Python tooling for continuous deployment

PyParis Novembre 2018

Arthur Lutz - Logilab

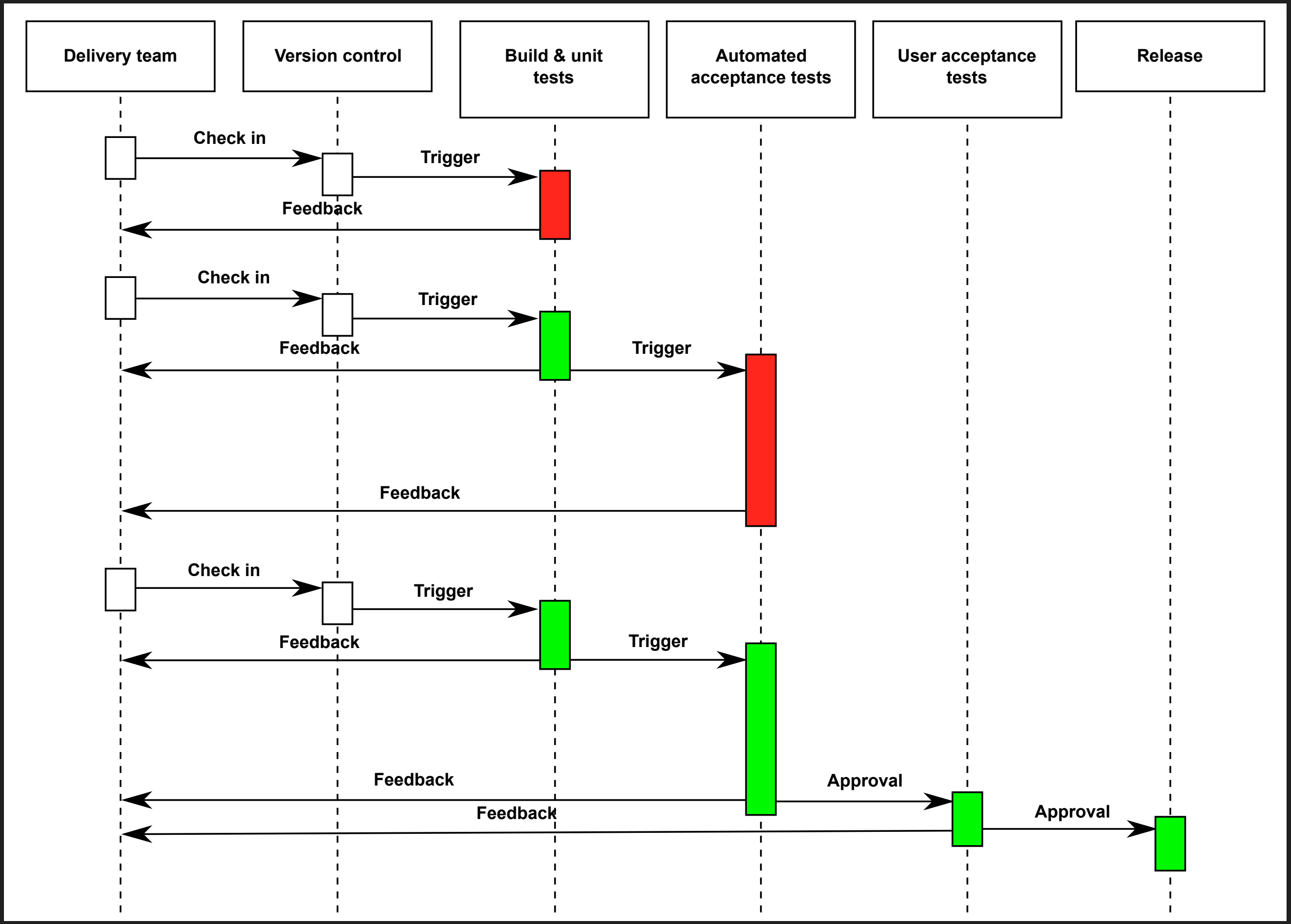
Introduction /me

- Arthur Lutz
- @arthurlutz
- @arthurlutz@social.logilab.org
- Logilab
- Python Nantes / Paris Salt Meetup

Introduction

How we benefited from a rich Python ecosystem to move from sprint-based delivery to continuous delivery.

