DevOps@MECH - a cloud infrastructure for reproducible research

Johan Philips[1], Herman Bruyninckx

Abstract— **Researchers from all domains are more and more confronted with the complexity of research data and software management. This is largely due to a transition towards computational sciences and is not always in par with the training and expertise of those researchers. As a result, one of the fundamental principles in science, reproducibility of research experiments and their results, is in danger. Another issue, partly caused by unmanageable software or undocumented experiments, is lack of research continuity, meaning it has become more and more difficult to build upon previous work. This is also due to the growing complexity in experimentation and lack of replication. In fact, science & engineering is faced with a reproducibility crisis. Also at our Department of Mechanical Engineering, researchers are in need of structural solutions to this growing problem. The goal of DevOps@MECH is to professionalise our research software and data management and educate our researchers using adequate technologies and tools. These tools help increase the reproducibility of their results and, hereby, improve the trustworthiness or their research. DevOps@MECH is a cloud infrastructure, coupled with continuous integration and deployment (CI/CD) and deploy using open source tools with an adequately large community: GitLab CI/CD [2] and Docker Swarm [1]. This abstract briefly describes the motivation behind DevOps@MECH, the current deployment with a set of typical use cases and an outlook for university wide deployment.**

I. Motivation

The reproducibilty crisis has hit all research domains. This has been acknowledged by the numerous publications on this issue in the past decade. A recent and well-cited survey in Nature [3] shows how severe this crisis is: 90% of those surveyed acknowledge there is a crisis, 52% of those surveyed in the field of Physics & Engineering acknowledge they are unable to reproduce their own results and 34% of those surveyed have no procedures in place to facilitate reproducibility.

We argue that one of the root causes is the lack of software training and skills for researchers, while their has been an increase of research software and data, partly due to the transition towards computational sciences. This discrepancy between the skill set and daily tasks of a an average researcher leads to unstructured and ill documented research experiments and endangers research reproducibility and research continuity. In literature this is sometimes referred to as the fourth paradigm, where science makes a shift toward data-intensive discovery and traditional researchers have to collaborate with data and software experts [6].

However, the basis of good research practice is research integrity, of which one of the fundamental principals involves properly managing the research process, including software and data. Moreover taking into account ethical issues, making honest claims based on validated data analyses & publishing articles with reproducible results.

Therefore, at our Department of Mechanical Engineering, we developed DevOps@MECH, a cloud infrastructure based on DevOps methodology, to optimally help researchers in properly maintaining and managing their research software and research data. By doing so, we increase reproducibility of research results, improve research continuity and improve credibility and trustworthiness.

Other work proposing DevOps for research exist, see e.g [5], [7] or [8], but with DevOps@MECH we aim to provide an institutional and production-ready solution that combines DevOps with the cloud.

II. Approach

A. DevOps methodology

DevOps is a software engineering term denoting the combination of software development and software operation [4]. It aims to automate and monitor all steps in a software lifecycle, from writing source code, running unit tests, compiling executables to releasing versions and deploying to a production environment. Also, it allows rapid prototyping by minimising the time between development phase and deployment phase. The DevOps@MECH infrastructure is built upon this methodology and automates as much as possible the workflow a researcher follows when developing research software and, by doing so, helps deliver better research software with reduced human error. Moreover, it helps with automating research experimentation workflows to improve research data management. It, therefore, implicitly increases reproducibility of research results as proper research software and data management are essential to be able to reproduce results.

It is true that DevOps is a software engineering technique, but many of the aspects and steps in the DevOps workflow are similar to those of a typical research experiment or process. A typical researcher (in Engineering) generates experimental data to analyse in order to publish the results in a paper. Research reproducibility dictates an independent research team should be able to do the same data analysis to reproduce those results. DevOps can help record all boundary conditions of an experiment to improve reproducibility. For example:

- Version control records a complete history of research data analysis scripts and allows the researcher to tag particular commits, e.g. for a particular publication. This

