

# *Container-Technology*

## *A Comprehensive Overview*

*An overview of how software is developed and implemented in a faster, simpler and more reliable manner through container solutions.*

*This overview is designed to answer the principal questions which arise around container technology and to give a fundamental understanding of the topic.*

# Table of Contents

<b>The Basics</b>	
What is Container Technology or Container Virtualisation?	3
What are the benefits provided by a container solution?	3
What are the differences between container solutions and virtual machines (VM)?	4
What are the risks associated with a container solution?	4
<b>The Implementation</b>	
Which specific improvements does container technology bring to update and implementation processes?	5
What are container orchestration platforms?	6
What is Kubernetes?	6
What is DevOps?	6
Are DevOps and the use of containers always inseparable?	6
Are there best practices for deploying container environments?	7
How steep is the learning curve for working with containers?	7
What is the best way to ensure a good start to a container project?	7
Container technology: managed or unmanaged?	7
Where can I find support for the development and implementation of container technology?	8
Where can I find tutorials/answers to specific application-based questions concerning containers and Kubernetes?	8
Which network systems/architectures are compatible with containers? Where are the limits?	8
How safe are container solutions? Compared to virtual machines?	9
How can internal buy-in for the implementation of containers be obtained?	9
How do I get my team on board?	9
Are there alternatives to Docker?	9

# The Basics of the Container Technology

## 1. WHAT IS CONTAINER TECHNOLOGY OR CONTAINER VIRTUALISATION?

Docker describes a container as a *standardised software unit*.

Container virtualisation enables an application to run without the need for a host operating system (OS). The container thus encapsulates an entire application including code, dependencies and configurations in one clearly-defined format.

## 2. WHAT ARE THE BENEFITS PROVIDED BY A CONTAINER SOLUTION?

The standardisation of a container and its independence from the host OS provide attractive benefits for developing and running software:

### #1 Applications can be more easily scaled with containers.

Container virtualisation is based on autonomous, independently-functioning units. This allows applications to be scaled up or down quickly and flexibly as needed.

### #2 Containers make applications start faster.

Starting up containers is faster as they eliminate the need for a dedicated guest operating system which would otherwise need to be booted up first and shut down again.

### #3 Containers use available resources efficiently.

Because containers have no need for a dedicated operating system, they actively save physical resources which are in turn available for other applications.

### #4 With containers, software can be developed in a more agile manner.

Containers are fast to build and deploy and they are reproducible, for example, a container started in the development environment is exactly the same as the one in production.

### #5 Containers are portable.

Since container applications are run largely independently of a host operating system, they can be migrated without much effort.

Do you wish a more thorough analysis? Read more about the advantages & disadvantages of container technology in our [nine blog](#) or download our Tech Whitepaper "[Container Virtualisation and its Usage](#)".

Read more about the [different solutions in Container Technology](#) in our blog.

### 3. WHAT ARE THE DIFFERENCES BETWEEN CONTAINER SOLUTIONS AND VIRTUAL MACHINES (VM)?

Containers and virtual machines (VM) represent two different approaches to hosting and running applications in a virtual environment. For VMs, a so-called 'hypervisor' distributes the server's physical resources to virtual machines within the operating system. Additionally, virtual machines need a dedicated operating system for each application. When using containers on the other hand, virtualisation is carried out directly on the operating system level. Therefore, containers can use the available resources significantly more efficiently than virtual machines (VMs) and are also booted up more quickly.

### 4. WHAT ARE THE RISKS ASSOCIATED WITH A CONTAINER SOLUTION?

Those who have had an in-depth look at the topic of container technology may have heard the following concerns:

#### « Containers are not secure enough for our requirements »

Containers are no less secure than any other environments. Just like with other setups, security depends primarily on the way the environment is set up and maintained.

#### « Container management is far too complex for us »

Just like any other change, migrating to a container-based system initially comes with a certain amount of effort. Applications need to be prepared, employees need to be trained and in some cases, deployment and work processes need to be automated. But in the long run, this investment is worth it, and it is quickly offset by the benefits of containerisation as outlined above. The expenditure can be further reduced with the support of knowledgeable specialists.

