



# Was gibt es Neues im Docker-Universum?

*Nicholas Dille*

Docker Captain, Microsoft MVP

[Container] 2017  
Conf]



# Nicholas Dille

Ehemann, Vater, Geek, Aikidoka

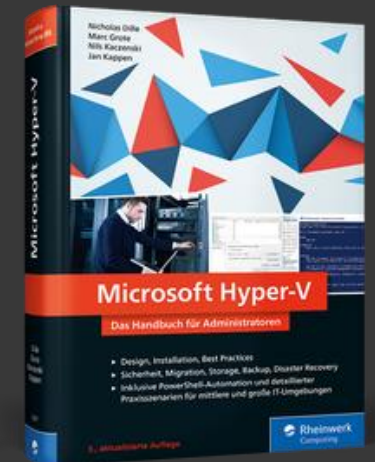
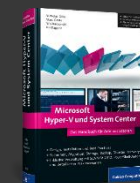
Microsoft MVP seit 2010

Docker Captain seit 2017

DevOps Engineer @ Haufe-Lexware

 <http://dille.name/blog>

 @NicholasDille



# Was macht Docker so besonders?

Am Anfang waren logische Partitionen (LPARs)

Veröffentlicht 1972 von unseren Eltern

Dann kamen Linux Containers (LXC)

Veröffentlicht 2008 von unseren älteren Geschwistern

Schnittstelle zu Control Groups und Namespaces

Zuletzt erblickt Docker das Licht der Welt

Gegründet 2013 von Solomon Hykes

Revolution der Container-Verwaltung durch Automatisierung

# Was sind Container?

#TBT: Container sind Prozessisolation

Umsetzung durch den Kernel

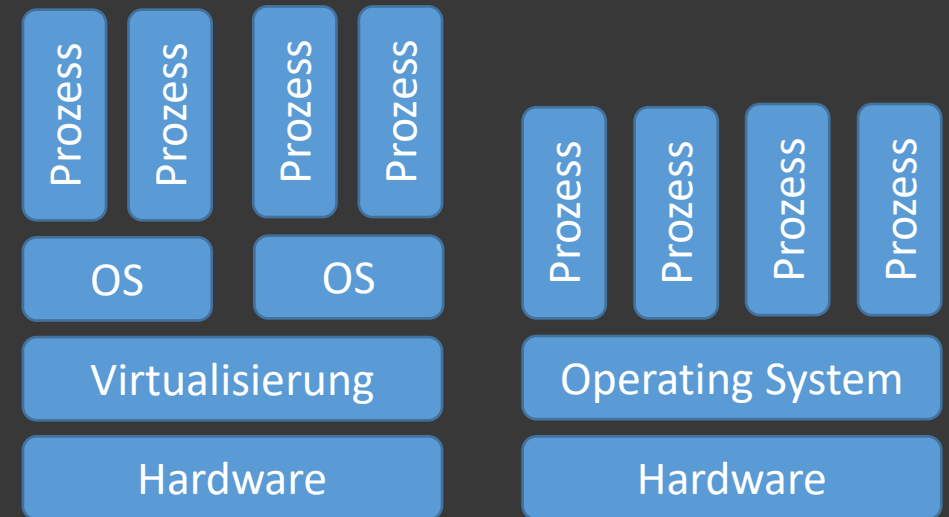
Ressourcen müssen geteilt werden

## Container vs. virtuelle Maschinen

Andere Ebenen der Virtualisierung

Gemeinsame Hardware vs. gemeinsamer Kernel

Container sind eine weitere Option



# Was Docker bietet

## Enterprise Edition

Universal Control Plane

Trusted Registry

Docker Certified

## Community Edition

Docker CLI

dockerd

Docker Hub

Machine



# Das Moby Projekt

Angekündigt auf der DockerCon (Mai 2017)

Veröffentlichung der Komponenten von Docker CE

## Bestandteile

runC / containerd / LinuxKit / SwarmKit

Notary / Registry / Compose

libnetwork / DataKit / BuildKit / VPNKit / InfraKit / HyperKit

## Downstream-Projekte

Docker CE / EE

Balena (Container Engine für IoT)

Steuerung durch Technical Steering Committee

# Was Docker bietet

## Enterprise Edition

Universal Control Plane

Trusted Registry

Docker Certified

## Community Edition

Docker CLI

dockerd

Docker Hub

Machine



SwarmKit

containerd

runC

LINUXKIT

Compose

Registry

Notary

libnetwork

DataKit

BuildKit

VPNKit

InfraKit

HyperKit

# LinuxKit

Minimalistisches, unveränderliches Linux

Definition in YAML

Bauen mit Moby

Basiert auf containerd

Alles ist ein Container (auch System-Dienste)

