

DOCKER - Eine Einführung

...aus Sicht eines Softwareentwicklers



(<http://www.nautsch.com/>)

Oliver Nautsch

- Inventage AG
- ab 2015 selbstständig

Software Craftsman

- Engineer, Coach, Architect, Trainer, Team Lead, Consultant, Gardener, ...
- Microservices, Open Source, TDD, DDD, CD, Virtualisierung, ...

email: oliver.nautsch@gmail.com (<mailto:oliver.nautsch@gmail.com>)

twitter: [@ollispieps](https://twitter.com/ollispieps) (<https://twitter.com/ollispieps>)

github: [@ollin](https://github.com/ollin) (<https://github.com/ollin>)

google+: [OliverNautsch](https://plus.google.com/+OliverNautsch/) (<https://plus.google.com/+OliverNautsch/>)

Familie, 2 Kinder

Jugs Board Member seit 2014

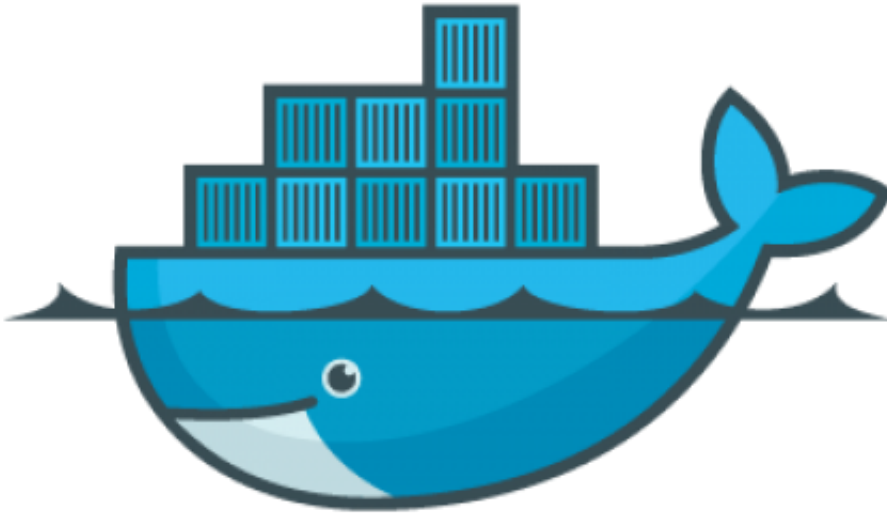
OSS an Schulen (<http://www.ossanschulen.ch/>), Hackergarten Zürich

(<http://www.meetup.com/Hackergarten-Zurich/>), Coders Only Zürich

(<http://zurich.codersonly.org/>)

. . . running . . .

