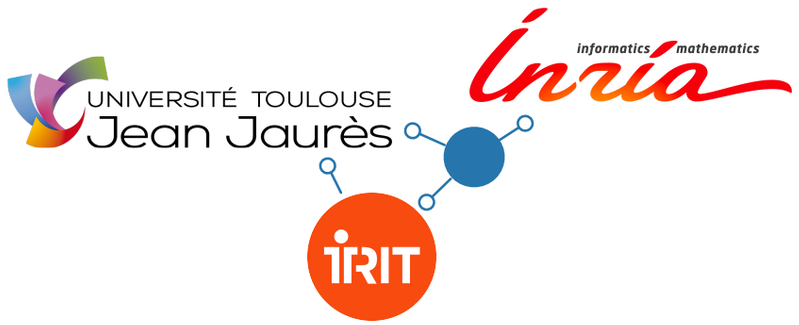
TEACHING DEVOPS

Toulouse, Rennes…

WORKSHOP DEVOPS'19

BENOIT COMBEMALE
PROFESSOR, UNIV. TOULOUSE & INRIA, FRANCE

HTTP://COMBEMALE.FR
BENOIT.COMBEMALE@IRIT.FR
@BCOMBEMALE



My personal experiences in teaching DevOps

- Course is integrated into a SE curriculum @ Master level
  - @Toulouse: focus on agility and collaboration
- Requirements:
  - Dev: skills in sw development, incl. design, architecture, programming, testing, etc.
  - Ops: skills in system administration, computer architecture, and virtualization

- Overall structure of the course:
  1. **A review of the literature** about the current practices at the major IT key players => ***assessment #1***
  2. **A project** with interleaved lectures => ***assessment #2***

Overview of the course

- **Introduction**
  - Some Facts on Modern Developments
  - State of the Practice (large-scale, polyglot, short term delivery…)
  - DevOps: current scope, concepts and principles

- **Get ready for DevOps!**
  - Execution platform (e.g., virtual machines, containerization and clouds),
  - Software architecture (microservice, stateless),
  - organizational concerns (gitflow and branching, continuous improvement…)

- **Test automation**, incl. flaky test, code and test coverage, mutation analysis, fuzzing.

- **Build Management, Configuration Management, Release Management**
  - Software Build (e.g., Maven)
  - Software Delivery (e.g., Docker, Docker Compose)
  - Software Deployment (e.g., Kubernetes)
  - Continuous Integration, Delivery and deployment (e.g., Jenkins)
  - Infrastructure as code

- **Measurement**: Logging, Tracing and Monitoring (e.g., LogStash, OpenTracing, Sonar, Kibana)

- **A/B and Canary Testing**

- **Resilience engineering** / testing (e.g., Chaos Engineering)

Opportunities

- Cross reference and link together various courses of the curriculum

- Emergence of free (and sometimes open) frameworks, (e.g., GitHub, travis, classroom…)

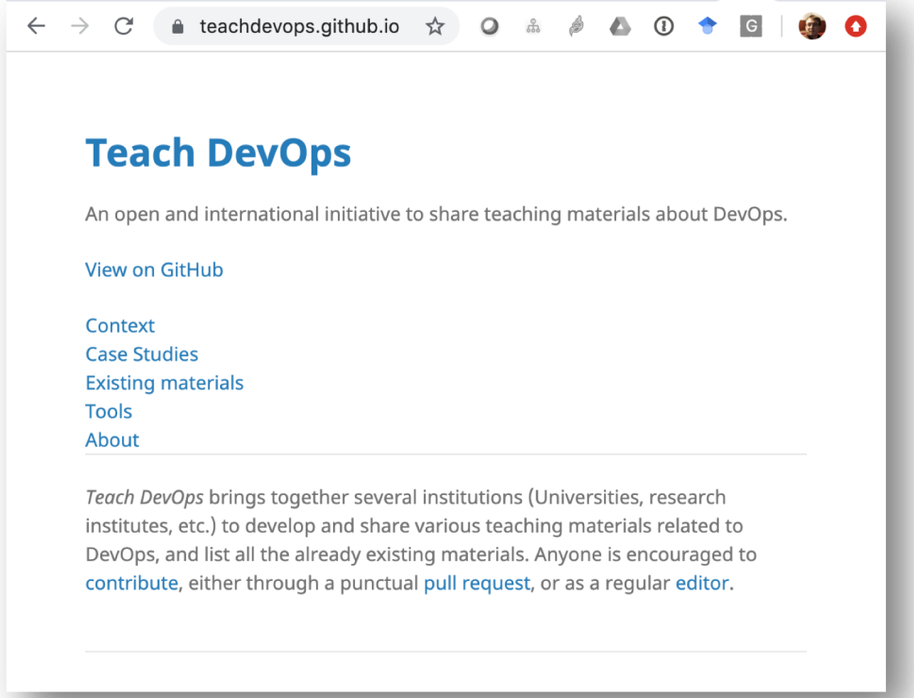
- *Easily fun for students! ☺*

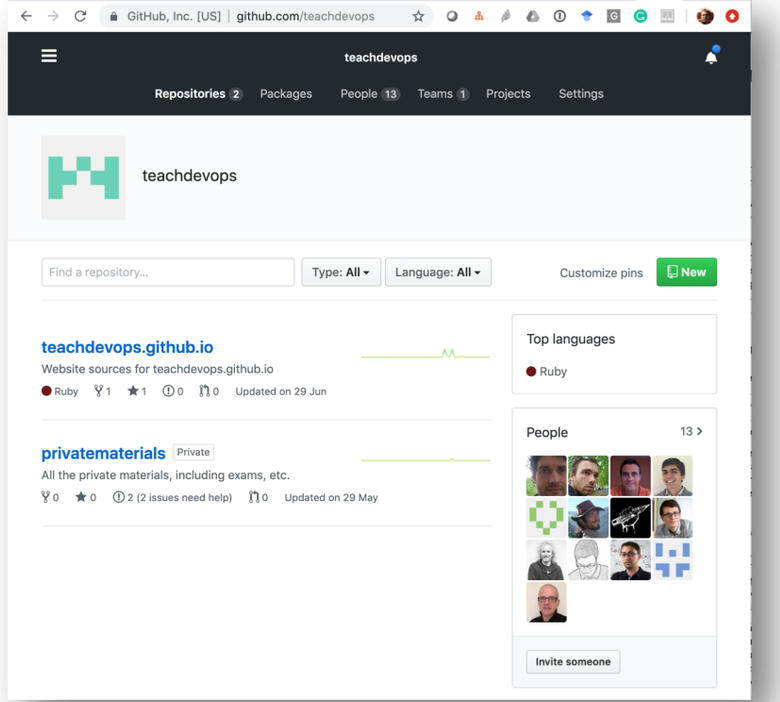
Difficulties

- Miss a **proper and sound theory** for DevOps
  - What are the main concepts and principles?
  - observability, stateless architecture, reproducibility and replicability, accountability, software lifecycle automation
- Often engage **heterogeneous languages and environments**
  - tooling overhead
  - unsupported heterogeneity (interoperability, synchronization, coordination…)

Call for Participation – The *Teach DevOps* Initiative

- International initiative about teaching DevOps
  - List all the already existing materials
  - Share experiences, discuss good and bad practices, underlying concepts and principles, etc.
  - Develop and share various teaching materials related to DevOps
  - Develop a common case study for illustration, labs and projects





https://teachdevops.github.io