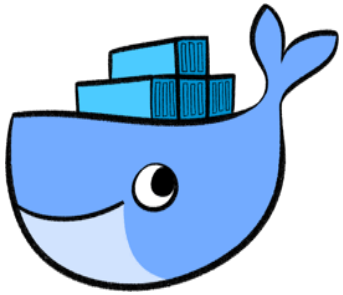
https://doi.org/10.20378/irb-53685

University of Bamberg

Running DSPACE, DSpace-CRIS with Docker



University of Bamberg

Universitätsbibliothek

steffen.illig@uni-bamberg.de

Rechenzentrum

andreas.eiermann@uni-bamberg.de

Wir wollen mit DSpace-CRIS testen & haben Probleme

Viele Bausteine : TomCat, Ant, Maven, Postgres, DSpace
Viele Commands (Bsp. Maven package) und Abhängigkeiten

https://wiki.duraspace.org/display/DSDOC5x/Installing+DSpace

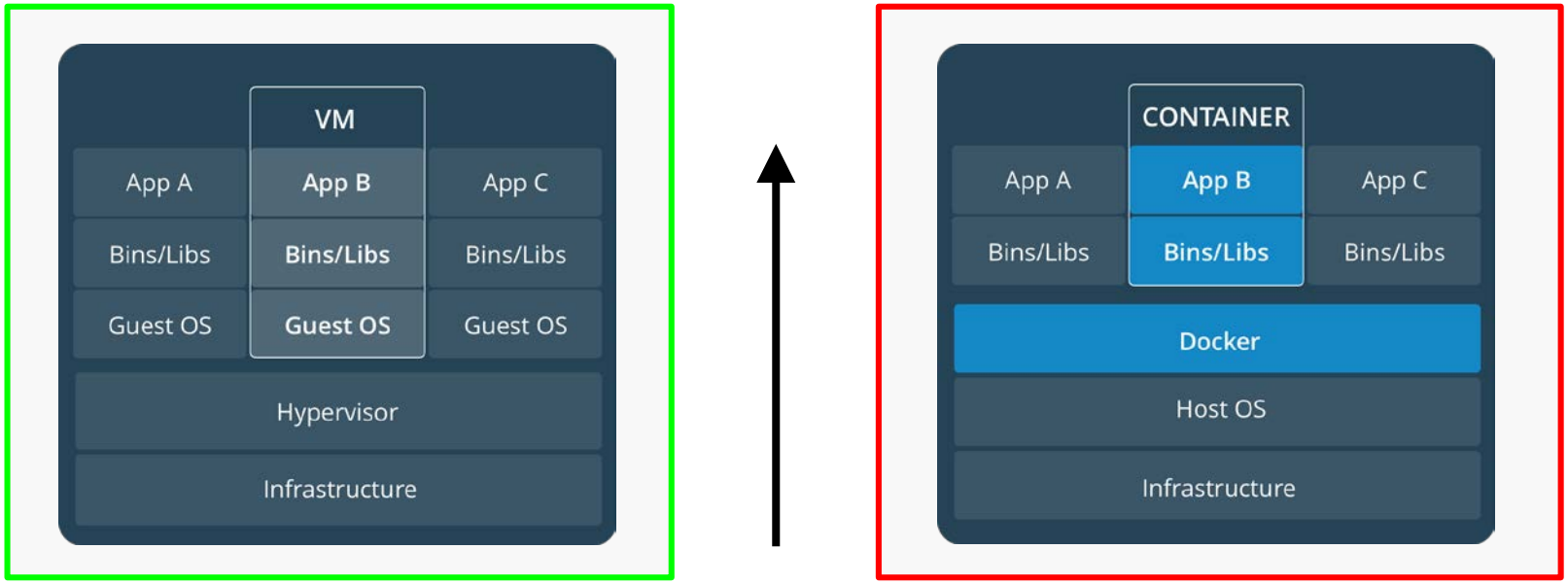
Wie können wir einfach und schnell ein System wie DSPACE oder DSpace-CRIS installieren?

Wie können wir das nachvollziehbar & reproduzierbar tun?

Zunächst: Was ist Docker?

