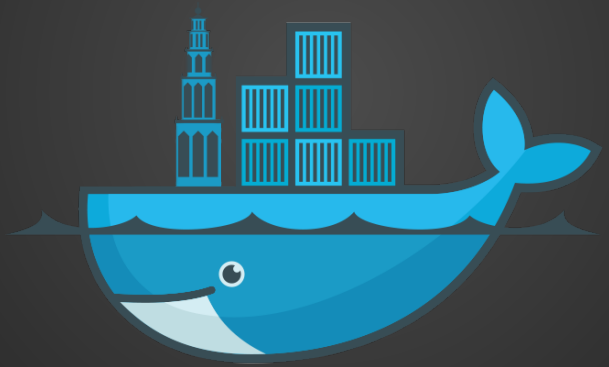
Continuous Integration and Continuous Delivery



Dirk Nederveen (Recras)

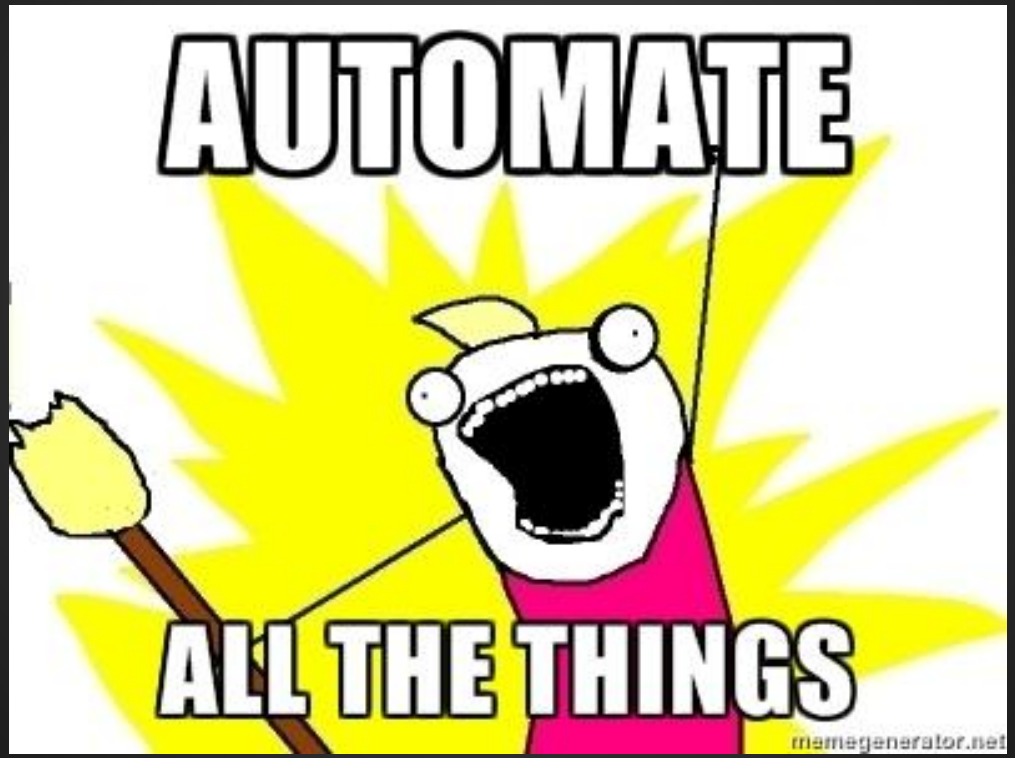
About Recras

- Web based Booking and CRM for recreational businesses
- Major software supplier for rope courses
- *Managing your business should take 2 hours a week*

About Recras - geeky bits

- 2 full time devs (Tijmen and me)
- ~3 releases each week
- Git flow: branch for each feature or fix
- *Prefer “It’s fixed” to “Working on it” as reply*

About Recras - geeky bits



What is Continuous Integration?

