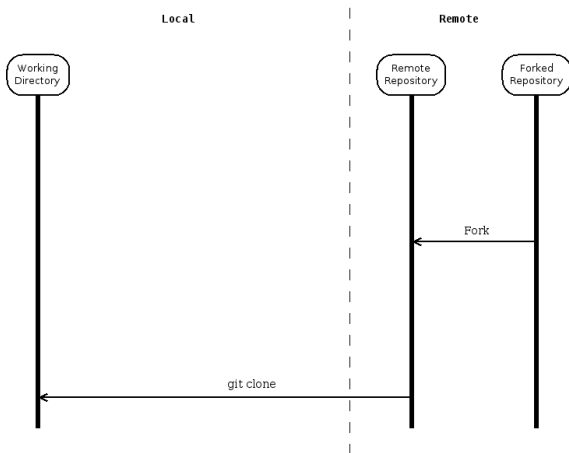
GitHub :: Forking

PUBLIC  **rosedu / git-demo**  Unwatch  Star 0  Fork 0

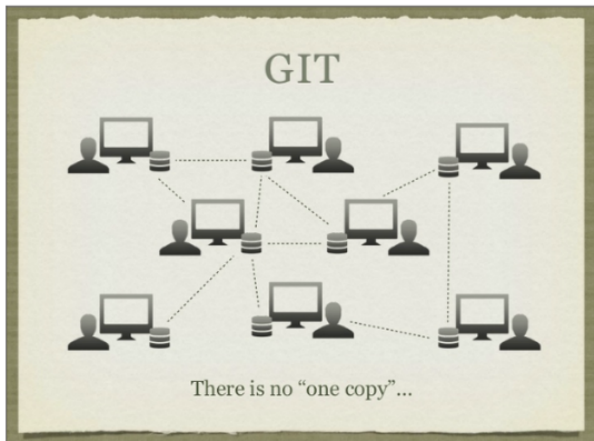


GitHub :: Cloning

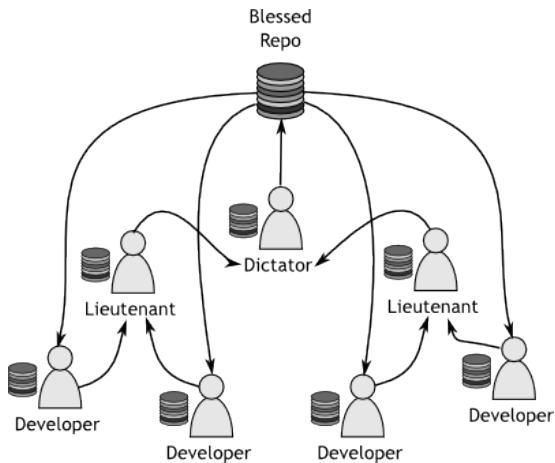
```
git clone URL
```



All repositories are created equal...

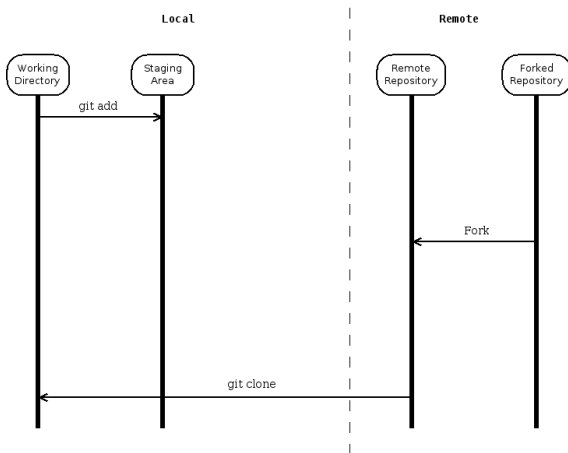


... but some are more equal than others



First Commit :: Making and Adding Changes

`git add`



Binary Files

- binary diffs too big → rewrite
- big size cost
- binary files are generated
 - Rule of thumb: no generated file must be included
 - object files, executables, libraries, PDF from \LaTeX sources, images from Dia or Inkscape, ...
 - must be included if not generated

Ignore

- local `.git/info/exclude`
- repository `.gitignore`: root or local to subdirectory
- system wide `excludesfile`

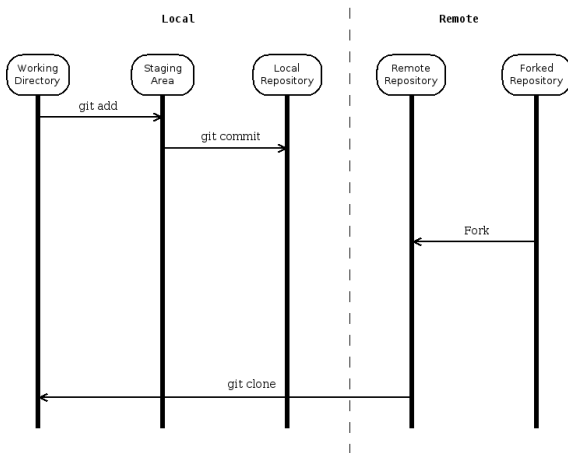
Ignore

- local `.git/info/exclude`
- repository `.gitignore`: root or local to subdirectory
- system wide `excludesfile`

