Title: Improving the Developer Experience of Dockerfiles

Author: João Pereira da Silva Matos

Supervision: Filipe Alexandre Pais de Figueiredo Correia

Date: November 21, 2022

Abstract

Nowadays, containerization is a technique used in a very large number of systems to address problems associated with deployment. The most popular tool used to perform this task is Docker. In order to use Docker, a developer must create a Dockerfile, a configuration file that is used to create the containers. Creating these files can be difficult [1], and even functional files can have problems. In fact, according to [2], "97.6% of the Dockerfile contains at least one security misconfiguration". For these reasons, there is a need for tools that aid with the generation and repair of Dockerfiles.

When it comes to generating the files, current solutions either require some preexisting documentation and the use of models [3] or a vast dataset that is hard to replicate [4].

For automatic repair, some work already exists [5]. However, the development experience could be improved by having this functionality integrated into a code editor where the problems and their solutions can be presented more clearly to the developer.

With this in mind, this dissertation aims to provide a tool that can help developers when creating or editing Dockerfiles and study the impact such a tool has on the development experience and the quality of the files that are created or modified using it.

To reach this goal, two tools created by former students to assist developers with these tasks, Hermit [6] (focused on generation) and Dockerlive [1] (focused on liveness and repair) will be improved and combined into the previously mentioned tool. The final tool will make it easier for developers to create Dockerfiles that follow best practices.

Furthermore, a study will be conducted to evaluate the performance of Dockerlive and the way it enhances the development experience. Industry participants will be involved in the study and several metrics like task completion time, container size and number of detected vulnerabilities will be used to assess the experience of the developers as well as the quality of the Dockerfiles produced.

Keywords: Dockerfile, Docker, File generation, File repair

ACM Classification: CCS - Software and its engineering - Software notation and tools - Software configuration management and version control systems

References

[1] David Alexandre Gomes Reis. Live Docker Containers. July 2020.

[2] Paolo Ernesto Prinetto, Dott Riccardo Bortolameotti, and Giuseppe Massaro. Security Misconfigurations Detection and Repair in Dockerfile. page 78.

[3] Chris Tomy, Tingmao Wang, Earl T Barr, and Sergey Mechtaev. Modus: A Datalog Dialect for Building Container Images. page 12, 2022.

[4] Hongjie Ye, Jiahong Zhou, Wei Chen, Jiaxin Zhu, Guoquan Wu, and Jun Wei. DockerGen: A Knowledge Graph based Approach for Software Containerization. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 986–991, July 2021.

[5] Jordan Henkel, Denini Silva, Leopoldo Teixeira, Marcelo d'Amorim, and Thomas Reps. Shipwright: A Human-in-the-Loop System for Dockerfile Repair. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1148–1160, May 2021.

[6] João Carlos Cardoso Maduro. Automatic Service Containerization with Docker. July 2021.