



Betriebssysteme

Winter 2024

Peter Sturm

4. Container

2



Virtuelle Maschinen

- Pflegen wir, als wären es Reale
- Installieren Updates
- Schauen wir uns bei Fehlern genauer an
- Geben wir bedeutungsvolle Hostnamen
- Hängen an deren Existenz





OS Layer Virtualisierung

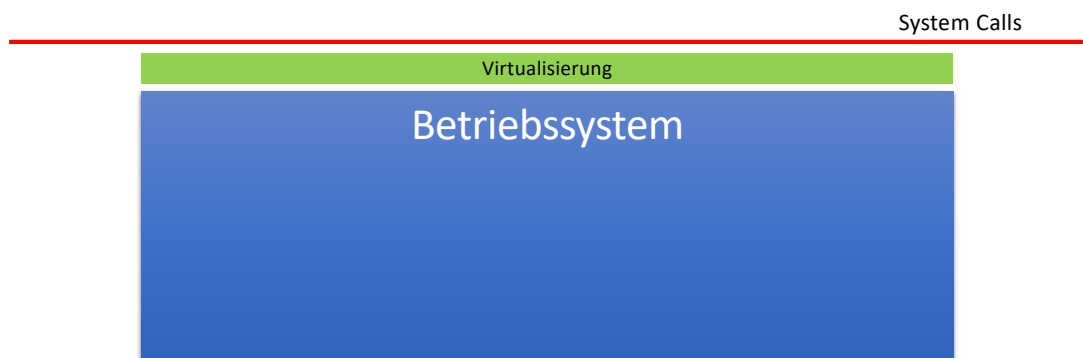
- OS virtualisiert hinter dem System Call API
 - chroot, Jails, u.a.

System Calls

Betriebssystem

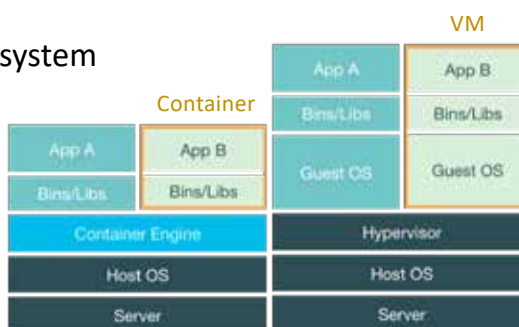
OS Layer Virtualisierung

- OS virtualisiert hinter dem System Call API
 - chroot, Jails, u.a.



Virtuelle Maschine → Container

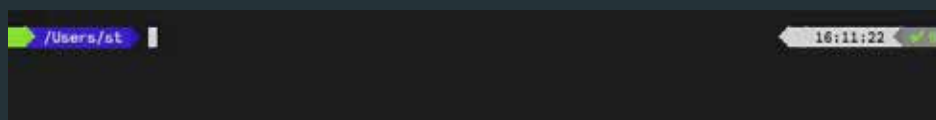
- Container sind zum Wegwerfen
- Sind sehr schnell instanziiert
- Gleicher Kernel wie Hostsystem
- Virtualisierung von
 - Prozessbaum
 - Netzwerkstack
 - Filesystem



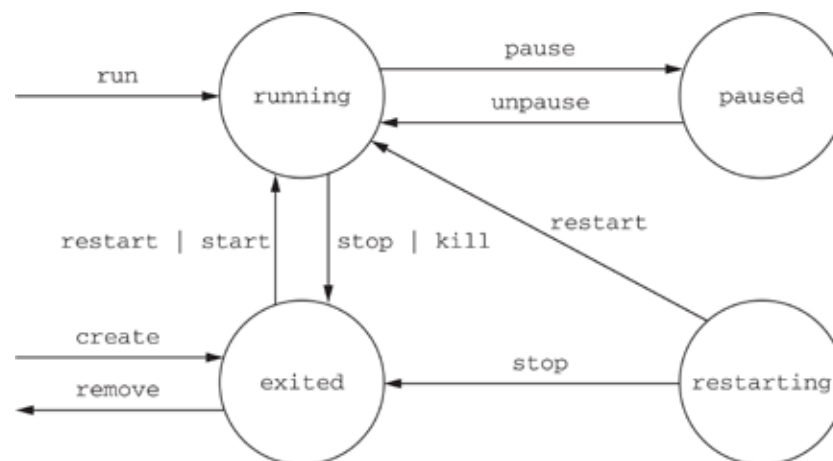
Container == Docker ?!



