Marc-Philipp Knechtle

Going public

Tips for publishing your own code

Table of contents

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

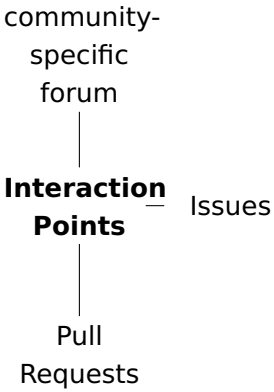
- git filter-repo
- BFG Repo-Cleaner

Encrypt sensitive data

Methods for encrypting and storing sensitive data:

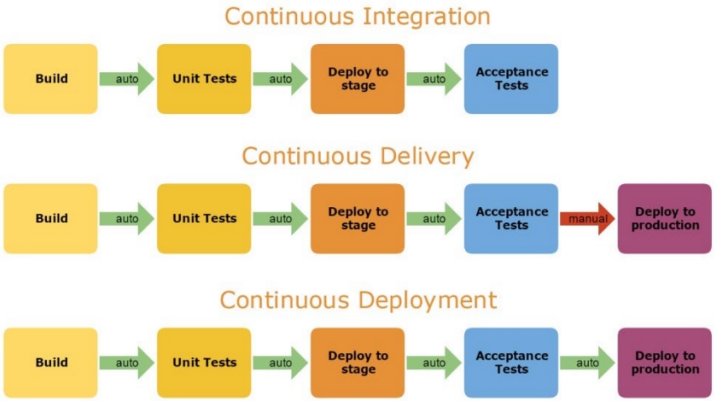
- Why shouldn’t i store the secrets separate from the repository?
- Methods for encrypting and storing sensitive data:
  - asdf
- github enrypted secrets
- git-secret
- technology specific encyption tool (e.g. ansible-vault)

Points of interaction with the community



Code of Conduct

- Goal: healthy, constructive community behaviour
- `CODE_OF_CONDUCT` file
- e.g. Contributor Covenant [1], Django Code of Conduct [2]


Continuous Integration
Build
auto
Unit Tests
auto
Deploy to stage
auto
Acceptance Tests
Continuous Delivery
Build
auto
Unit Tests
auto
Deploy to stage
auto
Acceptance Tests
manual
Deploy to production
Continuous Deployment
Build
auto
Unit Tests
auto
Deploy to stage
auto
Acceptance Tests
auto
Deploy to production

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