Scaleway

The cloud that makes sense

git 101

Scaleway

Overview

git by itself

git in team

Scaleway

git by itself

Scaleway

What problems does git solve?

- Distributed Social Coding
- Snapshots of your code (Version control)
- Try out new idea easily (Branches)

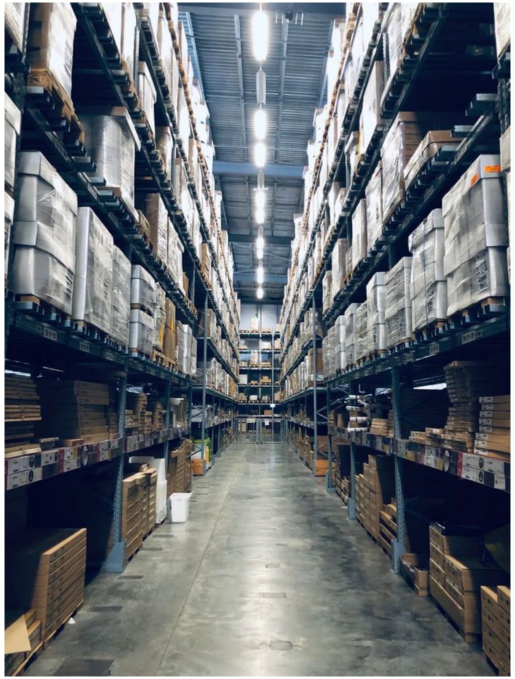
http://tom.preston-werner.com/2009/05/19/the-git-parable.html

What problems does git help to solve?

- Continuous integration
- Agile workflow
- Open Source in general

Scaleway

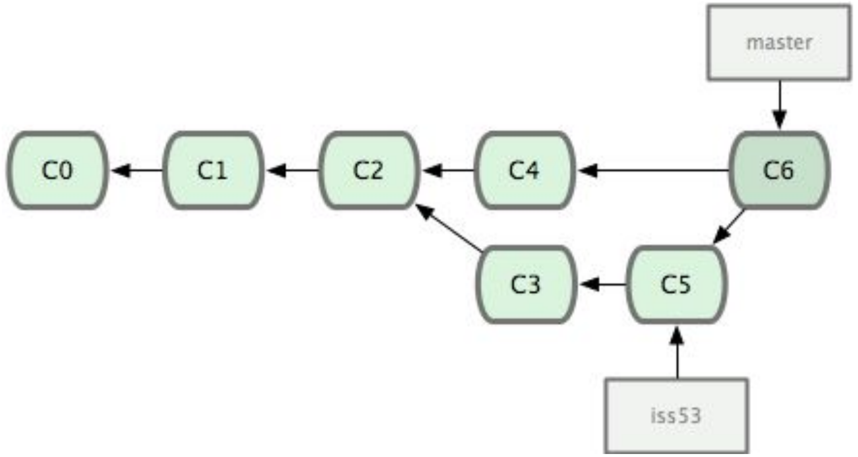
How does git works on your computer?



- git init / git clone
- git persists its state in .git/
- all git commands interacts with those files
- If this folder is gone, so is your history