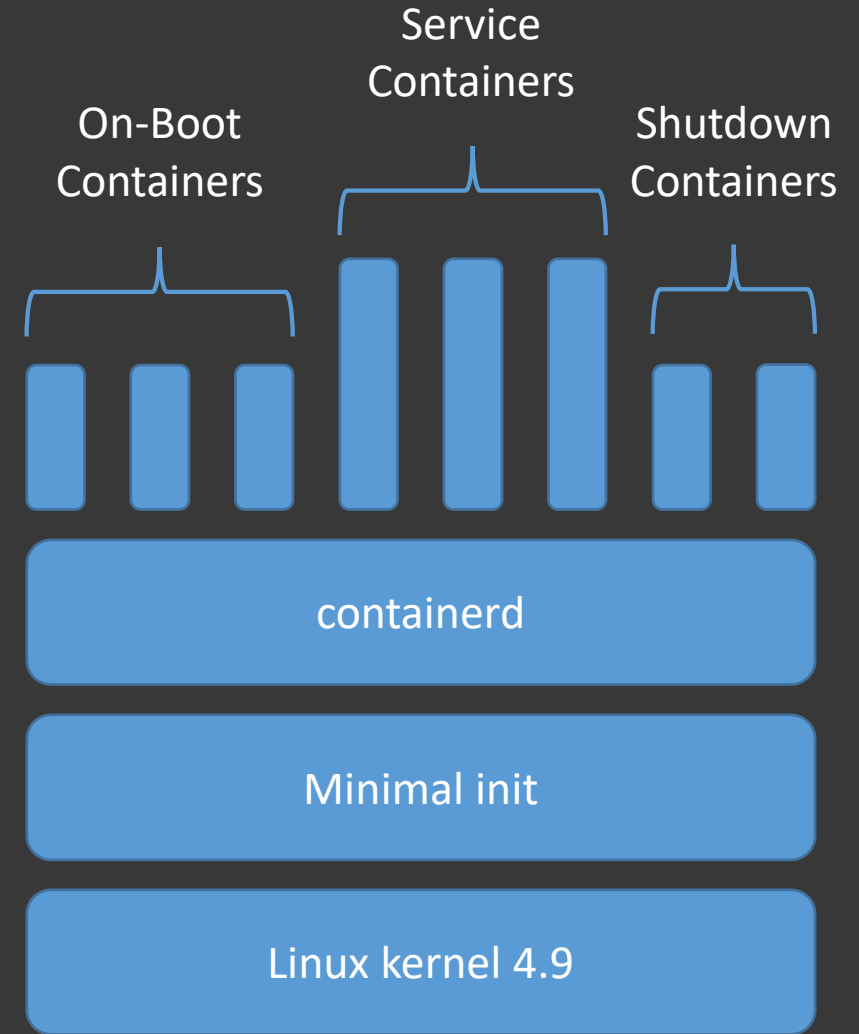
Images basieren auf Alpine Linux

Images sind mit Notary signiert

Unterstützung diverser Plattformen

DEMO

[https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo1\\_LinuxKit.md](https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo1_LinuxKit.md)





# containerd

## Verwaltung von Containern

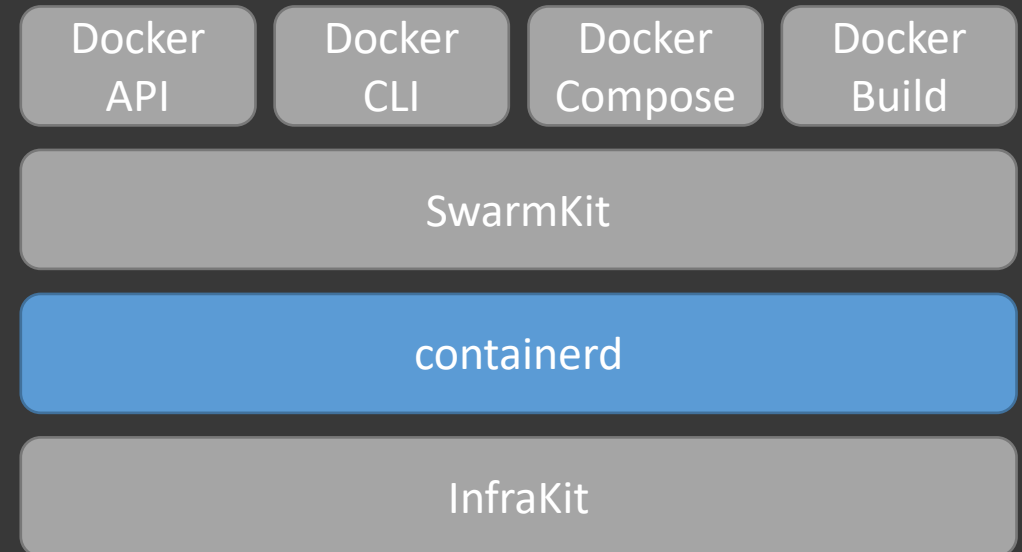
Basiert auf runC (reine Runtime)