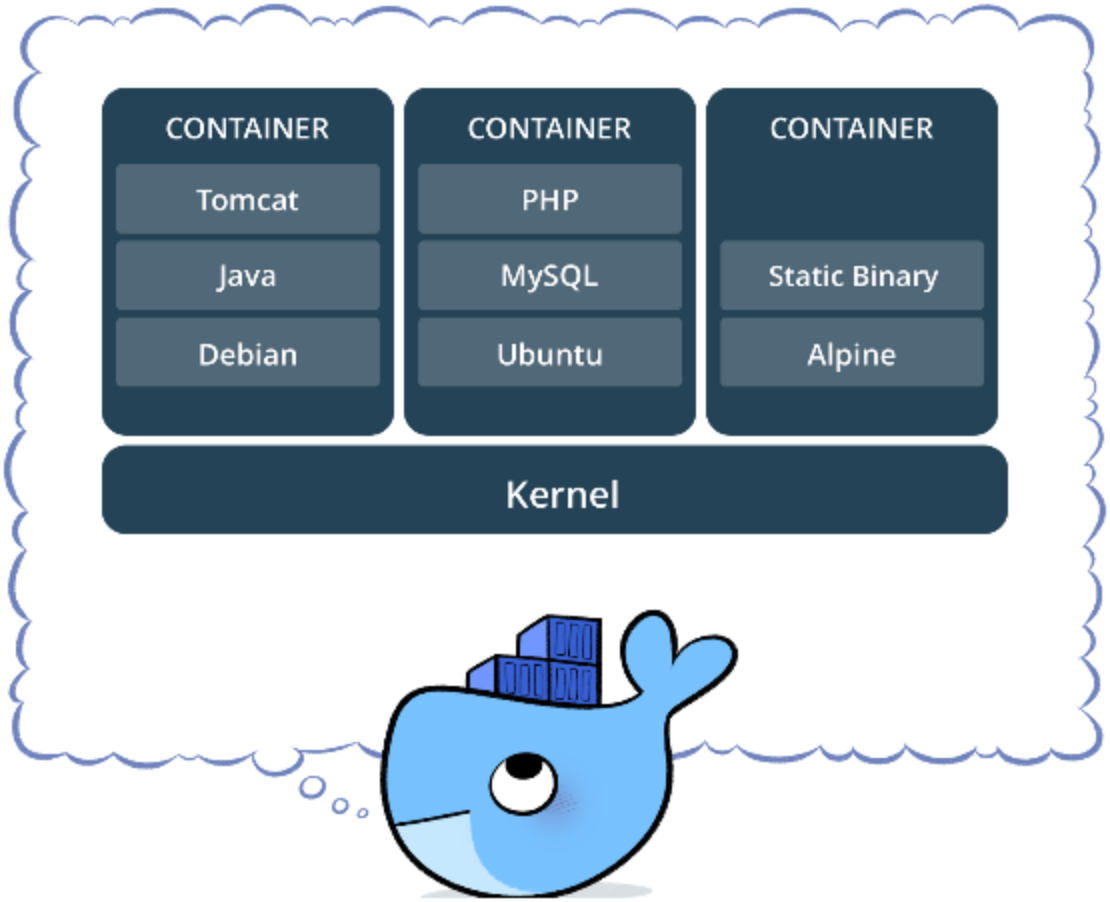
Docker kommt aus der Linux Welt
Nutzt Kernel Funktionalitäten um Prozesse zu isolieren

Vergleich **Virtualisierung** vs **Docker**



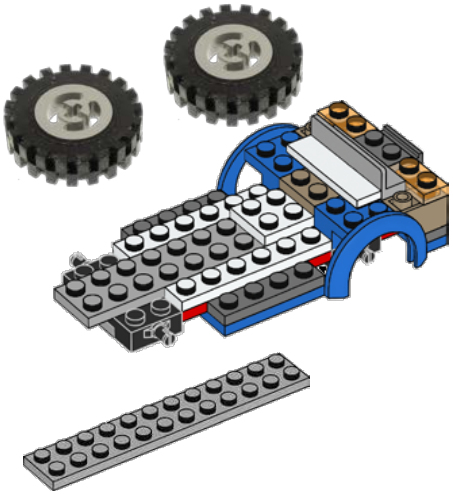
© www.docker.com/what-container

Zunächst: Was ist Docker?