Scaleway

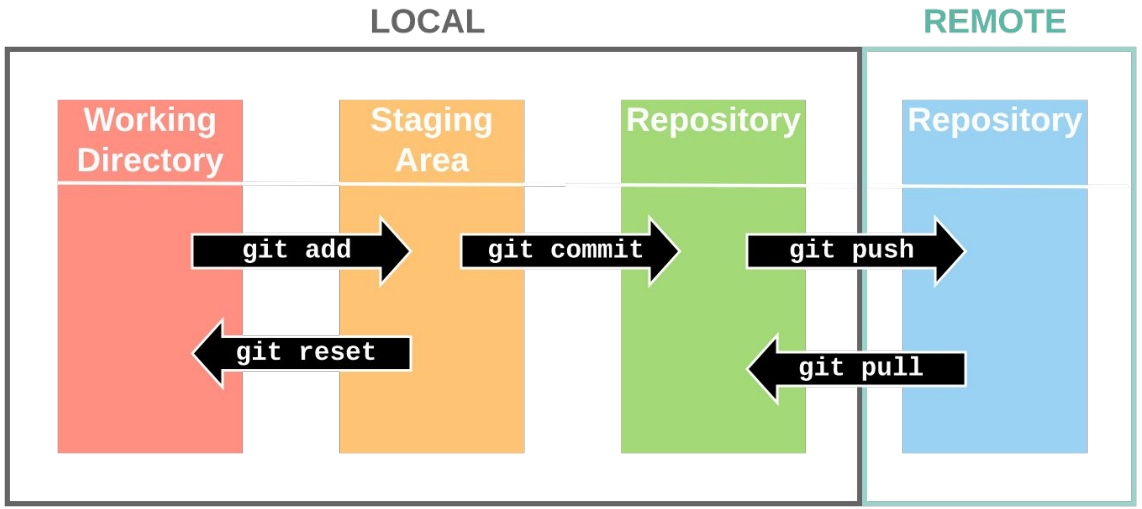
Visualizing git and branching



- http://git-school.github.io/visualizing-git/
- https://learngitbranching.js.org

Scaleway

How to communicate with git?



http://ndpsoftware.com/git-cheatsheet.html

General best practices on git

- Know your commits. It is your work, your craft.
- Don't commit everything you are modifying
  - add -p is your friend
- Write meaningful commit messages
  - Easier to understand during peer reviews

Scaleway

I've messed up and I don't know what to do

- Visualize
- Express with the right words what you want to do

Scaleway

Tips and tricks

- https://github.com/git-tips/tips
- git help
- git help --all
- git help --guide
- git help glossary
- https://github.com/k88hudson/git-flight-rules

git with your team

Scaleway

Workflow

Scaleway

Many choices
(git does not care)

Scaleway

github style

- main branch can be deployed. Always!
- All individual developers fork your project
- A feature per branch
- People open merge request if they want to have something merged
- feature is merged when reviewed, tested, accepted
- Other rebase if needed. That’s your job to keep track with main branch.

Scaleway

Who makes the call to merge something?

- 100 % dependent of the project
- Usually two reviewers to get something merged
- Depends on how many people work on your project

How can I stay up to date with upstream changes?

- merge or rebase strategy
    - http://gitforteams.com/resources/merge-rebase.html
- My favorite is rebase (easier to read on the commit DAG)

Scaleway

Eyes on the road

- git blame
- git log
- git reflog to see everything that happened on your local repository

Scaleway

Reviews

- Automated gatekeeper
- Peer review

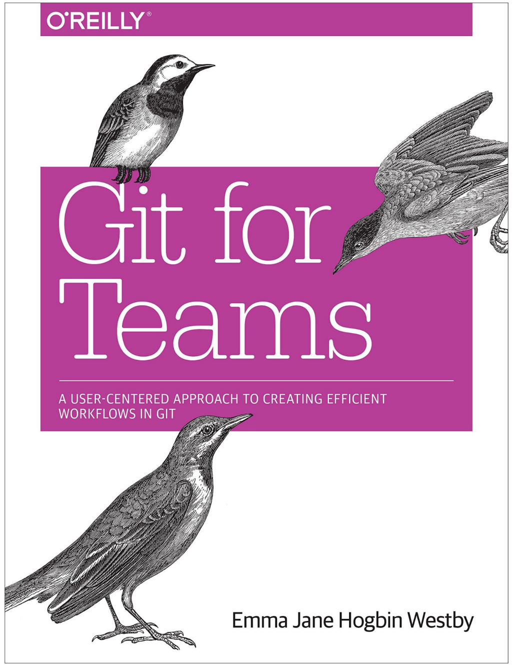
https://speakerdeck.com/nnja/code-review-skills-for-pythonistas-djangocon-2018

Scaleway

General best practices on distributed development

- One change per commits (no little changes on the side)
- Small commits that can be reviewed easily
- Add tests to your code that are launched automatically by CI/CD
- Published history should not be altered.
  - Don’t do that
  - Seriously DO NOT DO THAT

Reference:


O'REILLY
Git for Teams
A USER-CENTERED APPROACH TO CREATING EFFICIENT WORKFLOWS IN GIT
Emma Jane Hogbin Westby

- git for teams

Scaleway

Any questions?

rleone@scaleway.com

Scaleway