Title: Improving the Developer Experience of Dockerfiles

Author: João Pereira da Silva Matos

Supervision: Filipe Alexandre Pais de Figueiredo Correia

Date: November 22, 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 is not always straightforward [1], and even functional files can have problems. In fact, according to Prinetto et al. [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 development 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 any mistakes introduced by the automatic repair can be quickly addressed. Furthermore, having access to this functionality in an editor also increases the directness [6] of the developer's interaction with the file.

With this in mind, this dissertation aims to study how the development experience is affected by the introduction of automation in the Dockerfile creation and development processes.

To reach this goal, two tools created to assist developers with these tasks, Hermit [7] (focused on generation) and Dockerlive [1] (focused on liveness and repair), will be improved and combined into a tool that can help developers when creating or editing Dockerfiles. This tool will make it easier for developers to create Dockerfiles that follow best practices.

Furthermore, a study will be conducted to evaluate how Dockerlive enhances the development experience. Industry participants will be involved, and several metrics like task completion time, container size, and the number of detected vulnerabilities will be used to assess the experience of the developers as well as the quality of the Dockerfiles produced. This study falls under the category of Action Research.

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. doi: 10.1109/COMPSAC51774.2021.00133.

[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. doi: 10.1109/ICSE43902.2021.00106.

[6] John H. Maloney and Randall B. Smith. Directness and liveness in the morphic user interface construction environment. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology - UIST '95*, pages 21–28, Pittsburgh, Pennsylvania, United States, 1995. ACM Press. ISBN 978-0-89791-709-4. doi: 10.1145/215585.215636.

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

[8] Jordan Henkel, Christian Bird, Shuvendu K. Lahiri, and Thomas Reps. A Dataset of Dockerfiles. In *Proceedings of the 17th International Conference on Mining Software Repositories*, MSR '20, pages 528–532, New York, NY, USA, September 2020. Association for Computing Machinery. ISBN 978-1-4503-7517-7. doi: 10.1145/3379597.3387498.

[9] Investigating the inner workings of container image vulnerability scanners. https://oda.oslomet.no/oda-xmlui/bitstream/handle/11250/3017416/zarei-acit2022.pdf?sequence=1.

[10] Markus Linnalampi. Outdated software in container images. page 57.

[11] G. McArdle and R. Kitchin. Container Image Optimisation and Security Practices. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W1:19–25, September 2016. ISSN 2194-9050. doi: 10.5194/isprs-annals-IV-4-W1-19-2016.

[12] Scott McMillan. MAKING CONTAINERS EASIER WITH HPC CONTAINER MAKER. page 47.

[13] David Reis and Filipe F. Correia. Dockerlive : A live development environment for Dockerfiles. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–4, Roma, Italy, September 2022. IEEE. ISBN 978-1-66544-214-5. doi: 10.1109/VL/HCC53370.2022.9833145.

[14] Giovanni Rosa, Simone Scalabrino, and Rocco Oliveto. Fixing Dockerfile Smells: An Empirical Study, August 2022.

[15] Corrado Santoro, Fabrizio Messina, Fabio D'Urso, and Federico Fausto Santoro. Wale: A Dockerfile-Based Approach to Deduplicate Shared Libraries in Docker Containers. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 785–791, August 2018. doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00135.

[16] Jonas Sorgalla, Philip Wizenty, Florian Rademacher, Sabine Sachweh, and Albert Zündorf. Applying Model-Driven Engineering to Stimulate the Adoption of DevOps Processes in Small and Medium-Sized Development Organizations: The Case for Microservice Architecture. *SN Computer Science*, 2(6):459, November 2021. ISSN 2662-995X, 2661-8907. doi: 10.1007/s42979-021-00825-z.

[17] Yiwen Wu, Yang Zhang, Tao Wang, and Huaimin Wang. An Empirical Study of Build Failures in the Docker Context. In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 76–80, Seoul Republic of Korea, June 2020. ACM. ISBN 978-1-4503-7517-7. doi: 10.1145/3379597.3387483.