Towards Continuous Consistency Checking of DevOps Artefacts

Conference Paper · October 2021

DOI: 10.1109/MODELS-C53483.2021.00069

5 authors, including:

Alessandro Colantoni
Johannes Kepler University Linz
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Benedek Horváth
Johannes Kepler University Linz
6 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Ákos Horvath
Budapest University of Technology and Economics
59 PUBLICATIONS **991** CITATIONS
SEE PROFILE

Luca Berardinelli
Johannes Kepler University Linz
53 PUBLICATIONS **345** CITATIONS
SEE PROFILE

Towards Continuous Consistency Checking of DevOps Artefacts

Alessandro Colantoni
Johannes Kepler University
Linz, Austria
alessandro.colantoni@jku.at

Benedek Horváth*†, Ákos Horváth*
**IncQuery Labs cPlc.*
Budapest, Hungary
†Johannes Kepler University Linz
Linz, Austria
firstname.lastname@incquerylabs.com

Luca Berardinelli, Manuel Wimmer
Johannes Kepler University
Linz, Austria
firstname.lastname@jku.at

***Abstract*—DevOps tools are often scattered over a multitude of technologies, and thus, their integration is a challenging endeavour. The existing DevOps integration platforms, e.g., Keptn, often employ a family of languages for this purpose. However, as we have learnt from UML, SysML, and many others, a family of languages requires inter-model constraints to be checked in order to guarantee a high consistency between the different artefacts.**

In this work-in-progress paper, we propose a Model-Driven Engineering (MDE) approach for the continuous consistency checking of DevOps artefacts. First, we explicitly represent each artefact as a model, second, we establish links across them to set a navigable network of model elements; and third, we enable MDE services on top of this network.

We envision the possibility of using GitOps to pull the DevOps artefacts, executing services for checking consistency and performing model repairs, uploading the changes to the DevOps tools, and finally pushing the artefacts to Git, thus resulting in a continuous consistency checking process in practice.

***Index Terms*—DevOps, MDE, consistency management**

I. Introduction

According to Jabbari et al., DevOps is "a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices" [1].

The need to deal with a high number of categories and aspects to enable DevOps led to the proliferation of tools in heterogeneous technologies. By taking a look at the periodic table of DevOps tools [2], we can explore a selection over more than 400 DevOps tools, resulting in 120 DevOps products (e. g., Git, JMeter, Docker), organized into 17 categories (e. g., source control management, test, containers). This proliferation brought an explosion of specification and configuration languages for DevOps concerns. Consequently, to set up a DevOps platform that typically needs several tools, a DevOps engineer has to deal with several heterogeneous scripts that can be large, complex, and maintained by experts with specific competencies in the involved tools. Eventually, these scripts may need to be kept in sync. The complexity of the problem increases for all services that require a consistent overview beyond the boundaries of a single tool or script (e. g., consistency maintenance, change propagation, advanced dashboards).

Leveraging a single source of truth for such tools and scripts may be beneficial. In this respect, the term GitOps was first coined by Weaveworks in 2017 [3], and it refers to a practice for DevOps that adopts Git as the single source of truth for the artefacts. Divergences from the Git "truth" are detected by agents that trigger cloud activities like commits, deployments or rollbacks to maintain the synchronization.

In Model-Driven Engineering (MDE) [4], artefacts are models, which are the cornerstones throughout the whole engineering lifecycle. Models are machine-readable, abstract representations of the designed (software) systems, enabling analysis and design techniques on a higher level of abstraction. Recently, MDE is also being discussed in the DevOps context [5], [6].

In this paper, we tackle the problem of continuous consistency checking among heterogeneous DevOps artefacts. We outline a general approach from an MDE perspective and start its evaluation by implementing a Proof of Concept (PoC) to collect feedback about the approach's viability.

The proposed approach consists of several steps: (i) we transform the tools' textual configuration files to models envisioning a generalization of a previous work [7], (ii) weave the resulting models to each other via linking models, (iii) run validation rules as model queries leveraging the linking models for consistency checking, and (iv) execute model transformations to fix errors. Once corrected, the models are transformed back to the tools' native textual configuration files.

In order to support continuous consistency checking, we propose the adoption of GitOps practices, where all DevOps artefacts are maintained in Git. Agents can be set up to detect changes in the artefacts and trigger the consistency checking service, which can modify the artefacts and push them back to the repository.

For evaluation purposes, we adopt Keptn [8], an open-source tool for DevOps automation, configured via a family of languages based on JSON schema [9]. Keptn automatically pushes the configuration files to Git. We implemented a PoC [10] for consistency checking between two languages of the Keptn family used for Quality Assurance (QA): Service