- Docker-Projekt hat Container populär gemacht
- Linux Container (LXC) als leichtgewichtige Virtualisierung
- Vereinfacht Handling von Containern



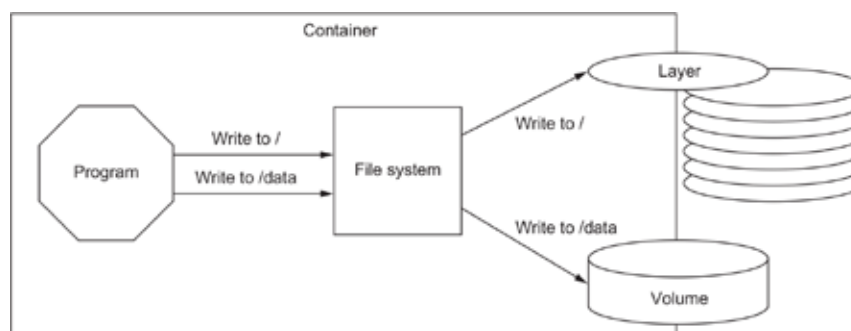
Container States



Aus "Docker in Action", Jeff Nickoloff, Manning

11

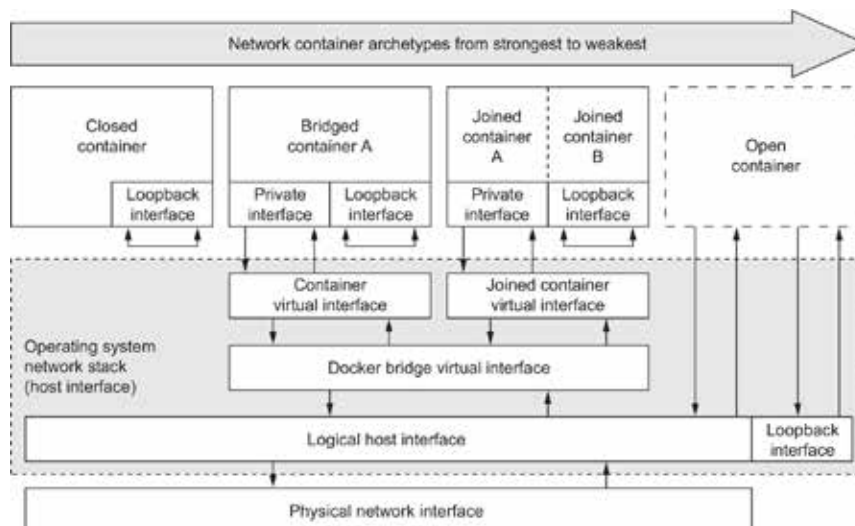
UFS und Volumes



Aus "Docker in Action", Jeff Nickoloff, Manning

12

Docker Networking

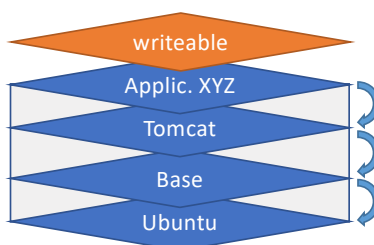


Aus "Docker in Action", Jeff Nickoloff, Manning

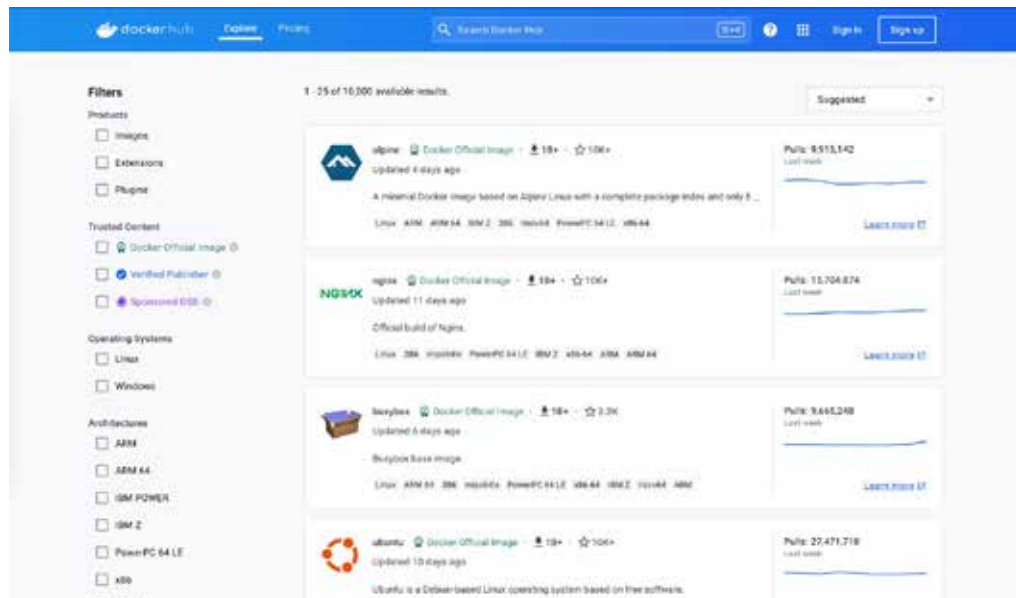
13

Docker: Union Filesystem