Fügt Pull und Push von Images hinzu

gRPC-API über UNIX-Socket

Kommandozeilenwerkzeug `ctr`

Gehört der CNCF



## DEMO

[https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo2\\_containerd.md](https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo2_containerd.md)

# Notary

Datenintegrität sicherstellen

Signieren beliebiger Daten

Prüfen von Signatur

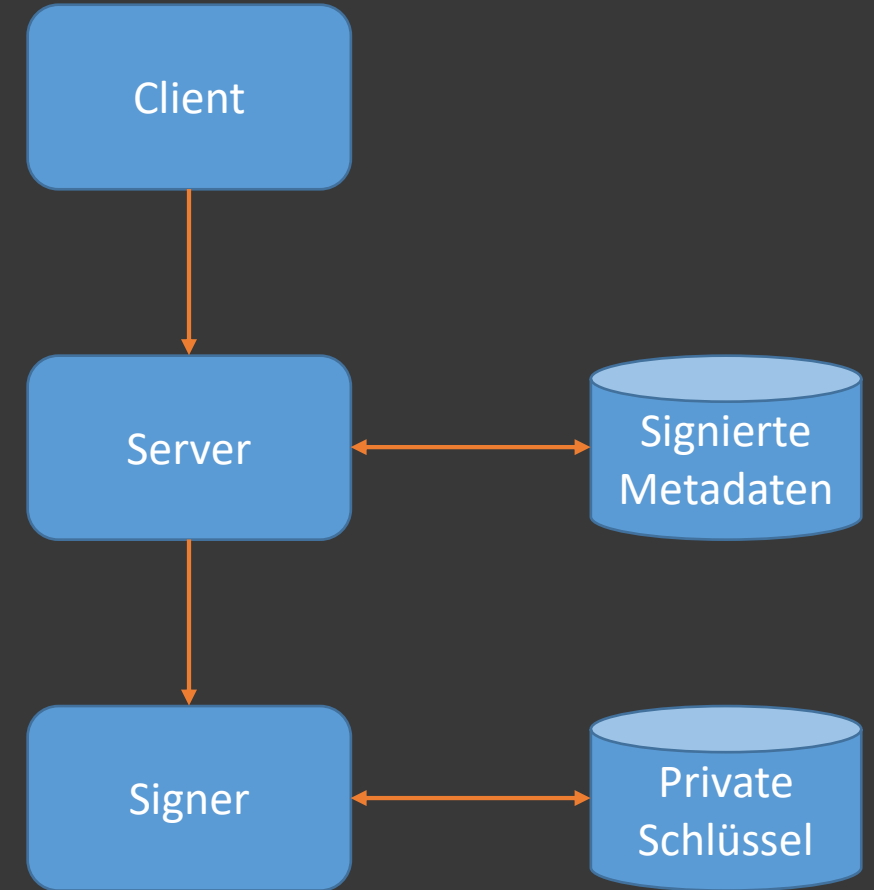
Client/Server-Architektur

Grundlage für Docker Content Trust

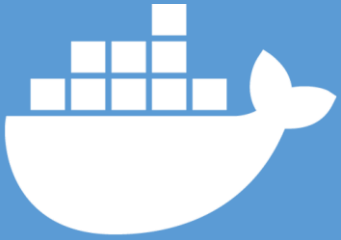
Gehört der CNCF

DEMO

[https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo3 Notary.md](https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo3%20Notary.md)



# Docker im Container-Ökosystem



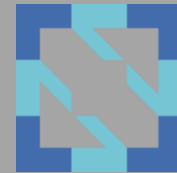
...initiierte  
(April 2017)