- `file.pdf`
- `/file.pdf`
- `!file.pdf`
- `/test/file.pdf`
- `*.pdf`
- `/*.pdf`
- `*.swp, *~, tags, TAGS, cscope.out`

First Commit :: Committing Changes

```
git commit
```



Commits

- small and frequent
- *Do one thing, do one thing well!*
- `git add --patch` or `git add -i`

Commit Messages

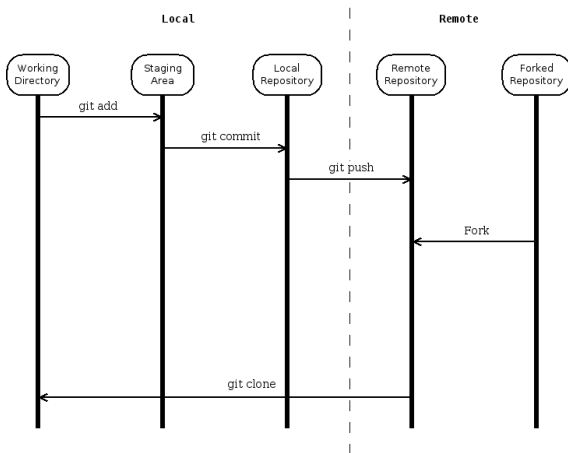
- descriptive
- proper phrases (Capital letter in front, ending with full-stop.)
- 50 characters on the first line
- other paragraphs afterwards describing the change
- <http://tbagery.com/2008/04/19/a-note-about-git-commit-messages.html>
- <http://whatthecommit.com>

Crap. Tonight is raid night and I am already late.

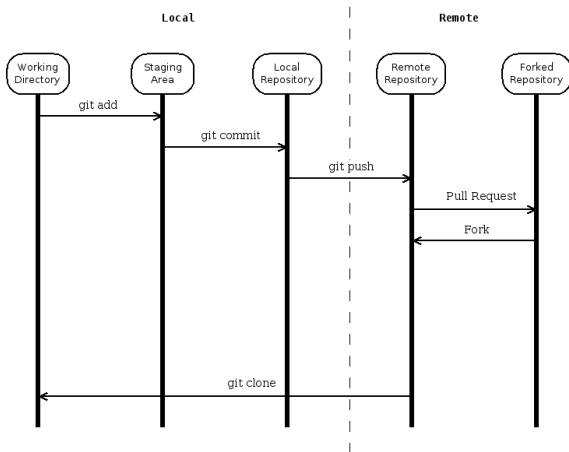
[permalink]

First Push

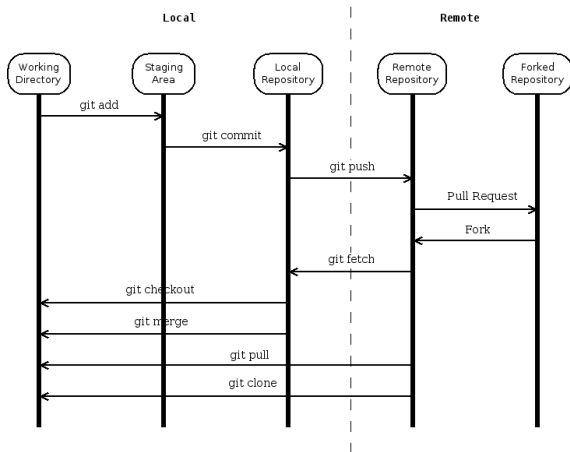
`git push`



GitHub :: Pull Requests



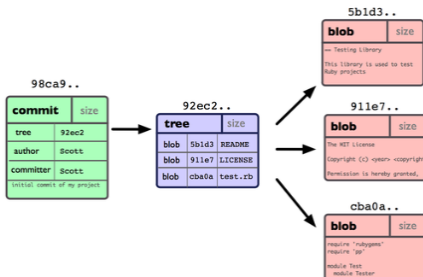
The Entire Picture



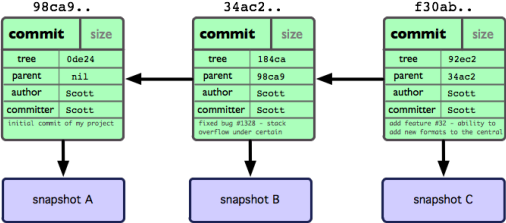
Viewing the History

- `git log`
- `gitk`, `gitg`, `tig`, ...