- Zeitdauer zum Starten: Nahezu 0
- Overlay/Union-Filesystem
- Schreibzugriffe werden in neuen Layer gelenkt



Docker Hub



Wo läuft docker?

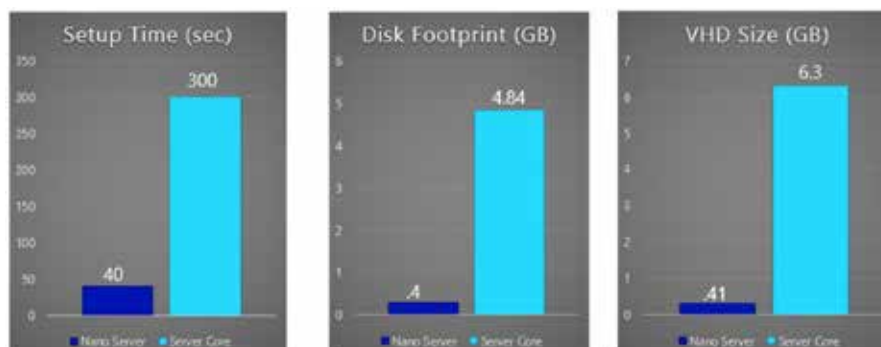
- Docker Container laufen überall
 - Linux: Direkter Kernel-Support
 - Windows: Virtuelle Maschine als Basis
 - Mac OS: Angepaßte Version
- Für Produktivbetrieb: Auf Docker spezialisierte Distros
 - Weniger installierte Software
 - Kleinerer Footprint
 - Bessere Isolation
 - Integrierte Management-Tools
- Public Clouds: In virtueller Maschine
 - AWS, Azure, ...