[1]Johan Philips is research expert in reproducible science at Department of Mechanical Engineering, KU Leuven, 3000 Leuven, Belgium johan.philips@kuleuven.be

way, he or she immediately finds the right version of those scripts to reproduce the results.
- Continuous Integration helps to automate the build, test and run of experiments to enable the researcher to run on different work stations (or remotely) with all dependencies included. This way, a colleague can easily reproduce the results.
- Continuous Integration also helps to share a software image of data analysis code, either open source or in binary form. This way, the researcher can share work with peers, or host it online, together with the publication.
- Continuous Delivery helps to automate the deployment of results, e.g. automatically publish a web page with documentation, references to papers, a dataset and the results. Generated completely from sources, scripts and publication.
- Continuous Delivery also helps to deploy complete simulation suites in the back end to offload algorithms to and run automated tests, freeing up valuable (CPU) time of the researcher.

B. Research data meta model

DevOps@MECH allows researchers to annotate their experiments before uploading their data sets. Properly tagging data with meta data will help manage research data much better. These meta data are stored in a database with unique identifiers to the storage location of respective data sets. Typically a research experiment would have meta data answering these questions:

- Who was involved in this experiment?
- What data were generated or collected?
- Why was this experiment conducted?
- When were the data generated?
- How were these data collected?

The more detailed information the researcher provides or by instrumentation of the research process the infrastructure can infer, the higher the chance of reproducibility. Transparancy of the whole research process is also key to improve trustworthiness of research results and overall credibility of research project.

C. DevOps@MECH deployment

The DevOps@MECH infrastructure is mainly built upon GitLab CI/CD and Docker Swarm. The former allows researchers to manage their research software and data analysis scripts with version control, build pipelines, simulation runs and automated deployment. The latter provides an IT infrastructure to host microservices (Docker services or stacks) such as databases, web applications, data analysis applications, or simulation environments. GitLab CI/CD and Docker Swarm have been fully integrated and we developed templates for typical CI/CD pipelines that researchers can use.

The Docker format also allows a uniform way of structuring research software and applications and a standardised and open way to share it with peers. The Docker Swarm runs on a 5 node cluster to allow load balancing if particular microservices are shared.

This makes DevOps@MECH a scalable solution for improved research data and research software management over the whole lifecycle of a research project. It offers tools at different stages and automates as much as possible the typical workflows a researcher follows. More over, it is from day one integrated with the professional ICT of the university with proper securiry, authorisation and authenication in place, in order to scale up in time.

It that sense, this infrastructure and the tools are the first step towards improved research reproducibility and trustworthiness.

III. Outlook

About a dozen research groups are currently using DevOps@MECH at our Department of Mechanical Engineering, storing circa 25 TB of research data and running about 40 microservices. Most microservices are frontend web applications to manage their research data and backend databases to store meta data from research experiments. The researchers are also supported by online tutorials on how to adequately manage their research software and data and also how to use the tools provided by DevOps@MECH. The cloud infrastructure is also well suited to scale up horizontally by adding more cluster nodes to the Docker Swarm.

This infrastructure is also serving as a pilot to scale up to a university wide research data and software management toolkit for researchers.

References

[1] Docker swarm. `https://docs.gitlab.com/ee/ci/`. Accessed: 2019-02-19.

[2] Gitlab continuous integration (gitlab ci/cd). `https://docs.gitlab.com/ee/ci/`. Accessed: 2019-02-19.

[3] Monya Baker. Is there a reproducibility crisis? a nature survey lifts the lid on how researchers view the crisis rocking science and what they think will help. *Nature*, 533(7604):452–455, 2016.

[4] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A software architect's perspective*. Addison-Wesley Professional, 2015.

[5] Maximilien De Bayser, Leonardo G Azevedo, and Renato Cerqueira. Researchops: The case for devops in scientific applications. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1398–1404. IEEE, 2015.

[6] Tony Hey, Stewart Tansley, and Kristin Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington, 2009.

[7] Ramtin Jabbari, Nauman bin Ali, Kai Petersen, and Binish Tanveer. What is devops?: A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, page 12. ACM, 2016.

[8] Karthik Ram. Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine*, 8(1):7, Feb 2013.