

docker

Beginner's Guide

fhLUG - 11. Juni 2019

Stefan Schöberl

fhLUG

Who am I?

Beginner's Guide

- Wozu Docker?
- Basics
- Dockerfiles, Registry & Netzwerke
- Docker Compose
- Tipps, Tricks & Nützliches
- Ausblick

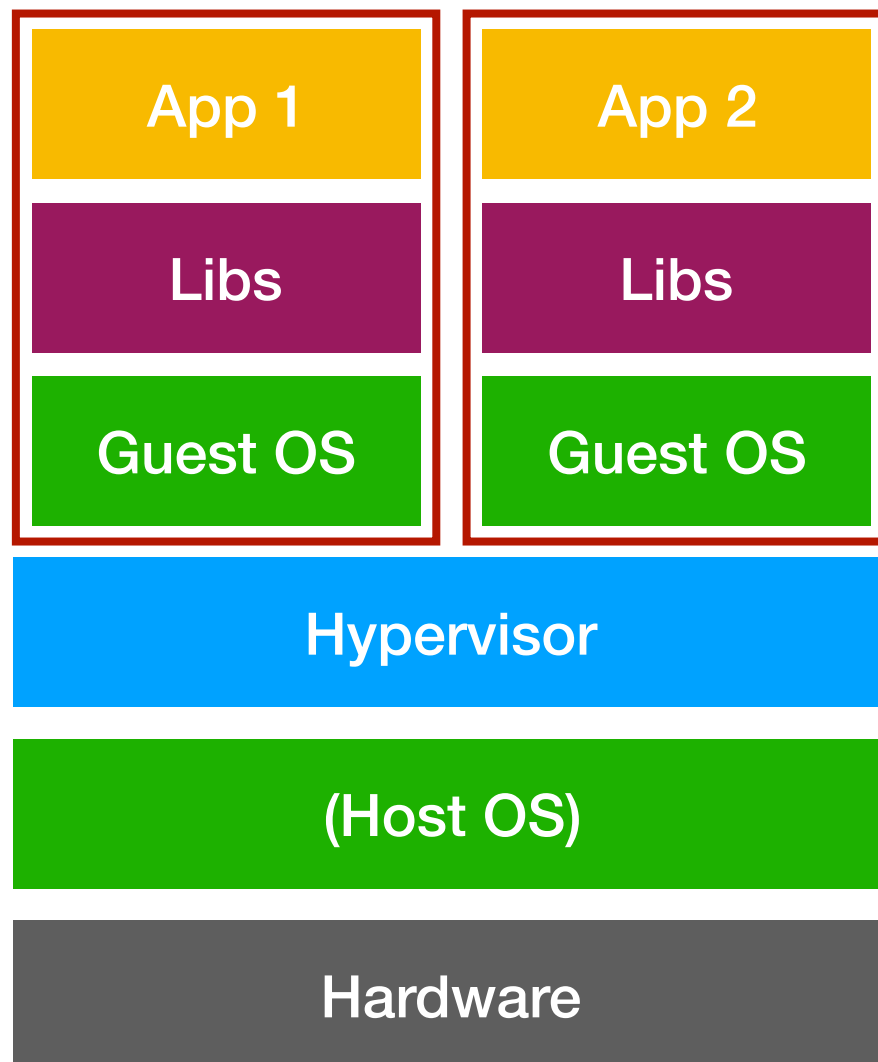
`github.com/stefanschoeberl/docker-beginners-guide`

Wozu Docker?

