Title

DevOps for scientific research

Professors

Danilo Pianini
email
website

Abstract

Current scientific research, both in computer science/engineering and in other disciplines, requires software development of some form: novel algorithms, libraries, simulation tools, extensions for existing software products --- all these are examples of software artifacts which are often essential for reproducing the results proposed in scientific papers.
At the same time, novel techniques (actually, a mixture of pre-existing techniques) merging development and operations in a single process (DevOps) are increasingly adopted in industry, fostering quicker and more solid development and lower times to marked.
This PhD course has the goal of showing methods and tools for realizing, automating, sharing, and maintaining software for scientific research.
The course is structured in two theoretical lectures and a lab day, for a total of 20 hours in which the students will acquire knowledge and experiment with the tools proposed.
The initial part of the course will focus on techniques for effective teamwork on a shared code base, using git as enabling tool; showing possible workflows, their mapping with common software development practices, and drawing considerations on the trade-off between workflow robustness and overhead. It will then discuss appropriate hosting services for both open and closed source scientific projects (including LaTeX papers).
The second part of the course will focus on automation and portability of the process that, from source code, leads to the final artifact. It could be a scientific paper, a simulation generating a set of charts, or an executable

software tool: the philosophy backing the automation process is unchanged, and encompasses dependency management, building, testing, reporting, and customization of the process. Here, Gradle will be proposed as leading tool, along with a look on Python's setuptools.
The third part of the discussion will finalize the discussion on the DevOps process by showing continuous integration and continuous delivery, enabling the student with an entirely automated process producing a deployed artifact for each development action.
Finally, appropriate ways of distributing software will be discussed, including an overview of licensing, which more often than thought prevents companies from using existing scientific tools, boosting both their competitiveness on maket and visibility and relevance of scientific research (and at times ending in collaborations which may attract funding).

Location

Aula 2.13 / Laboratorio 3.1
Università di Bologna --- Campus di Cesena
Via Dell'Università, 50
47522 Cesena (FC)

Schedule

2019-09-09, 10:00 -- 13:00, Aula 2.13
2019-09-09, 14:00 -- 18:00, Aula 2.13
2019-09-10, 10:00 -- 13:00, Aula 2.13
2019-09-10, 14:00 -- 18:00, Aula 2.13
2019-09-11, 10:00 -- 13:00, Laboratorio 3.1
2019-09-11, 13:00 -- 17:00, Laboratorio 3.1

Final Examination

Students of the course will be asked to equip one of their scientific software projects with a reasonable subset (where applicable, all) the techniques and tools presented in the course.
In case there is no suitable software project, a new toy project can be used to demonstrate familiarity with the techniques encompassed by the course.
A short list of projects which have been positively evaluated in the past

include:

A Python library for cell counting in microscopic fluorent images

An Android client for the "IoT Manager" framework

The "Inteactive 3D Campus" web application

TuSoW: Tuple spaces over the Web