- Kein Paketmanager
- Nur Docker
- Wenige Userland-Tools
- Hostsysteme werden zentral aktualisiert
 - Aktualisierungsquote festlegbar
 - Fallback-System auf 2. Partition installieren
 - Automatischer Reboot

Microsoft nano server

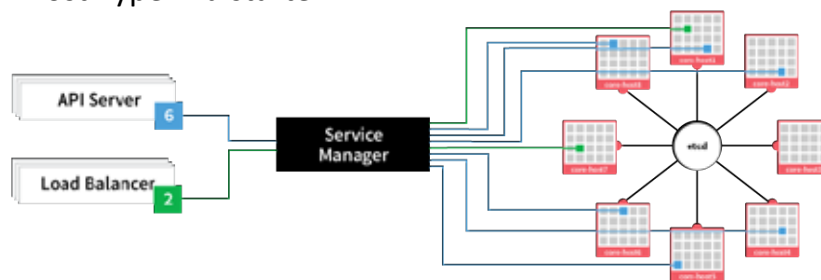
- Abgespeckter Server Core
 - Kein MSI Installer, kein WOW64 (= 32 Bit Support)
 - Keinerlei Nutzer-Interfaces (auch nicht lokales CLI)
 - Konfiguration über PowerShell bzw. Desired State Configuration (DSC)



CoreoS: Fleet



- Verteilt Container auf laufende Hosts
- Hält immer festgelegte Anzahl Instanzen am Laufen
- Startet bei Bedarf zusätzliche
- Tagging (z.B. app, db) erlaubt Container auf bestimmten Host-Typen zu starten



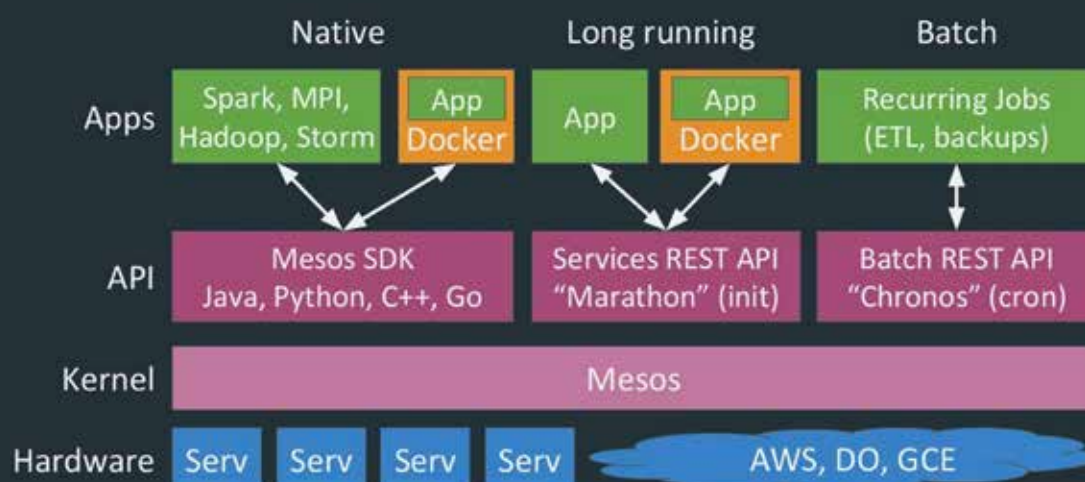
- Management und Orchestrierung von Containern
- Pods fassen mehrere Container zusammen
- Policies legen Eigenschaften der Dienste fest
 - Basisimages, Versionen, Ports, Volumes

Apache Mesos



- 2009 an der UC Berkeley erschaffen
- Abstraktion der Data Center Ressourcen
 - Datacenter Operating System
 - APIs für zahlreiche Sprachen
 - High Availability
- Docker-basierte Scheduler neben anderen Frameworks
- Nutzer: Twitter, Airbnb, Apple's Siri

Mesos = Data Center OS Kernel



Docker Swarm

- Integraler Bestandteil von Docker
- Zugang über Standard Docker API
 - Unterstützung vorhandener Tools
- { Docker Hosts } = Virtueller Docker Host
 - Node API