12 Projekte

...spendete containerd (März 2017)  
und Notary (Oktober 2017)

...spendete Spezifikation und runC (Juni 2015)



**CLOUD NATIVE**  
COMPUTING FOUNDATION



**OPEN** CONTAINER  
INITIATIVE

# Docker ♥ Kubernetes

Angekündigt auf der DockerCon EU (Oktober 2017)

Integration in Docker Universal Control Plane (UCP)

Verwaltung von Stacks auf SwarmKit und Kubernetes

Übersetzung von docker-compose.yml

Bundled Docker CLI and Kubernetes CLI

Installation auf neuen Nodes via Docker UCP

Enthalten in Docker für Mac

Enthalten in Docker für Windows

Docker für Mac/Windows installiert Kubernetes (single node)



# Was Docker bietet

## Enterprise Edition

Universal Control Plane

Trusted Registry

Docker Certified



## Community Edition

Docker CLI

dockerd

Docker Hub

Machine



SwarmKit

containerd

runC

LINUXKIT

Compose

Registry

Notary

libnetwork

DataKit

BuildKit

VPNKit

InfraKit

HyperKit

Kubernetes



# Linux Container on Windows (LCOW)

## Linux-Container unter Windows

Voraussetzung ist Windows (Server) 1709

Benötigt Containers und Hyper-V

## Linux-Container laufen unter Hyper-V

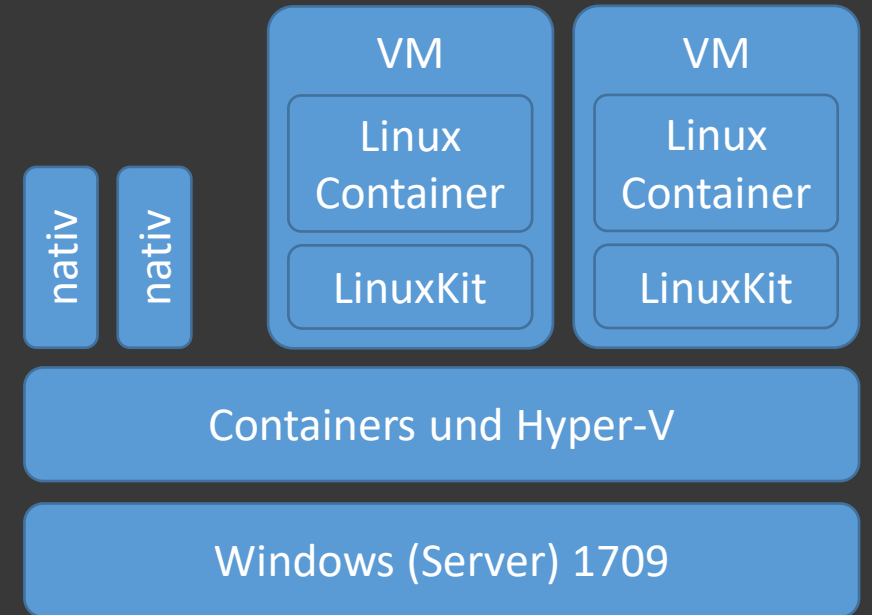
Jeder Linux-Container erhält eine dedizierte VM

## Universelle Docker CLI

Es zeigt Container beider Plattformen an

## DEMO

[https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo4\\_LCOW.md](https://github.com/nicholasdille/Sessions/blob/master/2017-11-15%20Docker%20%40%20ContainerConf%20Mannheim/Demo4_LCOW.md)



# Multi-Stage Builds

## Neue Syntax für Dockerfile

Bauen in mehreren Schritten

Mehrere FROM-Abschnitte

## Vorteile

Trennen von Build Env und Runtime

Kleinere Images

Weniger Pakete installiert

## Pipelines statt Multi-Stage Builds

```
$ cat HelloWorld.java
class HelloWorld {
    public static void main(String[] a) {
        System.out.println("Hello world!");
    }
}
```

```
$ cat Dockerfile
FROM openjdk:8-jdk as builder
COPY HelloWorld.java .
RUN javac HelloWorld.java

FROM openjdk:8-jre
COPY --from=builder /HelloWorld.class .
CMD [ "java", "HelloWorld" ]
```

