



Video Tutorial Docker

von Timo, Elisabeth, Sara

Gliederung

1. Was ist Docker?
2. Docker in Verteilten Systemen
3. Docker Beispiel

Was ist Docker?



- Open-Source-Plattform
- Container-Management System
- Von der Firma Docker Inc.
- Populär ab 2015



Containerisierung

„Die Software wird dazu zusammen mit all Ihren erforderlichen Komponenten, Abhängigkeiten, Informationen über Laufzeitumgebungen und Systemwerkzeugen als Softwarebündel verpackt. Dieses ausführbare Bündel wird Container genannt. [...] So kann ein Container auf einer Windows, Linux oder MacOS Maschine sowie in der Cloud oder sogar in einem weiteren Container laufen“ [Manuel Lucas Hesse 2022, S. 11]

- Container beinhalten eine Software bzw. Programm inklusive der dazu benötigten Abhängigkeiten Bibliotheken etc.
- Methode um Software in einer isolierten Umgebung auszuführen
- Wird direkt auf dem Hostbetriebssystem ausgeführt und nutzt dessen Kernel
- Container ist Plattformunabhängig
- Container einfach und schnell startbar

Andere Containerisierungs-Anbieter

- LXC (Linux) ohne zentralen Daemon
- Hyper-V jeder Container kann über eigenen Kernel verfügen
- Podman ähnlicher Aufbau zu Docker aber auch ohne zentralen Daemon
- rkt mehrere Container können als eine einzige Entität ausgeführt werden = Pod

Docker Entstehung & Problemstellung

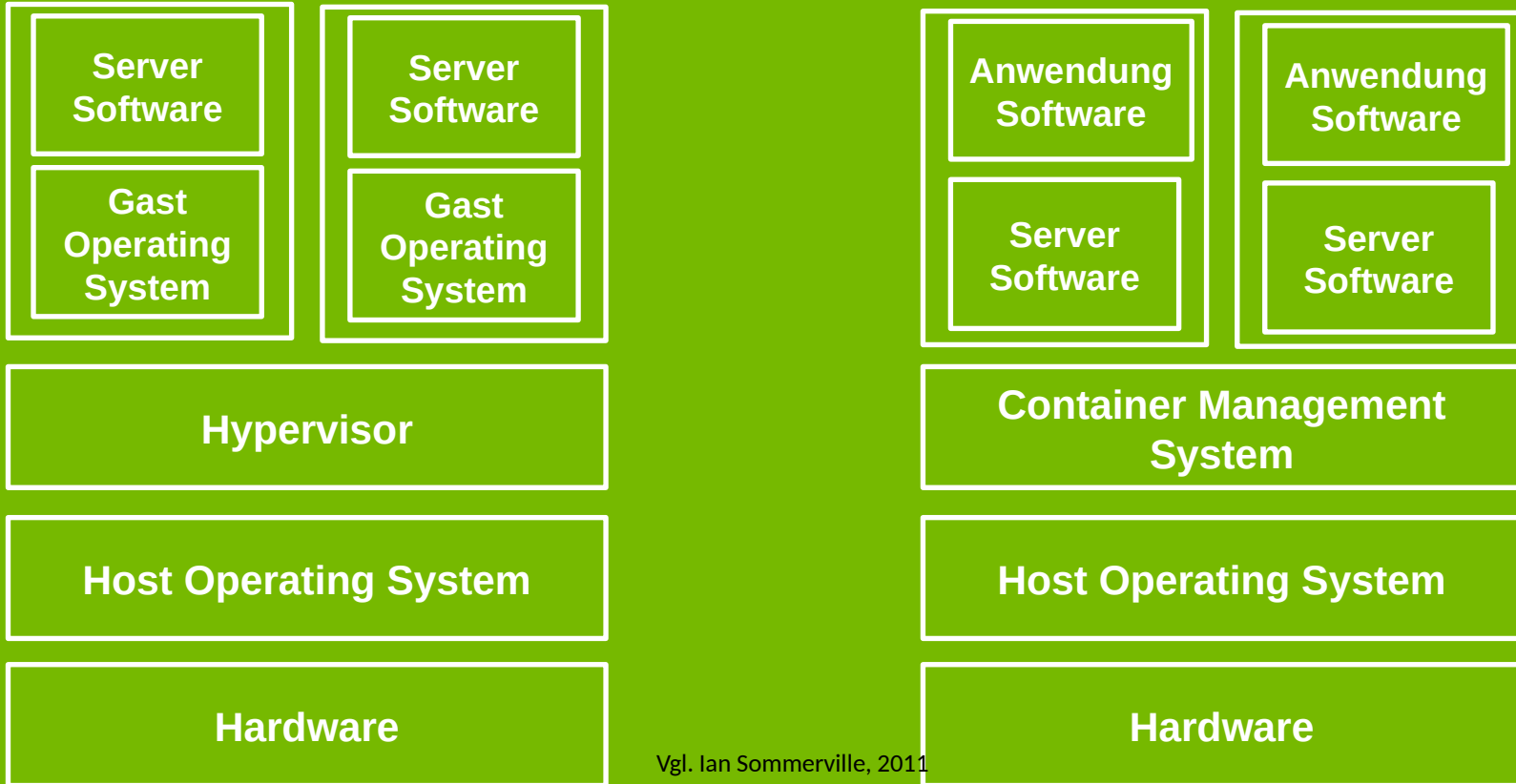
Problemstellung:

- Software schwer auf anderen Computern zu installieren
- Zusätzliche manuelle Installation von Bibliotheken etc. notwendig
- Unterschiedliche Betriebssysteme führen zu Problemen bei der Kompatibilität

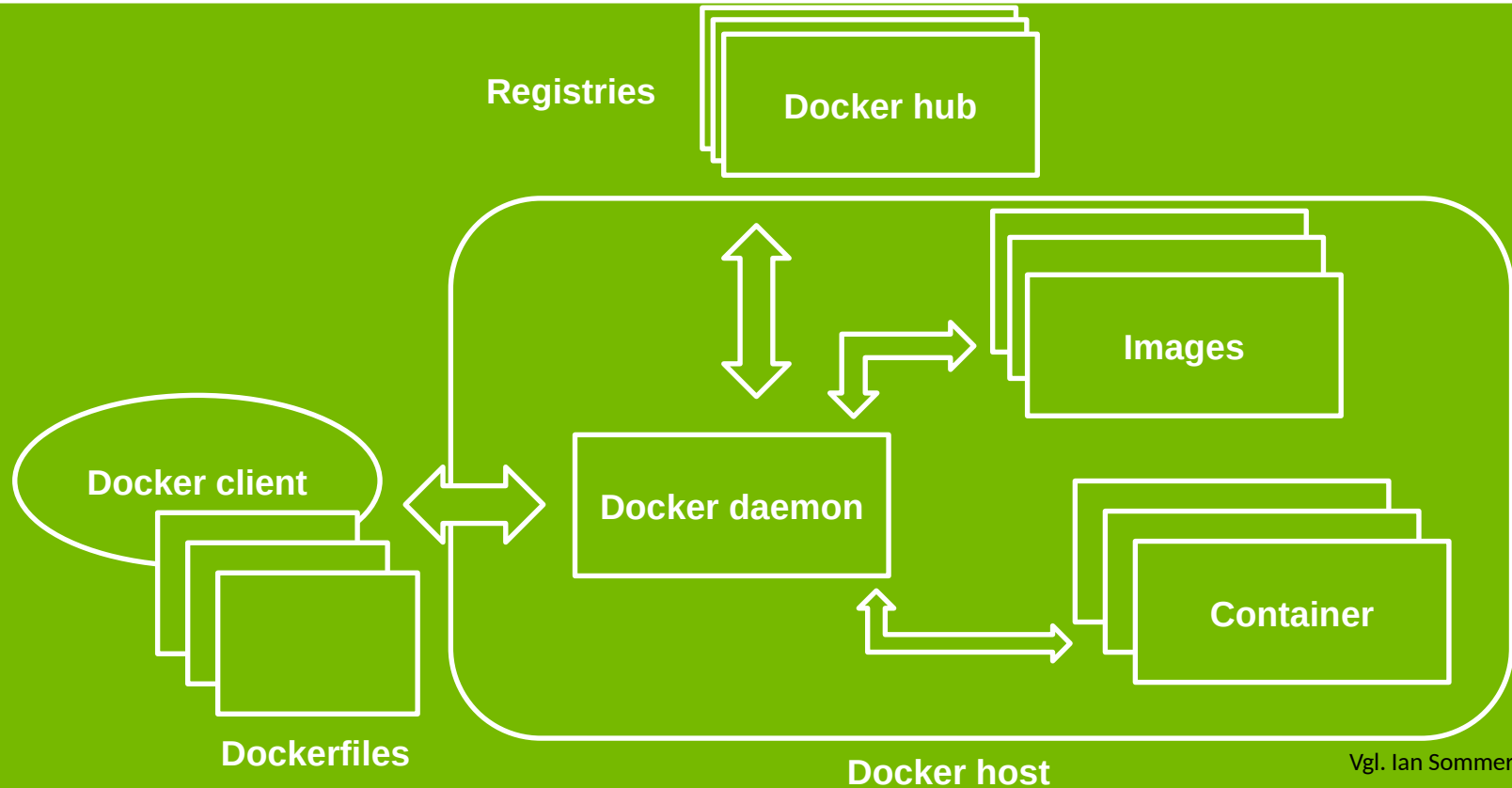
Lösung durch Docker:

- Docker verpackt alle benötigten Bibliotheken etc. in einen Container zur Auslieferung
- Container läuft auf jedem Computer
- Keine Abhängigkeiten der Software von Bibliotheken etc. des Hosts

Unterschied zur Virtuellen Maschine



Bestandteile von Docker



Docker Hub

- Online-Register in welchem Images gespeichert und geteilt werden
- Kann private und öffentliche Images enthalten
- Images können auch anonym hochgeladen werden -> Gefahr der Malware
- Zum Beispiel werden fertige Images für gängige Anwendung (z.B. MySQL, Nginx Webserver) zur Verfügung gestellt
- Spart Zeit und Mühe, da die Images nicht neu erstellt werden müssen

Container-Technologien in Verteilten Systemen

htw.

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Herausforderungen Verteilter Systeme

- **Heterogenität**
 - Knoten haben unterschiedliche Betriebssysteme, Speicher, Uhr, ...
- **Abhängigkeiten** (Dependencies)
 - Anwendungen können bestimmte Bibliotheken oder Frameworks benötigen
- **Zuverlässigkeit/Fehlertoleranz**
 - Verteilte Systeme müssen Fehlern gegenüber robust sein
- **Skalierbarkeit**
 - Weitere Instanzen müssen hinzugefügt werden können
- ...

Wie kann man Container-Technologien nutzen, um mit diesen Herausforderungen besser umzugehen?

Container in Verteilten Systemen

Vorteile:

- Portabilität: Anwendungen laufen unabhängig vom Betriebssystem
- Isolierung: Vermeidet Konflikte
- Konsistenz: Anwendung funktioniert auf allen Systemen gleich
- Abhängigkeiten: Werden bereitgestellt
- Fehlertoleranz: Fehlgeschlagene Anwendungen automatisch neu starten (z.Bsp. Mit Kubernetes)
- Skalierbarkeit: Es können viele Container gleichzeitig verwaltet werden (z.Bsp. Kubernetes, Docker compose)

Anwendungszwecke - Beispiel

Beispiel: Automatisiertes Unit Testing

- Automatisierter Test braucht mehrere Stunden und beansprucht mehrere Geräte

Ohne Docker:

- Der Test wird in Teile aufgeteilt
- Es wird gewartet, bis genügend Geräte verfügbar sind
- Test läuft auf den Geräten
- Lange Wartezeiten
- Problem: Dauert zu lange, schlechte Nutzung von Zeit/Ressourcen

Anwendungszwecke - Beispiel

Mit Docker:

- Der Test wird in mehr Teile geteilt, die alle in einem eigenen Container laufen
- Mehrere Container laufen gleichzeitig auf dem gleichen Gerät

Ergebnis:

- Kürzere Dauer
- Weniger Geräte und Ressourcen genutzt
- Mehrere Tests gleichzeitig
- Keine Wartezeiten, wenn andere Tests schon laufen



VIELEN DANK FÜR EURE AUFMERKSAMKEIT! :)

www.htw-berlin.de

Quellen

- **Containerisierung einer Webanwendung, automatische Integration und Bereitstellung ihrer Softwareänderung von Manuel Lucas Hesse, aufgerufen am 05.01.2025**
https://monami.hs-mittweida.de/frontdoor/deliver/index/docId/14547/file/BA_52260_Manuel_Lucas-Hesse_geschwaerzt.pdf
- **Softwareengineering, 9.Auflage von Ian Sommerville**