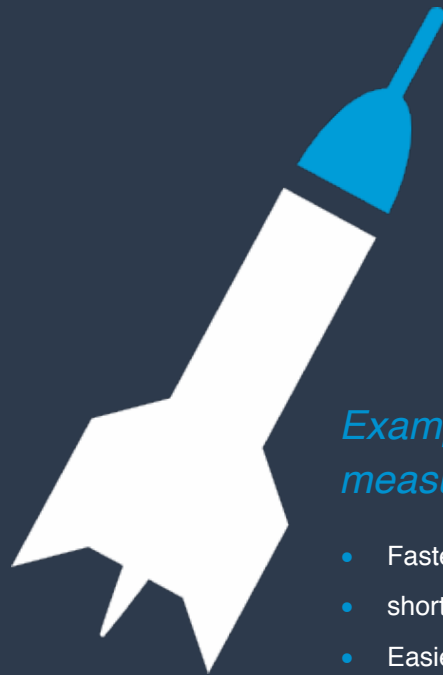
#### « We have a deployment strategy. We do not need containers »

Containers do not replace a deployment strategy, but rather complement and optimise it. The implementation of container technology does not mean that you have to turn your current setup completely on its head. There are various options for integrating containers into an existing IT strategy, e.g. in combination with or supplemental to virtual machines.

## *This is how you can implement Container Technology...*

### **5. WHICH SPECIFIC IMPROVEMENTS DOES CONTAINER TECHNOLOGY BRING TO UPDATE AND IMPLEMENTATION PROCESSES?**

During update and implementation processes, speed and absence of errors are important measurement factors. And this is exactly where the benefits of containers can be felt most strongly.



#### *Example: Speed as measurement value*

- Faster boot-up
- shorter feedback cycles
- Easier scaling (it is only necessary to set the configuration and the platform takes it over automatically)
- Rolling Deployments without interruptions (New versions are available immediately)

## **6. WHAT ARE CONTAINER ORCHESTRATION PLATFORMS/CLUSTER MANAGERS?**

Container orchestration platforms are used to manage individual containers. They form a dynamic system which groups and places containers, manages their connections and ensures communication between them. This type of platform (or cluster) can be run on-premise or on established cloud platforms (e.g. Google Cloud). The container orchestration can be managed with OpenShift or Kubernetes, for example.

## **7. WHAT IS KUBERNETES?**

Kubernetes is used to orchestrate several containers. This means that the open source platform supports automation of the processes of setting up, running and scaling containerised applications, and providing storage.

## **8. WHAT IS DEVOPS?**

DevOps as an organisational and leadership culture is based on three principles, "the three ways": the flow principle, the feedback principle and the principle of continuous learning and experimentation.

Read more about the [«The Three Ways»](#) - the DevOps Principles and Why Other Teams in your Company Could Benefit from them.

These three ways are essential to implementation in order to achieve the overarching goal of getting better and faster. The three ways serve as a framework for processes, procedures and practices which are clearly focused on streamlining collaboration.

## **9. ARE DEVOPS AND THE USE OF CONTAINERS ALWAYS INSEPARABLE?**

DevOps as a philosophy and containers as a technology make an optimal team. However, they are not interdependent, which means that using the DevOps method is possible without the use of containers, just as container virtualisation is possible without DevOps. The Docker deployment strategy generally advises having as little manual intervention as possible (automatisation) in order to ensure that containers are run as smoothly as possible and to benefit from all advantages. The DevOps method can also be fully employed when working with conventional servers, for example where the simplification of organisational processes is concerned.

# *Rely on Docker or Kubernetes if you want to start implementing containers*

## **10. ARE THERE BEST PRACTICES FOR DEPLOYING CONTAINER ENVIRONMENTS?**

Docker Compose is often the first point of contact with container orchestration. Yet, this tool is primarily suitable for local development and thus cannot appropriately be used as a container orchestration platform. For Kubernetes, there are tools, such as Kompose and compose-on-kubernetes, which translate Docker Compose files to API objects. However, a fundamental understanding of these objects is a necessary condition for deploying applications, autonomously running the troubleshooting process in case of problems and for debugging the installations.