Was ist Docker

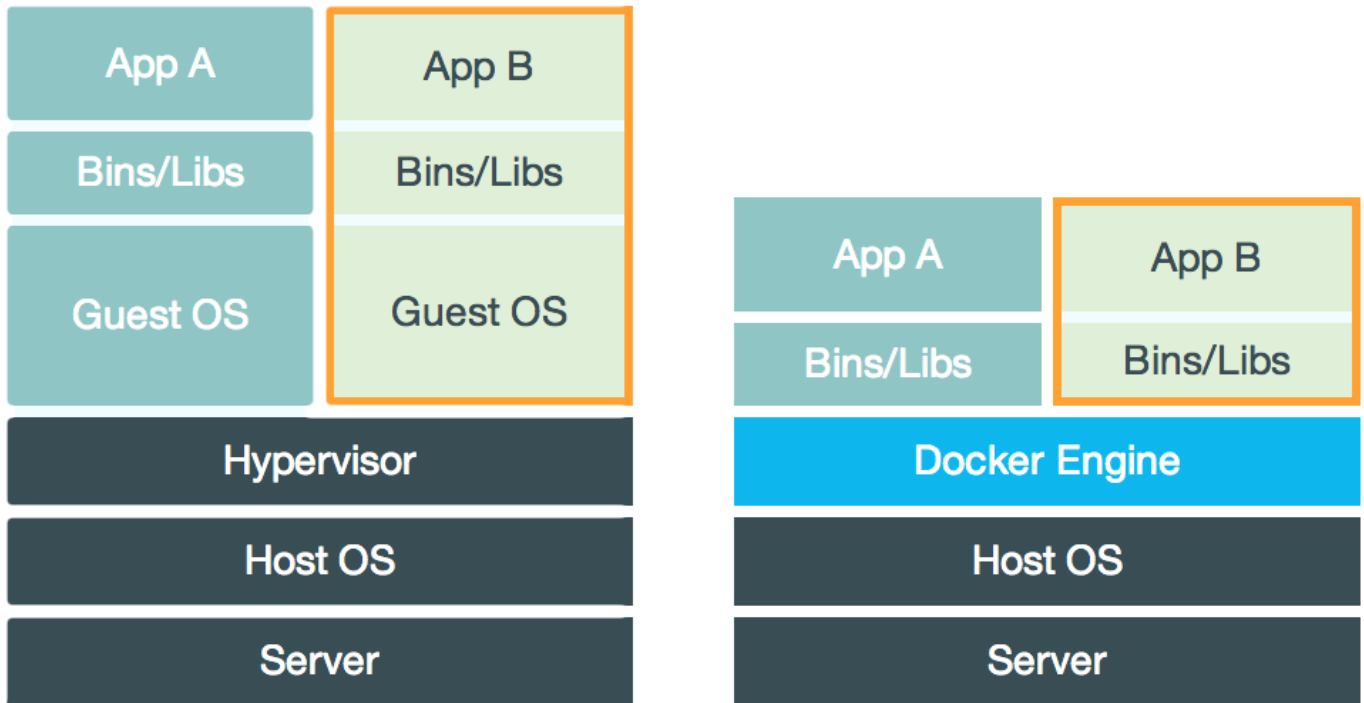


docker

(<http://www.docker.com/>)

- Open Source Software
- Applikationen in Software Containern
- OS-Level Virtualisierung auf Linux

Unterschied zu virtuellen Maschinen



- Beispiel: virtualbox via Vagrant (<https://www.vagrantup.com/>) vs docker

Anwendungsfälle

- Entwicklungs- und Buildumgebungen
- Continuous Integration / Continuous Delivery
- Produktionsumgebungen
- Mandantenfähigkeit
- Server Konsolidierung
- Applikationsisolierung
- einfachere Konfiguration (immutable infrastructure)
- ...

Docker Elemente



Andrew Clay Shafer
@littleidea



Follow

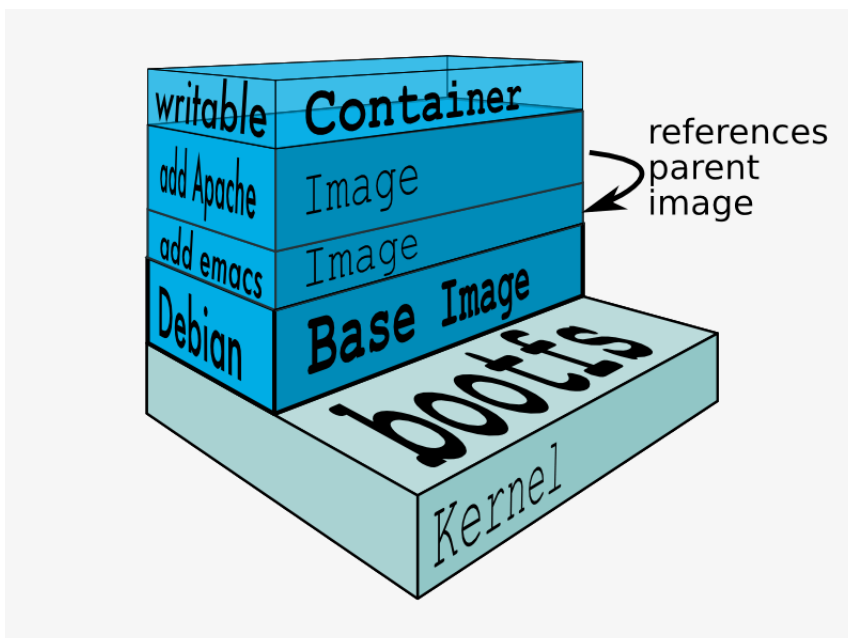
The Docker story:
'usability > functionality'