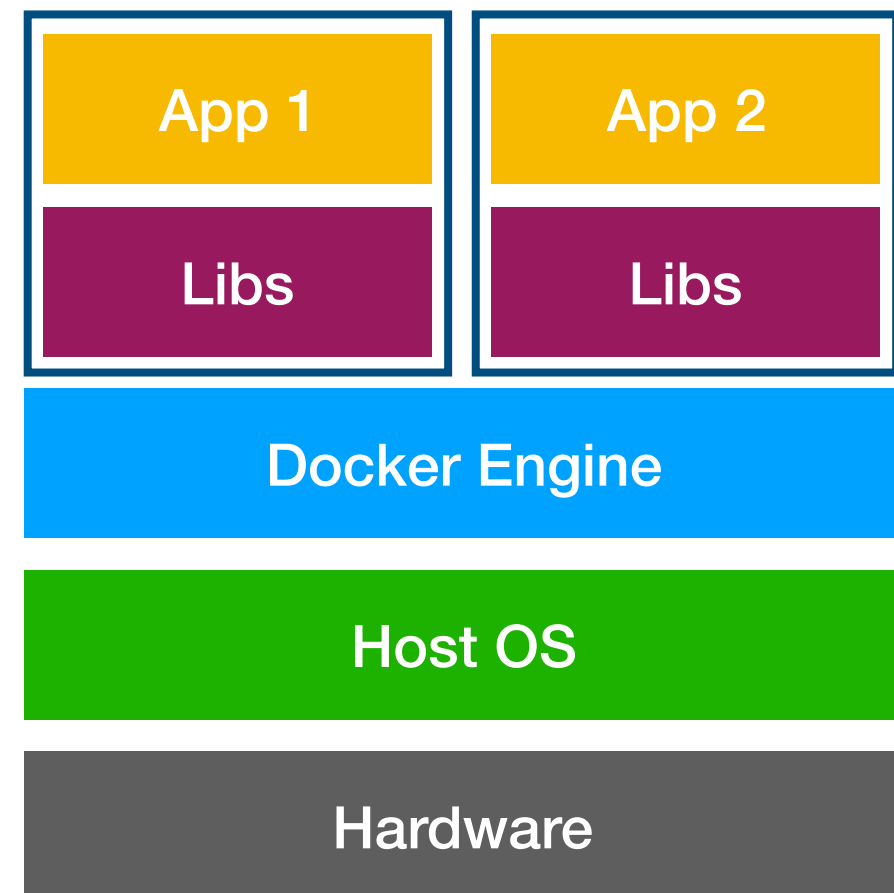
- Schlanke Alternative zu virtuellen Maschinen
- Einfache Verwendung
- „It works on my machine“
- CI/CD

