

Docker in Software Architecture

Daniel Koch¹ and Sebastian Sergelius²

¹daniel.koch@helsinki.fi

²github.com/sebazai

ABSTRACT

Please provide an abstract of no more than 300 words. Your abstract should explain the main contributions of your article, and should not contain any material that is not included in the main text.

Keywords: docker, software architecture, containers

1 INTRODUCTION

With the rise of DevOps and the use of Docker, containerization has become more and more relevant to think about even in software architecture. DevOps does not have any official definition. One formal definition is given by Jabbari et al. (2016):

”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”.

Another way to put DevOps is that it consists of the development of software and operations and that DevOps means that development, release, configuration, and monitoring are all done by the same people, rather than having separate teams for every part (Kousa, 2022).

Containerization is common as a part of DevOps. Docker is one of the most common containerization software (<https://www.aquasec.com/cloud-native-academy/docker-container/docker-orchestration/>) since its release in 2013. In this technology brief, we will explain what is meant by containerizations, how it compares to virtual machines and how do you use docker as a part of software architecture.

2 OVERVIEW OF THE MAIN FUNCTIONAL FEATURES (WHAT DOES IT DO FOR YOU?) IF APPLICABLE

Tähän varmaan siitä virtuaalikoneista vs kontit, kuten miten se abstrahoi os level vs hardware?

3 A TECHNICAL OVERVIEW (WHAT IS IT?)

4 DOCKER ARCHITECTURE

In this section we will be discussing the Docker architecture and its vital components that builds the technology under study. The Docker architecture is based on a client-server model where the client is called Docker client and the server is called Docker HostInc. (2022a); Security (ote). The client and the host do not necessarily need to be on the same system, as the client can communicate with the host using the Docker REST API through either a UNIX socket or a network interface. Docker images are written in a Dockerfile, which defines the steps how to build and run the image. These built images can be stored in a private or public Docker registry.

4.1 Docker Client

The Docker client is the component that the user interacts with. The client provides a command line interface (CLI) that enables the user to give commands to the Docker Host daemon.

4.2 Docker Host

The Docker host contains of many components, such as the daemon, images, containers, networking and storage, that provides the environment for running and executing applicationsSecurity (ote). The daemon is responsible for handling container related commands that are received from the client, such as pulling pre-built images from a registry, executing instructions defined in a Dockerfile and building and running the containers on the server.

4.2.1 Docker Images

Docker Images are read-only binary files that are used to build containersInc. (2022a). Images are built using a Dockerfile that defines the instructions for the daemon to execute. Each instruction creates a layer in the image built. If an instruction is changed in the Dockerfile, the daemon will only rebuild the image on the layers affected by the change. Once the image is built, you can run the image in a container.

4.2.2 Docker Containers

Docker Containers are isolated environments which runs applicationsSecurity (ote). The container contains all the necessary dependencies that are required for running the application. Running containers are isolated from other containers and also have a good isolation from the host machineInc. (2022b). Containers can be connected to networks and storagesSecurity (ote). The Docker environment has various options for defining the networking. }insert networking¿ here?

4.3 Docker Registry

A Docker registry is a service for storing and downloading images(Security, ote). A registry can be publicly or privately hosted. Some most known public registries are Docker Hub and Docker Cloud.

5 QUALITY ATTRIBUTES

The qualities (-ilities) that the TUS claims to support (what?) and justifications for the claims (how?) Based on how Docker is built, the system has some qualities hoisted in it. In this section, we will

5.1 Portability

One key aspect of Docker is its Portability. With software portability, we refer to the usability of the same software in different environments, such as the underlying operating system or hardware (Wikipedia, 2022). Portability is achieved based on how the Docker technology is built. Each container includes all the dependencies required for the application to work and only requires an underlying operating system and infrastructure that supports the Docker Engine (Kousa, 2022).

5.2 Security

There was a paper about this in IEEE iirc, going to find it later. But at least with docker, your software is isolated from other software.

5.3 Scalability

Scalability is one of the main selling points of Docker and especially if bundled with microservice architecture the scalability is great. With Docker, you can easily deploy as many containers of your service as you need and even adjust the number of containers during runtime.

5.4 Maintainability

Docker separates all applications into their own virtual environments. With this, different applications don't interfere with other applications and each application can have all its dependencies easily and reliably every time without affecting other software. This allows different applications to use different versions of the same dependencies.

6 WHAT IS IT NOT SUITED FOR (ANY INHERENT LIMITATIONS OR REPORTED ISSUES) - MAY BE CHALLENGING TO RE-SEARCH

Perhaps

7 ARCHITECTURAL IMPACT - WHAT ASPECT OF APPLICATION/SYSTEM ARCHITECTURE IS AFFECTED BY USING IT AND HOW?

yes

8 TOOL SUPPORT / AUTOMATION PROVIDED (CAN BE ALSO FROM 3RD PARTIES) TO HELP APPLICATION DEVELOPERS BE MORE PRODUCTIVE

9 ACKNOWLEDGMENTS

Additional information can be given in the template, such as to not include funder information in the acknowledgments section.

REFERENCES

- Inc., D. (2022a). Docker overview. <https://docs.docker.com/get-started/overview/>. [Online; accessed 28-January-2022].
- Inc., D. (2022b). Docker security. <https://docs.docker.com/engine/security/>. [Online; accessed 06-February-2022].
- Jabbari, R., bin Ali, N., Petersen, K., and Tanveer, B. (2016). What is devops? a systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016, XP '16 Workshops*, New York, NY, USA. Association for Computing Machinery.

- Kousa, J. (2022). DevOps with Docker. <https://devopswithdocker.com/>. [Online; accessed 27-January-2022].
- Security, A. (2022, howpublished = <https://www.aquasec.com/cloud-native-academy/docker-container/docker-architecture/>, note =). Docker architecture.
- Wikipedia (2022). Software portability — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Software%20portability&oldid=1062447939>. [Online; accessed 27-January-2022].