- Developers working on shared mainline
- Automated builds and automated testing
- Make it easy to get latest deliverables

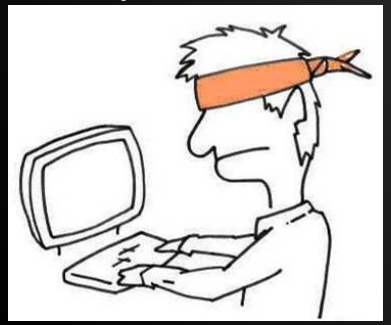
- Continuous Delivery: ensure mainline is always in a state that can be deployed

What is Continuous Integration?

Martin Fowler (2006):

When I've described this practice to people, I commonly find two reactions: "it can't work (here)" and "doing it won't make much difference". What people find out as they try it is that it's much easier than it sounds, and that it makes a huge difference to development. Thus the third common reaction is "yes we do that - how could you live without it?"

The CI continuum



CI/CD Grocery list

- Development process
  - git-flow, github-flow, git-process, unstable trunk, …
  - If merging is easy, keep main branch stable;
    if merging is difficult, keep main branch unstable

CI/CD Grocery list

- Development process
- Repeatable builds
  - Documentation is the first step!
  - Challenge: start a clean VM and install your project
    - `cat ~/.bash_history > INSTALL.sh`
  - Easy for you, easy for new hires

  - Useful tools: Vagrant, Ansible, Docker

CI/CD Grocery list

- Development process
- Repeatable builds
- Repeatable tests
  - Start *somewhere*, expand tests as you go
  - Make sure your peers know how to run tests

CI/CD Grocery list

- Development process
- Repeatable builds
- Repeatable tests
- Automated builds
  - A service that runs your tests
  - Add VCS hooks for automatic awesomesauce
  - Build bot tests all branches
  - Jenkins is a good option

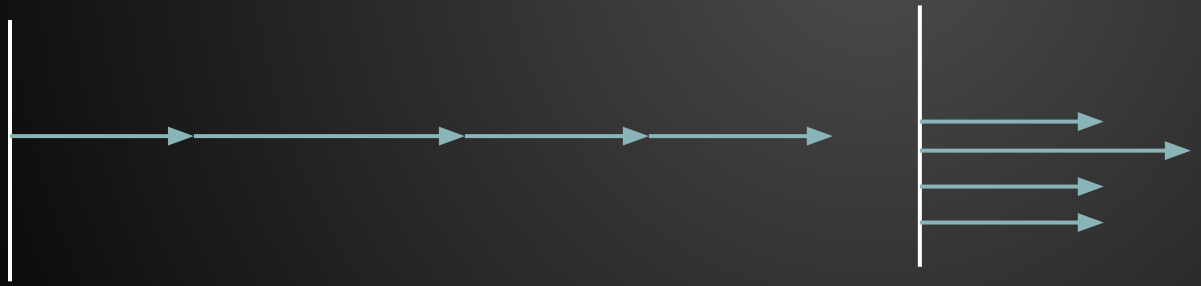
CI/CD Grocery list

- Development process
- Repeatable builds
- Repeatable tests
- Automated builds
- Make deployments a non-event

CI with Docker at Recras

Why containerize?

- Run several test suites at the same time
- cf.



vs

CI with Docker at Recras

- Single container image per test suite
  - Fetches specified version
  - Starts mysql, nginx, php5-fpm
  - Runs test suite