VM vs Docker

VM



Container



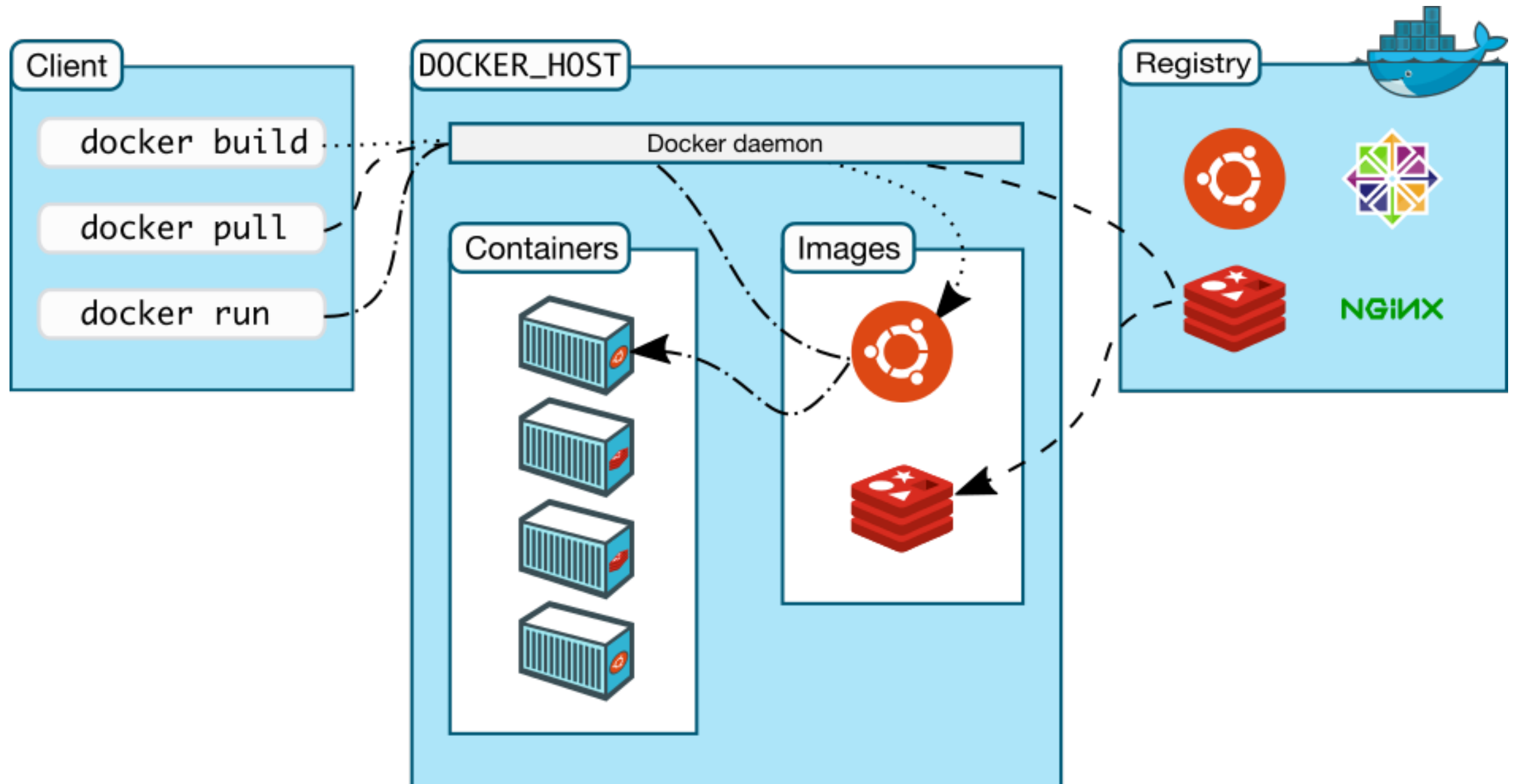
Live-Demo

Hello World

```
$ docker run hello-world
```

github.com/stefanschoeberl/docker-beginners-guide

Docker Architektur



Container

- zustandslos → jederzeit neu erstellen
- Port-Mappings
- Volumes
- Konfiguration (idealerweise) über Umgebungsvariablen

Live-Demo

Container I

```
$ docker container
```

github.com/stefanschoeberl/docker-beginners-guide

Container II

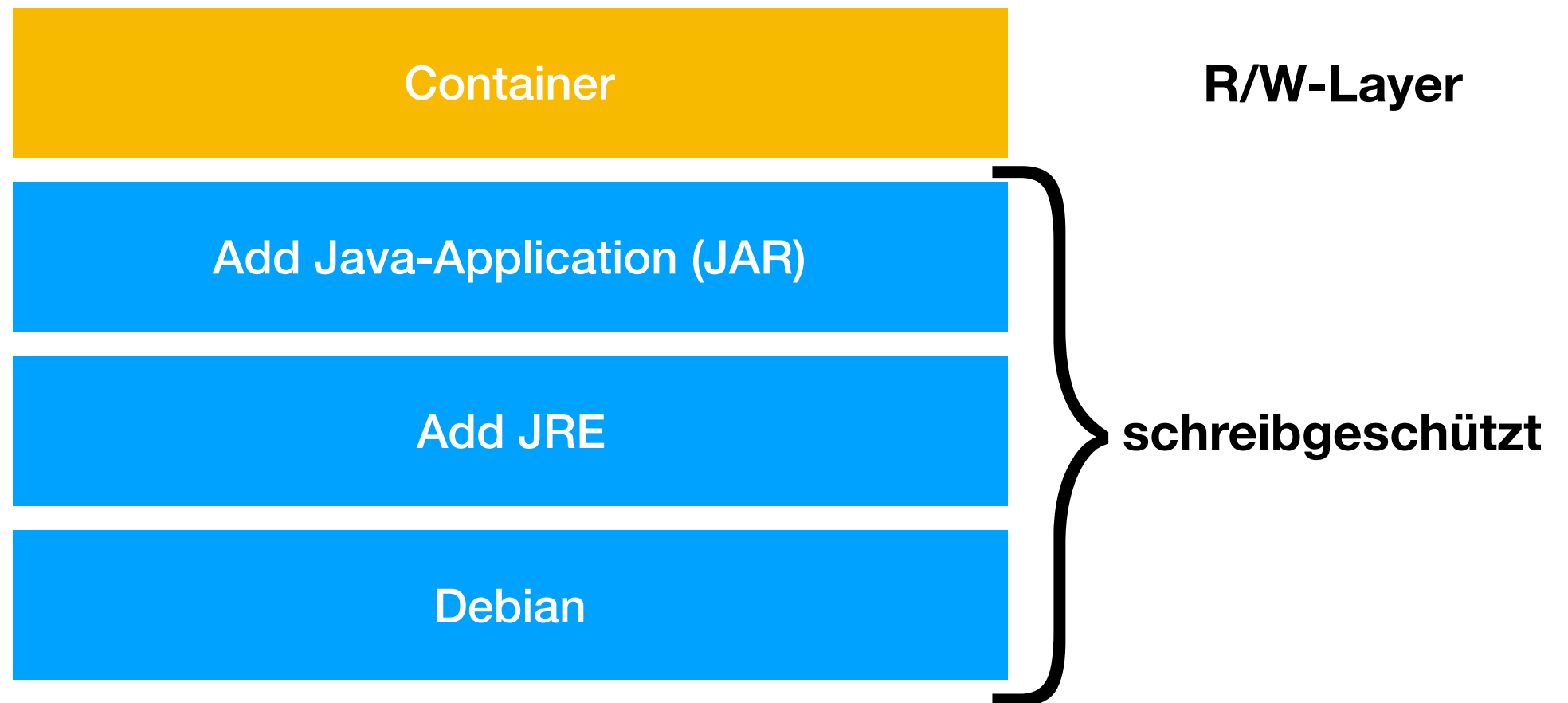
```
$ docker container run  
                                logs  
                                exec
```

