



Marc-Philipp Knechtle

Going public

Tips for publishing your own code

Table of contents

- 1 Hide your secrets!
- 2 Creating a healthy community
- 3 Consistency achieved with continuous integration
- 4 Documentation
- 5 Promote your code

Sensitive data in git

- Problem: (Accidental) commits of secrets
- e.g. passwords, application secret keys, OAuth secret keys, SSH keys
- The file is still accessible in history, even if the original file was deleted
- Security risk, even in private repositories

Removing sensitive data

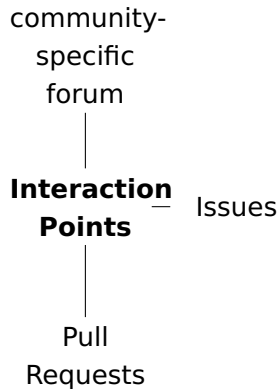
- git filter-repo
- BFG Repo-Cleaner

Encrypt sensitive data

Methods for encrypting and storing sensitive data:

- Why shouldn't I store the secrets separate from the repository?
- Methods for encrypting and storing sensitive data:
 - ▶ git-secret CLI
 - ▶ github encrypted secrets
 - ▶ technology specific encryption tool (e.g. ansible-vault)

Points of interaction with the community

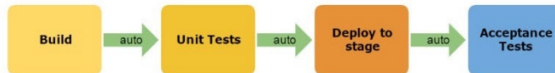


Code of Conduct

- Content: set of rules, with norms, responsibilities and proper parties
- defines the actions of a single person in a organisation
- Goal: healthy, constructive community behaviour
- CODE_OF_CONDUCT file (in the git repository)
- e.g. Contributor Covenant ¹, Django Code of Conduct ²

Continuous Integration overview

Continuous Integration



Continuous Delivery



Continuous Deployment



Github Actions

- CI/CD platform, used for testing Pull Requests
- Reactions on other events such as an issue being created
- workflows can run on VMs or a self-hosted instance
- workflows consist of jobs, which is a shell script or an action
- actions are custom github actions applications (e.g. pulling a repository, setting up the toolchain)
- runners are the servers on which the process is performed

Documentation

This was included in my task description, but is also topic of the work of @Nico.
Should I include this in my presentation?

Ideas for promoting your code

■ asdf

References