PHD Thesis of Daniel Sokolowski

Reliable Infrastructure as Code for Decentralized Organizations

Abstract

IT must be reliable for organizations to thrive and quickly adaptable for their swift reaction to their environment. Agility is vital, and DevOps achieves these goals by empowering independent cross-functional teams in decentralized organizations and automating the entire software pipeline. Infrastructure as Code (IaC) is the critical tool to automate software operations, including infrastructure provisioning, application deployment, and configuration. Beyond simple IaC scripts, developers implement IaC programs in programming languages like TypeScript and Python. Such IaC programs are software, and their reliability is crucial to the functionality and security of the deployed systems. Still, techniques for the rapid development of reliable IaC programs are missing, limiting organizations' agility. Specifically, developers lack automation for deployment coordination and updating and quality assurance tools for, e.g., testing and verification.

We surveyed 134 IT professionals, finding that coordination across deployments is commonly needed and often requires manual coordination, even though IT professionals believe automated coordination yields better agility. However, automated approaches are centralized, limiting team independence and agility in decentralized organizations. To solve this issue, we propose automating coordination across deployments in a decentralized fashion through µs ([mju:z] "muse"), a novel IaC solution. With µs, teams have separate IaC programs, which express and jointly automate the correct order of operations across deployments. We further show how µs enables safe updating through IaC programs, preventing updates from breaking distributed transactions or workflows.

Beyond automating the coordination of IaC programs, we address the reliability of IaC program code. To unblock studies, we built a dataset of 37 712 public IaC programs. In initial analyses, only a vanishing fraction implements tests. We identified that available testing techniques are either slow and resource-intensive or require excessive development effort. To solve this dilemma, we propose ACT, an extensible automated unit testing approach that enables testing IaC programs quickly in hundreds of configurations, often without writing additional testing code.

This dissertation studies the coordination and testing of IaC programs. Empirically motivated, we present µs for safe deployment coordination and updating in decentralized setups and ACT for efficient testing of IaC programs. Our contributions nurture future research and reliable deployments in decentralized organizations, ensuring agility.