github.com/stefanschoeberl/docker-beginners-guide

Image

- Vorlage für einen Container
- schreibgeschützt
- UnionFS
- Layers
- Tags
- Dockerfile

Layers



Images

```
$ docker image ls
```

Registry

- zentraler Speicherort für Images
- Docker Hub als „offizielle Registry“
- Registry als Docker-Image ;) (folgt später...)

Live-Demo

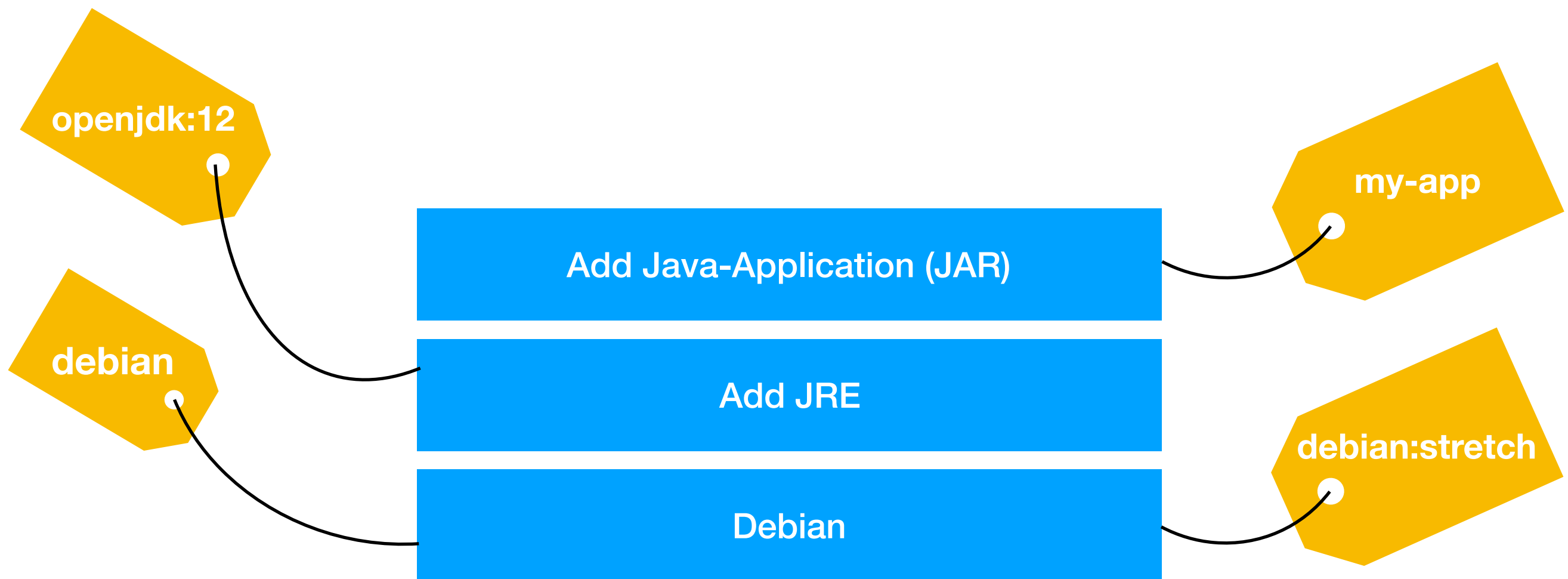
Docker Hub

hub.docker.com

Tags

- Name für Images
- `registry / repository / image : version`
 - `debian:9`
 - `my-registry.com/fhLUG/demo:v1.2.3`
- Default-Registry: Docker Hub

Tags



Ein paar Worte zu Commands...