# Multi-Arch Images

Images funktionieren nur auf einer Plattform

Container laufen aber auf vielen Architekturen und Betriebssystemen

Erstellen „virtueller“ Images

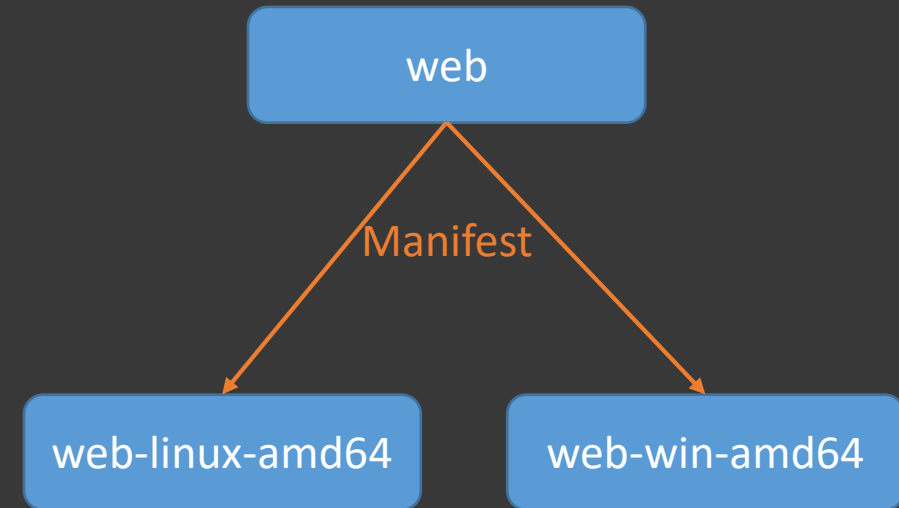
Spezielles Manifest verweist auf äquivalente Images unterschiedlicher Plattformen

<https://github.com/estesp/manifest-tool>  
(by Docker Captain Phil Estes)

Offizielle Images sind bereits umgestellt

In der Tiefe von Stefan Scherer (Docker Captain)

<https://youtu.be/kvJkKhkAnCM>





# Multi-Arch Images: openjdk

```
$ docker run mplatform/mquery openjdk:8-jdk
```

```
Image: openjdk:8-jdk
```

```
* Manifest List: Yes
```

```
* Supported platforms:
```

- linux/amd64
- linux/arm/v5
- linux/arm/v7
- linux/arm64/v8
- linux/386
- linux/ppc64le
- linux/s390x

```
$ docker run mplatform/mquery openjdk:8-jdk-nanoserver
```

```
Image: openjdk:8-jdk-nanoserver
```

```
* Manifest List: Yes
```

```
* Supported platforms:
```

- windows/amd64:10.0.14393.1770

# Multi-Arch Images: hello-world

```
$ docker run mplatform/mquery hello-world
```

```
Image: hello-world
```

```
* Manifest List: Yes
```

```
* Supported platforms:
```

- linux/amd64
- linux/arm/v5
- linux/arm/v7
- linux/arm64/v8
- linux/386
- linux/ppc64le
- linux/s390x
- windows/amd64:10.0.14393.1770
- windows/amd64:10.0.16299.19

# Play with Docker

## Cloud-Service zum Erlernen von Docker

Kostenfreie Docker-Hosts für vier Stunden

Jeder Host nutzt Docker-in-Docker (dind)

<https://labs.play-with-docker.com/>

## Tooling

Anmeldung per SSH (ssh -p 1022 10-0-1-3-48a594c4@host1.labs.play-with-docker.com)

Treiber für docker-machine (<https://github.com/play-with-docker/docker-machine-driver-pwd>)

## Kudos an die Autoren

Docker Captains Marcos Nils und Jonathan Leibiusky

# Ökosystem

## Rancher

Grafische Verwaltung diverser Orchestrierer (v1)

Open Source-Konkurrent zu Docker UCP

Verwaltung für Kubernetes (v2)

## Storage

Persistenter Storage ist eine Herausforderung

Docker hat Infini gekauft

## Security

Zusätzliche Abschottung

Isolation des Netzwerkverkehrs





# Wo steht Docker heute?

Getrieben durch Shareholder Value

Docker EE muss Community-Aktivitäten finanzieren

Standardisierung des Ökosystems

Docker ist eine der treibenden Kräfte

Container sind Mainstream

Relevanz durch Zusammenarbeit

Konkurrenten wachsen zusammen

Docker öffnet sich Kubernetes

Inwiefern öffnet sich Kubernetes auch Docker?!