## **11. HOW STEEP IS THE LEARNING CURVE FOR WORKING WITH CONTAINERS?**

This depends on the desired degree of autonomy. When running the container platform is entirely outsourced, basic container knowledge is sufficient. When the platform is run in-house, migrating to a container-based architecture requires comprehensive training for developers and systems engineers. Without a profound understanding of the use of Docker and of the chosen orchestration platform, this is not possible.

## **12. WHAT IS THE BEST WAY TO ENSURE A GOOD START TO A CONTAINER PROJECT?**

A basic workshop on Docker and Kubernetes makes for an ideal basis. The acquired knowledge is a fundamental prerequisite for understanding containers.

## **13. CONTAINER TECHNOLOGY: MANAGED OR UNMANAGED?**

Due to the complexity brought forth by building and running the underlying infrastructure and the rapid development of the technology which is used, we advise handing this part of the operation on to a managed service provider. By doing this, you can focus on your application and on running your containers.

#### **14. WHERE CAN I FIND TUTORIALS/ANSWERS TO SPECIFIC APPLICATION-BASED QUESTIONS CONCERNING CONTAINERS AND KUBERNETES?**

The Kubernetes and Docker blogs are a good knowledge base for specific questions. Various online tutorials can be found there, as well as on several other specialist blogs, which can offer a quick answer to particular problems.

#### **15. WHICH NETWORK SYSTEMS/ARCHITECTURES ARE COMPATIBLE WITH CONTAINERS? WHERE ARE THE LIMITS?**

Containers can act as a useful supplement or alternative to VMs. In principle, containers can also be linked to VMs within a network if this is required in the specific case. Important: containers run on Windows or Linux kernels, but Windows options currently come with some significant drawbacks.

#### **16. HOW SAFE ARE CONTAINER SOLUTIONS? COMPARED TO VIRTUAL MACHINES?**

Containers are just as secure as traditional server solutions and VMs. What is essential is how these environments are set up. The issues regarding operations and security measures differ only when it comes to the security model. Sensible options for ensuring security can be chosen on every level of the architecture - whether containers are used on-site or in the cloud. Ultimately, security in general strongly depends on the security of the application itself.

[Read in an interview with our Platform Team Leader Tom Whiston, how containers can improve security in the cloud.](#)



[Read our case studies](#) to find out how our customers have experienced working with us as external experts in the field of cloud computing and containers.

### **17. HOW CAN INTERNAL BUY-IN FOR THE IMPLEMENTATION OF CONTAINERS BE OBTAINED?**

Just as with any other innovation or investment, the most convincing arguments when talking about implementing containers are showing cost-cutting effects and an improved time to market. Experience has shown that practical case studies and examples help enormously with this.

### **18. WHAT ARE THE REALISTIC COSTS AND RESOURCES NEEDED WHEN IT COMES TO IMPLEMENTING AND RUNNING A CONTAINER SOLUTION?**

Cost varies depending on the size and complexity of the project, the training effort needed, whether the container platform is run on an infrastructure in-house or off premises. Therefore it is important to analyse the situation thoroughly beforehand – maybe with the advice of an external expert – to be able to assess the upcoming expenditure.

### **19. HOW DO I GET MY TEAM ON BOARD?**

Migrating to a container setup can be linked to a large initial effort and a steep learning curve. Acceptance for the change in the team is best achieved by clearly pointing out the advantages of a container solution: saving time through automated processes, agile working methods, scalability, etc.

### **20. ARE THERE ALTERNATIVES TO DOCKER?**

Owing to the open source character of container technology, there is a constant stream of new tools which replace Docker applications. There is, for example, Podman and Buildah. The former replaces Docker command line and can be used to run standalone containers. Buildah, a shell-based Linux tool, allows for the creation of container images. Additionally, there are several runtime solutions, such as CRI-O, which is used instead of Docker as a runtime on Kubernetes.

---

**Book now a call with a nine expert in the field of container technology...**

...and learn from us how you can benefit from container technology.

**BOOK A CALL**

---

## *This may also be of interest to you:*

### WHITEPAPER «CONTAINER VIRTUALISATION AND ITS USAGE»



*Learn how the use of containers can support you and optimise your processes.*

- Everything you need to know about containers
- Different deployment processes - classic vs. containerised
- Process optimisation thanks to containers
- How a CI/CD pipeline can look like
- Influence of Docker and OpenShift on Development Processes

**DOWNLOAD NOW**