- früher: `docker [command]`
- jetzt: `docker [command] [subcommand]`
- <https://blog.docker.com/2017/01/whats-new-in-docker-1-13>

Dockerfile

- Bauanleitung für Image
- Startet immer mit einem Basisimage
- RUN, COPY und ADD erzeugen neue Layers

Dockerfile Commands

- FROM <image>
- COPY <src>... <dest> (und ADD)
- RUN <command>
- WORKDIR <path>
- CMD <command>
- ENV <key> <value>
- EXPOSE <port>

Live-Demo

Dockerfile I

```
$ docker image build
```

github.com/stefanschoeberl/docker-beginners-guide

Dockerfile II

```
$ docker image build
```

github.com/stefanschoeberl/docker-beginners-guide

Best practices

- „ephemeral“
- eine Aufgabe pro Container
- Anzahl an Layers reduzieren
- keine unnötigen Sachen installieren
- .dockerignore
- `apt-get update && apt-get install` in einer Zeile
- Build-Cache nutzen
- Multi-Stage
- Formatierung

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/

Live-Demo

Lokale Registry

```
$ docker image tag  
push  
pull
```

github.com/stefanschoeberl/docker-beginners-guide

Netzwerke

- Kommunikation zwischen Containern
- Container können Netzwerk beitreten
- Container finden andere Container über Name

Live-Demo

Netzwerke

```
$ docker network
```

github.com/stefanschoeberl/docker-beginners-guide

Docker Compose

- lange Run-Befehle
- manuelles Aufsetzen von Containern und Netzwerk
→ aufwendig bei vielen Containern
- eine Möglichkeit: Shell-Skripte
- eleganter: docker-compose.yml-File

docker-compose.yml

- YAML
- deklarative Definition von Services (=Containern)
 - Image
 - Ports
 - Volumes
 - Umgebungsvariablen
- Alle Container automatisch in einem Netzwerk

Live-Demo

Docker Compose

```
$ docker-compose up
```

github.com/stefanschoeberl/docker-beginners-guide

Tipps, Tricks und Nützliches

- Programme/Server einfach und unkompliziert testen
- ~~„It works on my machine“~~ → „It works on **any** machine“
- lokale Datenbank zum Entwickeln
- traefik
- GitHub CI
- IDE Support

Live-Demo

Lokale Datenbank

```
$ docker-compose up
```

github.com/stefanschoeberl/docker-beginners-guide

Wo sind meine GB hin??

- Container löschen
- Images löschen
- Docker Desktop Reset

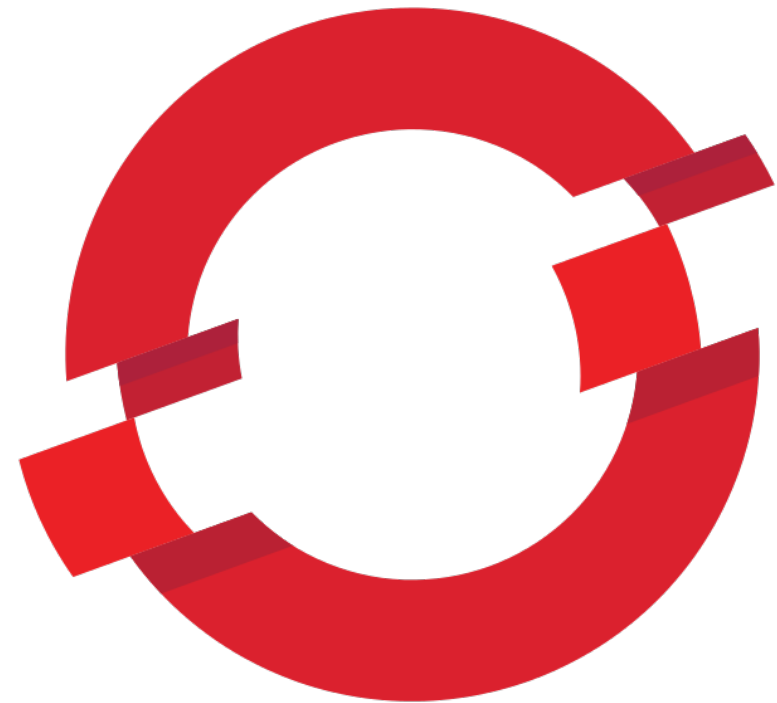
Cleanup

```
$ docker ... [rm|prune]
```

... und wie geht's weiter?



kubernetes



OPENSIFT

Fragen?

