

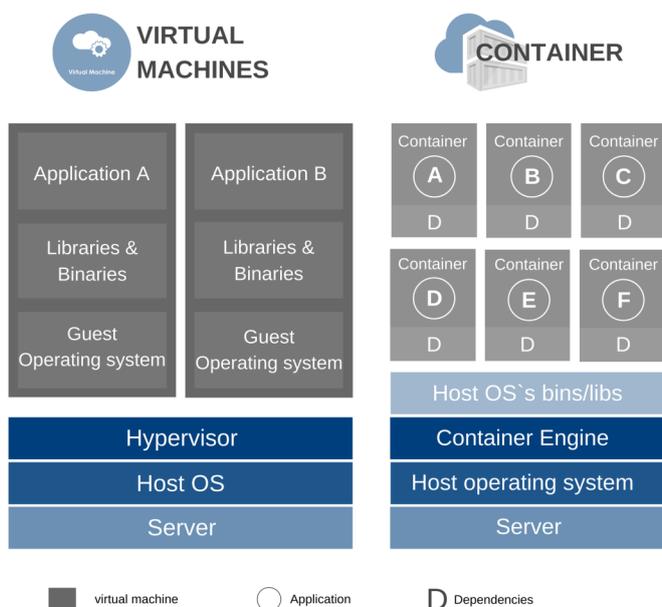
## DOCKER CONTAINER VS. VIRTUAL MACHINE

### Is Docker with containers as an isolated application really the better alternative to virtual machines?

We'd like to answer this question with a simple "yes" or "no" and you probably already guessed it - it's not that simple. You can read about how Docker works as a container-based technology and what the benefits are in our article "faster, better, Docker". In this blog, we'll talk about the differences between virtual machines and containers and help you get up to speed by describing useful terminology .

*Virtual machines* (also called VM's) are virtual computers that provide the same functions as physical computers - they run applications and an operating system. However, virtual machines are computer files that run on a physical computer. They function as separate computer systems.

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



Virtual machines and container applications and their dependencies

### The differences in the operating system

Virtual machines always run a complete operating system including the kernel. This demands more system resources, memory, and storage space. Docker, as a container type, on the other hand, only runs the necessary components of an operating system.

Containers can be configured so that only those services are included in the container that is necessary for running, for example, an app. More containers than virtual machines fit on a server.



## Differences in isolation & security

Should the software become infected with a virus, virtual machines provide a high level of security. They allow isolation from the host's operating system from other virtual machines. Containers offer somewhat less security here for the time being, as the applications are mostly isolated from the host and other containers in a simpler way. The emphasis here is on for the time being, because security can be increased, for example by the Linux kernel extension SE-Linux (Security-Enhanced Linux). This consists of a kernel patch and extensions for system programs and thus again ensures a high degree of security.

## The main differences between virtual machines and containers

	VIRTUAL MACHINES	CONTAINERS
<b>Startup Time</b>	MINUTES	MILLIESECONDS
<b>Disk Space</b>		
<b>Portability</b>		
<b>Efficiency</b>		
<b>Operating System/ Kernel</b>	DEDICATED	SHARED

*Detailed differences between virtual machines and Docker*

## Will container-based technologies completely replace virtual machines?

Now a question arises - do VMs still have their raison d'être at all, because we read many advantages that speak for Docker and containers? Lilli Kaufhold, Senior Data Scientist, has an answer to this and explains what part container-based technologies play in the everyday life of Supper and Supper.

*At Supper and Supper, we use both technologies - Docker and also virtual machines. Our standard use case is to launch a virtual machine in the cloud and then deploy our services in Docker containers on those machines. Usually, the host provider (e.g. Azure or AWS) takes care of setting up the VM with a particular image. We don't see the real hardware underneath. In my day-to-day business, I only actively manage Docker.*

*So virtual machines still play an important role and will continue to do so. They may just be less visible to the user in the future as they are integrated as a standard security measure.*

## Important terminologies in Docker

As already stated beforehand, you will find some important terminologies here, which you will come across again and again in the area of container-based technologies.

### Docker-Image

A Docker image is a file that consists of multiple layers and is used to execute code in a Docker container. It is a set of instructions from which Docker containers are created.

### Docker-Container

A Docker container is a runtime instance of an image. It allows developers to package applications with all their components, such as libraries and other dependencies.

### Docker- File

This is a text document that contains necessary commands that help to assemble a Docker image during execution. The image is created using a Docker file.



## Docker-Engine

This is the software that hosts the containers and is a client-server-based application. It consists of several main components:

SERVER	The server is responsible for creating and managing Docker images, containers, networks, and volumes on the Docker server. It is called a daemon process
REST-API	The REST API defines how the applications can interact with the server. It instructs it what to do.
CLIENT	The client is a Docker command-line interface (CLI). This allows us to interact with Docker using the Docker commands
DOCKER HUB	Docker Hub is the official online repository where you can find other Docker images that are available. You can find, manage, and share container images.

As we have now learned, Docker is indeed a good alternative to virtual machines, however, it makes sense and is still possible to rely on both technologies, as we did with Supper and Supper.

Still have questions, suggestions or just want to say hello? Feel free to contact us via email: [info@supperundsupper.com](mailto:info@supperundsupper.com)