

Faster, Better, Docker!

You've heard terms like Docker and containers before, but you're wondering - what exactly is this all about? At the beginning, it can all seem a bit complicated, but don't worry, we'll answer some questions step by step. Find out what containers are all about, why Docker is a better alternative to virtual machines as an isolated application, and why you should focus on resource-saving options for hosting and cloud applications in this blog.

THE open source program for faster startup and better security

Our world is software-driven and easy deployment of software applications at minimal cost is a key requirement. Tech giants like Google, Facebook or Microsoft use containers to enable secure deployment of your processes in cloud production environments. The Docker software platform provides a technology that abstracts the application from a runtime environment. The runtime environment includes runtime library, standard libraries, programming interfaces, runtime variables, and hardware and software component via operating system functions. Similar to virtual machines, this is done through virtualization.

Docker is a set of Platform-as-a-Service (PaaS) products. Platform-as-a-Service refers to a service that provides a computing platform in the cloud for developers of web applications. These can be both rapidly deployable runtime environments (typical for web applications), but also development environments. These are used with little administrative effort and without purchasing the underlying hardware and software, called containers. These support the entire software lifecycle from design to development, testing, delivery, and operation of the web application over the Internet and the computer systems aligned for it.

Docker makes development efficient and predictable

Docker takes away repetitive, mundane configuration tasks and is used throughout the development lifecycle for fast, easy and portable application development – desktop and cloud. Docker's comprehensive and to end platform includes UIs, CLIs, APIs and security that are engineered to work together across the entire application delivery lifecycle.



Containers are a simple solution for dynamically deploying and transferring applications in your appropriate environments, entirely without compatibility issues. Likewise, they are stateless, isolated from each other, and do not persist data if this is not desired.

Containers - packaging microservices with clear advantages

To isolate applications in a loose environment (e.g., runtime environment or development environment), Docker works with an image-based deployment model. This makes it possible to share an application or a package of services with all their dependencies across multiple environments.

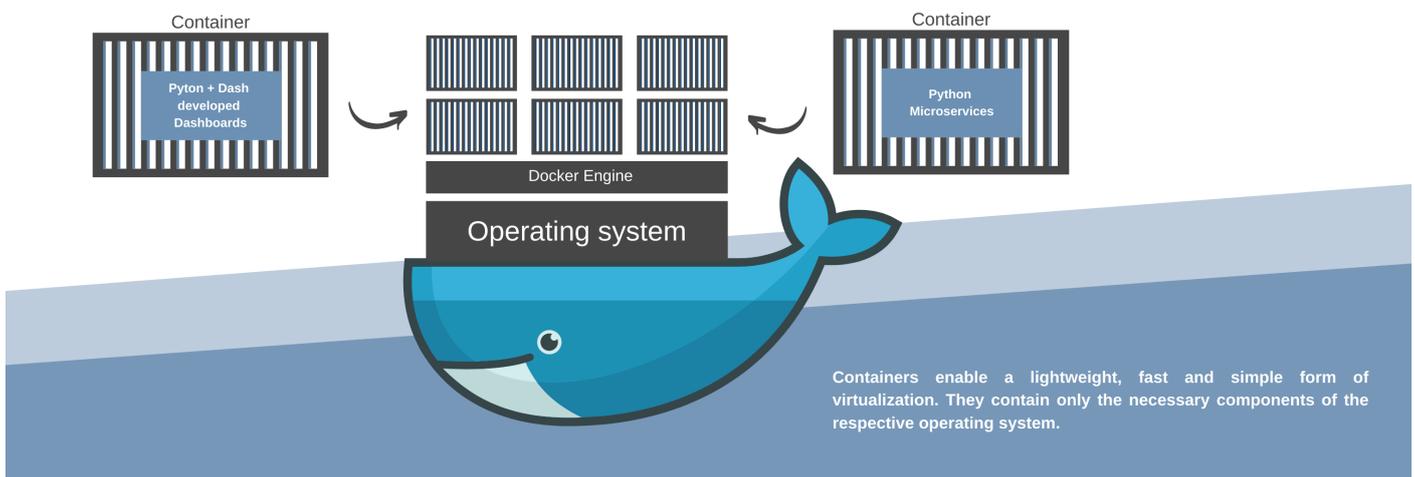
Simply explained: **The purpose of a container is the ability to run multiple processes and apps separately from each other and from the host system.** This results in more security during use and without interrupting the application, the infrastructure is used more efficiently and resources are saved at the same time. With resource quotas, it is even possible to limit the amount of memory a container can hold. This improves the efficiency of the environment and also prevents an imbalance in the distribution of these resources across their environment. Applications can be deployed with a significant reduction in time and successive versions of a container can even be tracked. Differences can be investigated and, moreover, previous versions can be reverted to.

The Docker application advantages:

- Fast application deployment
- Portability on all machines
- Unified environment
- Version control and component reuse
- Lightweight and minimal effort for rollouts
- Easier maintenance of individual modules
- High safety during application
- Optimal use of infrastructure
- Registry: The Registry is a stateless, highly scalable server-side application that stores and distributes Docker images. The Registry is open source and licensed under the Apache license



Fun fact: The term container is actually based on the physical container as we know it from ships, for example. From the outside, a shipping container is equipped with standardized interfaces and can thus be transported on any ship designed for it - regardless of its contents. This is exactly how it works with our Docker containers. Separated from its contents, the container can thus be executed on any computer system aligned for it.



Docker container in the application including operating system, Docker engine and the corresponding applications

The alternatives to Docker? We don't want to say that we don't know of any real alternatives, because for developers there are Python virtual environments, for example. However, these only cover a small part of what Docker can offer us. Docker as a web application makes our work efficient and predictable.

You now have your first Docker basics behind you. Of course, this is a very broad topic, so feel free to check out our **Docker vs. virtual machine** article.

In this blog, you got an overview of what Docker is, what we use containers for, and what the big advantage of this Platform-as-a-Service product is.

Do you have questions, suggestions or just want to say hello? Feel free to contact us via email: info@supperundsupper.com