Bonus

Multi-Stage-Dockerfile

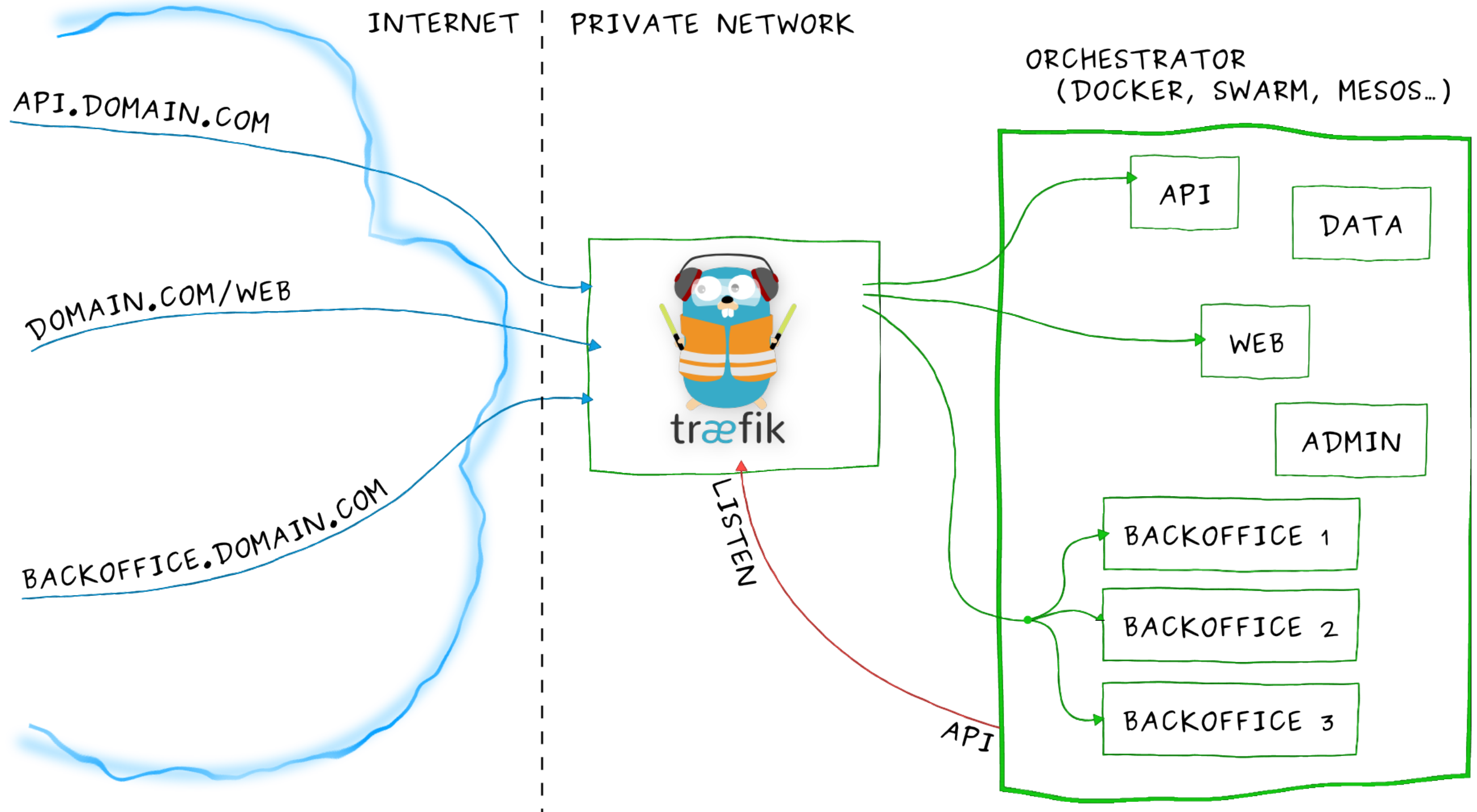
- Probleme mit einfachen Dockerfiles
 - Container-Größe reduzieren umständlich
 - Sourcecode und Buildtools nach dem Kompilieren nicht mehr benötigt
- Lösung: Multi-Stage-Dockerfile
 - Kompilieren erfolgt in einem Container (Stufe 1)
 - Ausführen erfolgt in einem anderen (Stufe 2)

Live-Demo

Multi-Stage-Dockerfile

`github.com/stefanschoeberl/docker-beginners-guide`

traefik



Live-Demo

traefik