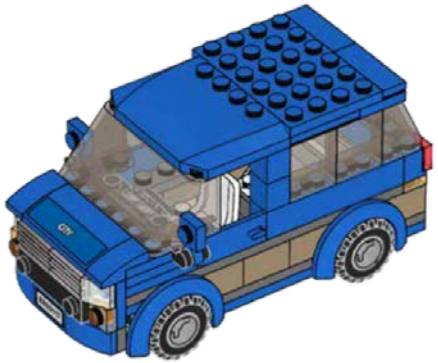


© www.docker.com/what-container

Zunächst: Die wichtigsten Konzepte von Docker



Dockerfile (Bauplan)



Docker Image

docker-compose.yml

© www.lego.com

Lösungsvorschlag: DSpace-CRIS & Docker

Dockerfile

Bauplan für eine DSpace-CRIS Version. Ergebnis ist ein Docker Image

und

docker-compose.yml

definiert Ports (z.B. 8080)
definiert Volumes für Daten
startet Applikation (hier DSpace-CRIS & Postgres Container)

In Kooperation mit *4Science* haben wir die Dateien für Sie auf GitHub zur freien Verfügung eingestellt: https://github.com/4Science/dspace-docker

Lösungsvorschlag: Fazit

Drei Befehle um DSpace-CRIS zum Laufen zu bekommen:

```
git clone https://github.com/4Science/dspace-docker
cd dspace-docker
docker-compose up -d
```

Ja, aber ...

- Image beinhaltet Quellcode und Build-Tools (Maven)
  … ist daher komplex und groß

- “docker-compose up”
  … dauert sehr lange (laden und bauen)

Können wir das nicht noch optimieren?

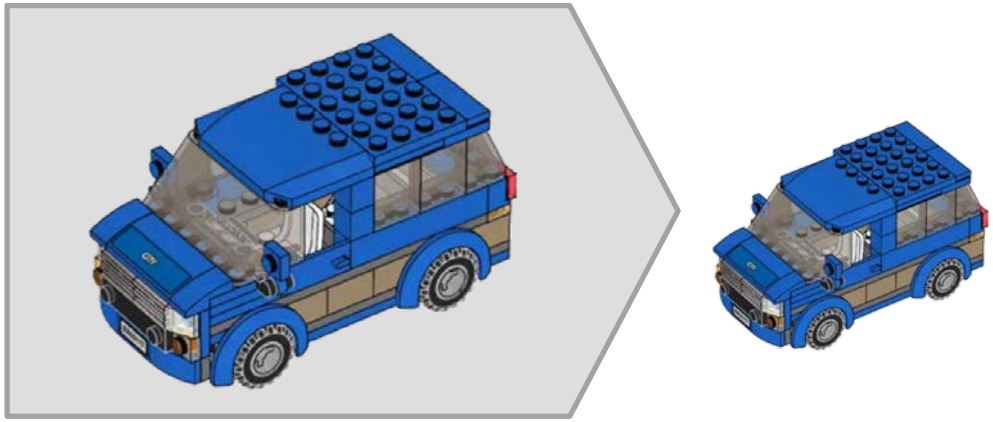
© www.lego.com

Ausblick 1: Imagegröße reduzieren

Bisher ist die Imagegröße ist mit DSpace Quellcode, Maven, Maven Cache, ..
in Summe ca. **7 GB**

Lösungsansatz:

Künftiges Image enthält nur Laufzeitumgebung (Java, Tomcat) & DSpace-CRIS
Dazu Überarbeitung des Dockerfiles (Multistage build)
Künftige Größe nur noch ca. **3 GB**



© www.lego.com

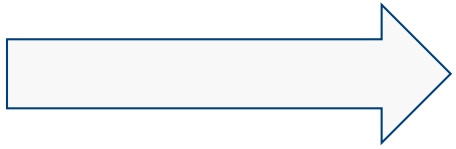
Ausblick 2: Start beschleunigen

Bisher werden mit Hilfe des Dockerfiles die Bestandteile von DSpace-CRIS aus unterschiedlichsten Quellen heruntergeladen & anschließend lokal kompiliert.

Lösungsansatz:

- Docker Image in einer Registry speichern (z.B. DockerHub)
- Nachnutzung des Images spart Bandbreite/Zeit (lokales kompilieren entfällt)

Docker Image



docker-compose.yml

Wunderbar: Noch eine letzte Frage …

Wie können wir die Kooperation im Team gestalten? Mit mehreren internen Entwicklern oder mit externen Firmen wie *4Science* oder *The Library Code*?

Welche Lösung bietet sich hier mit Docker an?

© www.lego.com

Ausblick 3: Entwicklung im Team

CI/CD - Pipeline

Customization DSpace-CRIS

Dockerfile

docker-compose.yml

Docker Image