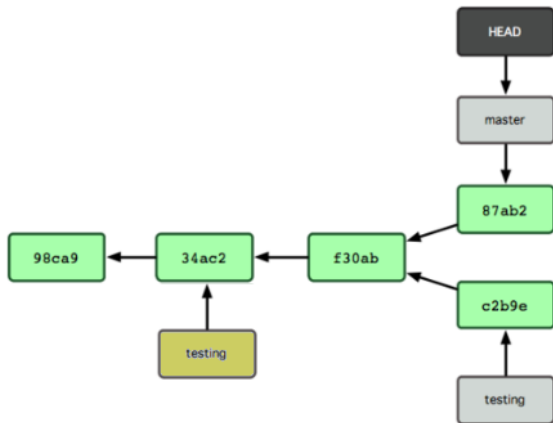
Commit = snapshot, not delta



Each commit knows his parent



Tags & Branches



Viewing the History

- `git show id`

Viewing the History

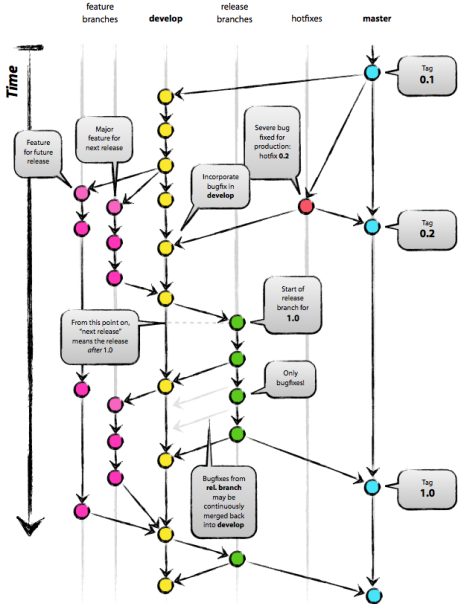
- `git show id`
- HEAD
- tag
- HEAD^ HEAD^^, ...
- HEAD~10
- hash
- partial hash

Branching

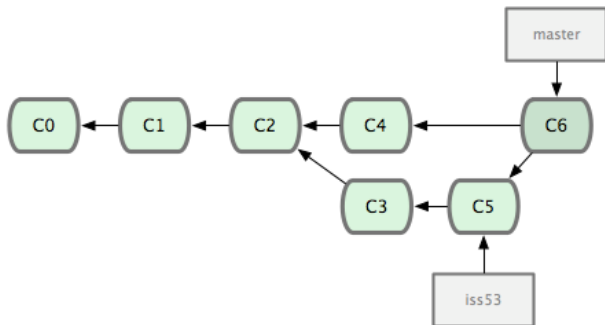
- collaborative development
- nonlinear development
- fast to create, easy to use
- the most powerful feature of Git

- `git branch`
- `git checkout`

Branching model



Merge

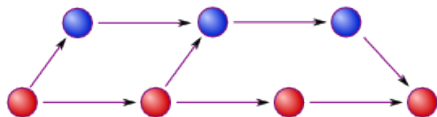


Branches and Clones

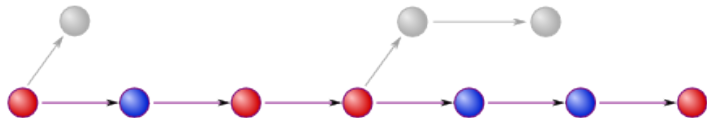
- each clone can be seen as a new branch
- `git pull` makes a merge commit
 - solve conflicts
 - simultaneous changes
- extra commit not always needed


```
git pull --rebase
```

Loopy history (git pull):



Nice history (git pull --rebase):



Conflicts

- what happens when editing same file?

Conflicts

- what happens when editing same file?
- Git manages conflicts as far as possible

Other Cool Git Features

- `git blame`
- `git bisect`
- `git cherry-pick`
- `git stash`
- `git send-email`, `git format-patch`
- `git filter-branch`
- Git hooks

Git is...



Local Workflow

- ① create repository: `git init .`
- ② create files/directories: `editors, mkdir, cp`
- ③ add **content**: `git add`
- ④ commit: `git commit -m "Message"`
- ⑤
- ⑥
- ⑦
- ⑧ goto 2

Contributor Workflow

- ① clone repository: `git clone URL`
- ② create files/directories: `editors, mkdir, cp`
- ③ add **content**: `git add`
- ④ commit: `git commit -m "Message"`
- ⑤ for more local work goto 2
- ⑥ refresh repository: `git pull --rebase`
- ⑦ push changes: `git push origin master`
- ⑧ goto 2

Keywords

- repository
- clone
- commit
- tree
- tag
- HEAD
- remote
- push/pull
- branch
- merge
- patch
- staging area
- fetch/checkout

Links

- Git Immersion
- git ready
- Git Magic
- What is git?
- Git: The Lean, Mean, Distributed Machine
- Tech Talks Git (Mircea Bardac)
- A branching model
- Workflow GitHub
- Git posts on ROSEdu Techblog