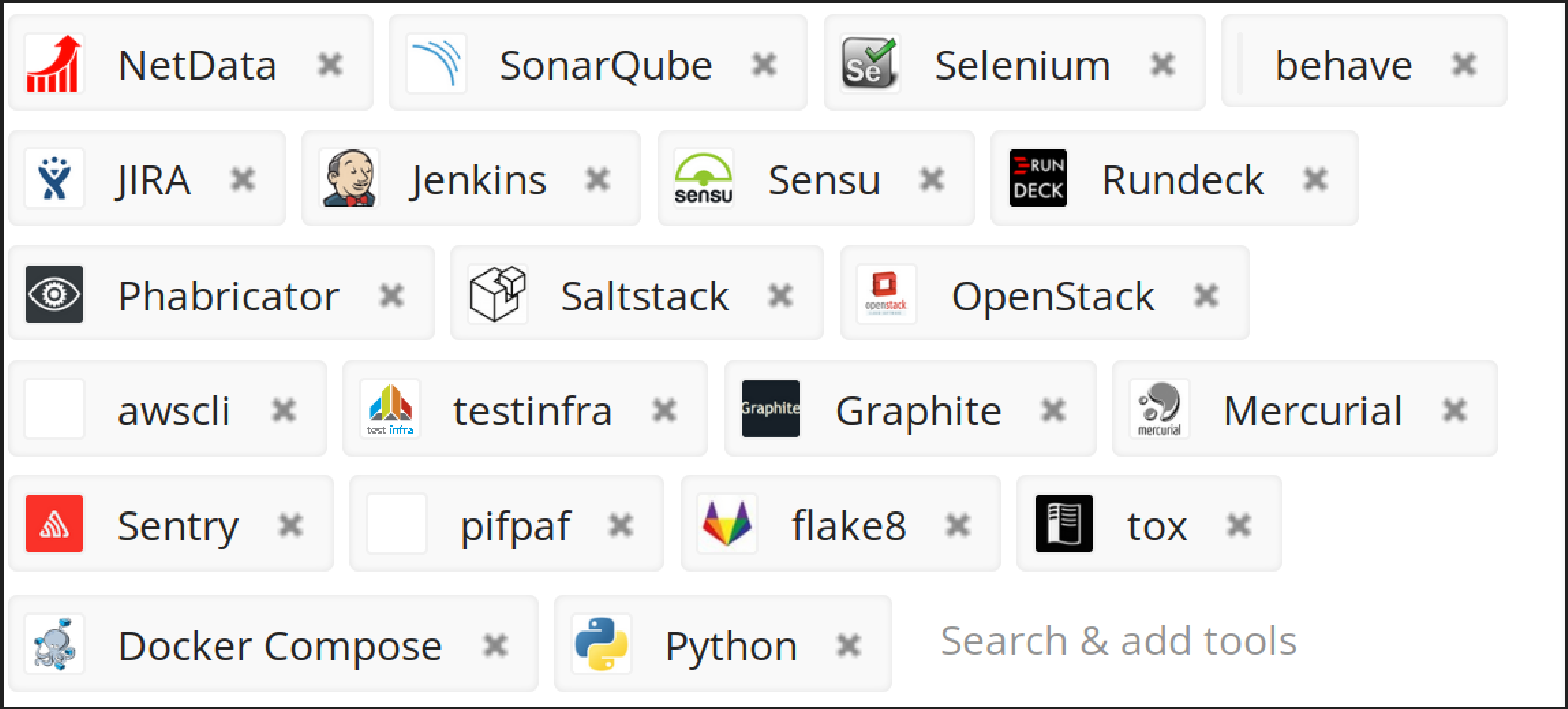
Continous delivery



Source: wikipedia

Python Stack

(ignore the order)



NetData ✖ SonarQube ✖ Selenium ✖ behave ✖

JIRA ✖ Jenkins ✖ Sensu ✖ Rundeck ✖

Phabricator ✖ Saltstack ✖ OpenStack ✖

awscli ✖ testinfra ✖ Graphite ✖ Mercurial ✖

Sentry ✖ pifpaf ✖ flake8 ✖ tox ✖

Docker Compose ✖ Python ✖ Search & add tools

View it on StackShare

Reproducible environment

- docker using volumes for dev parts (equivalent to `pip install -e`)
- docker-compose to set up postgres, redis, celery, etc. using `docker-compose.override.yml` and then `.env`
- same docker images for `acceptance testing`, `pre-production` and `production` with configuration via environment variables

Test often, break, test again

> *Tools used in dev, CI/test and supervision.*

- tox - reproducible test envs
- pytest - test runner
- flake8 / isort
- pifpaf - functionnal testing (launch daemons in pytest fixtures)

Ramping up CI

1. draft-based
2. parametric (manual)
3. public changeset (hg phases) + user-centric
4. component-centric
5. feature-centric

- jenkins (not python) with jenkins-job-builder
- phabricator + jenkins = Differential jenkins job and `JenkinsFile`

Acceptance test often, break, test again

- BDD with behave / robber
- selenium python bindings

Push, decrease quality, push again

Continuous Quality Control

- Jenkins test metrics and coverage
- SonarQube (not python) with SonarPython
- Peer review with mercurial + evolve (`amend`) + phabricator mercurial extension (`phabsend`)

sonarqube

Ship often, break, ship again

- rundeck (not python) & rundeckrun for shared deploy logs
- generate unique version information
- tag for validation the shipped Jira tickets
- **sentry & raven-python for continuous collection of errors**

Ship often, measure, ship again

- collect metrics with carbon and serve with graphite-api
- collecting custom metrics see cfmgmtcamp: Use Saltstack to deploy a full monitoring and supervision stack

Agile infrastructure

- saltstack infrastructure as code, deploy multiple environments
- salt-cloud portable cloud scaling
- openstack for in-house cloud hosting
- awscli used for example to restore production database snapshots in pre-production

SALTSTACK

Agile infrastructure - supervision and metrics

- sensu (not python) supervision using sensu-formula, ease of deploy of new checks
- netdata supervision and metrics with python.d.plugin
- testinfra to test out complex scenarios

Dashboards everywhere

- Jira python bindings to extract data from Jira
- requests to extract data from RunDeck
- *Badges everywhere!* https://shields.io (not python)
- grafana (not python) serving data from graphite

Project impacts

- shorter tickets, split them into tiny chunks
- unstable environments can be OK - work with the users
- feature flags to encourage early integration of code
- more peer review, collective appropriation
- no more semantic versionning

Future

- Merge request / *feature stack* centric generated environments
- Automation of data migration between environments
- Use of version control in sentry and regression identification
- OpenShift / OKD / Kubernetes type continuous deployment
- Dev environments could be less necessary

Logilab is hiring !

Conclusion / Questions

- Thanks for your attention
- Any questions ?
- Slides (full of links !) : html, pdf