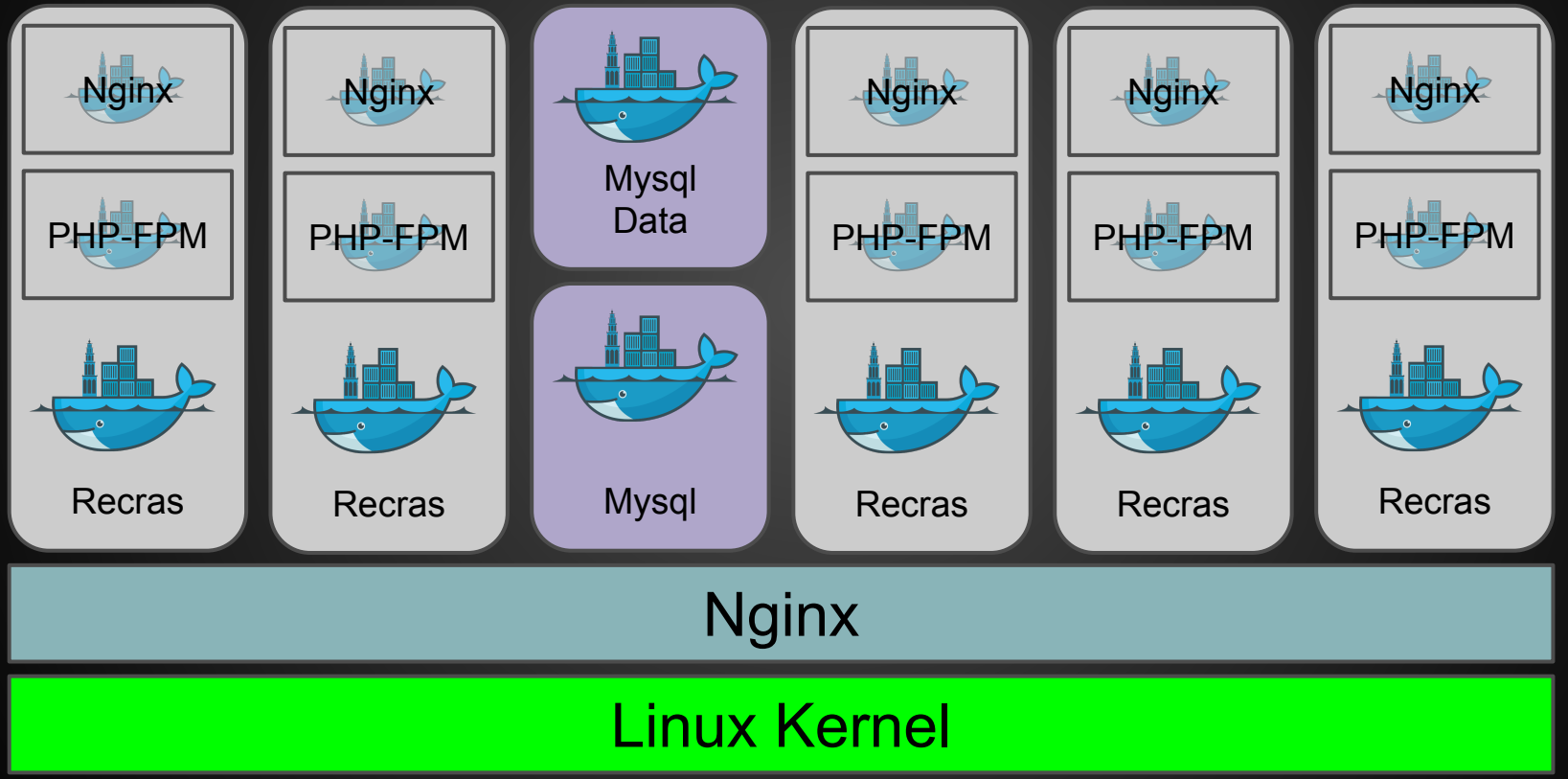
CI with Docker at Recras

Single container is not a great solution

Room for improvements:

- Container per service
- Use `fig` or Docker Compose
- Containerize test suites & use links

CI with Docker at Recras



Why Docker instead of ...?

Containers vs virtual machines

- Startup time
- Memory footprint

Docker vs other container tech

- Linux
- Ease of use (vs raw LXC)
- Buzzword compliant & hipster proof

Where we’re going

Containerize all things CI:

- Jenkins in Docker
- … creating container images per branch
- … running several test suites per commit

Where we’re going

Containerize all things deployment:

- Jenkins builds container image per tag (Docker inception)
- Deploy instances

That’s all, folks

Discussion and/or questions?

Email: dirk@recras.nl
Twitter: @nederdirk