whoever has ears to hear, let them hear

↩ Reply ↻ Retweeted ★ Favorite ⋮ More

- Docker Hub (<https://hub.docker.com/>) ist eine Registry für Images
- Docker Engine
 - Docker Daemon (REST API (https://docs.docker.com/reference/api/docker_remote_api/))
 - Docker CLI
- Image
 - Layers (wie git commits or changesets für ein Dateisystem)
 - ein Parent
 - r
- Container
 - rw via Union File System

Docker Techniken



- cgroups - control groups - Linux Kernel Funktion zum begrenzen, messen und isolieren von Ressourcen (CPU, Speicher, disc I/O etc.)
- kernel namespaces - trennen der Sichtbarkeit von Ressourcen in anderen Gruppen (Prozesse, Netzwerk, user IDs, Dateisysteme, etc.)
- Overlay-Dateisystem (AuFS oder Btrfs)

Running a Container

BASH

```
sudo docker run --rm -i -t ubuntu:14.04.1 bash
root@35d2e9236656:/# exit
sudo docker ps -a
```

- *docker* - das CLI
- *run* - erzeugt einen Container und startet ihn
- *--rm* - entfernt den Container wieder, wenn er beendet wird
- *-i* - interactive / STDIN bleibt offen
- *-t* - Terminal
- *ubuntu:14.04.1* - <repo_name>[:<tag>] , repo_name - [<location>/][<user>/]
<name>
 - Beispiele:
 - myrepo.nautsch.ch:12345/nautsch/myimage:1.0
 - ubuntu
- *bash* - der auszuführende Befehl

Erzeugen eines Images - via commit

BASH

```
sudo docker run -t -i ubuntu:14.04.1 bash
root@dcde95ca3e5c:/# touch huschihops.txt
root@dcde95ca3e5c:/# exit
sudo docker ps -a
sudo docker commit -m="added huschihops.txt" -a="Oliver Nautsch" dcde95ca3e5c
ollin/huschihops:1.0
```

. . . - via Dockerfile

```
FROM ubuntu:14.04.1
MAINTAINER Oliver Nautsch <oliver.nautsch@gmail.com>

RUN touch huschihops.txt
```

TXT

```
sudo docker build -t ollin/huschihops:1.0 .
Sending build context to Docker daemon 2.56 kB
Sending build context to Docker daemon
Step 0 : FROM ubuntu:14.04.1
---> c4ff7513909d
Step 1 : MAINTAINER Oliver Nautsch <oliver.nautsch@gmail.com>
---> Using cache
---> 98658a58fa03
Step 2 : RUN touch huschihops.txt
---> Running in 2d4d554cbf2a
---> 01ee073272ec
Removing intermediate container 2d4d554cbf2a
Successfully built 01ee073272ec
```

BASH

```
sudo docker images
REPOSITORY          TAG          IMAGE ID          31
CREATED            VIRTUAL SIZE
ollin/huschihops    1.0          01ee073272ec
seconds ago        225.4 MB
. . .
```

Image - Container

```
docker inspect jenkins
. . . (Metainformationen ueber Image als JSON, u.a. ExposedPorts)

docker create -t -p 8080:8080 --name myjenkins1 jenkins # erzeuge job 'aaaa'
docker ps
docker start myjenkins1
docker ps
docker inspect myjenkins1
docker run -d -t -p 8081:8080 --name myjenkins2 jenkins # erzeuge job 'bbbb'
docker ps

docker rm -f myjenkins2 myjenkins1
```

BASH

- erzeugen vs. starten

- Ports von Container an Host herausgeben
- `docker rm` und die Daten sind weg!

Data Volumes

- spezielle Verzeichnisse in einen oder mehreren Containern
- geht am Union File System vorbei
- Änderungen werden nicht berücksichtigt, wenn neues Image erzeugt wird
- persistent bis kein Container sie mehr benutzt

Verzeichnis von Host mounten

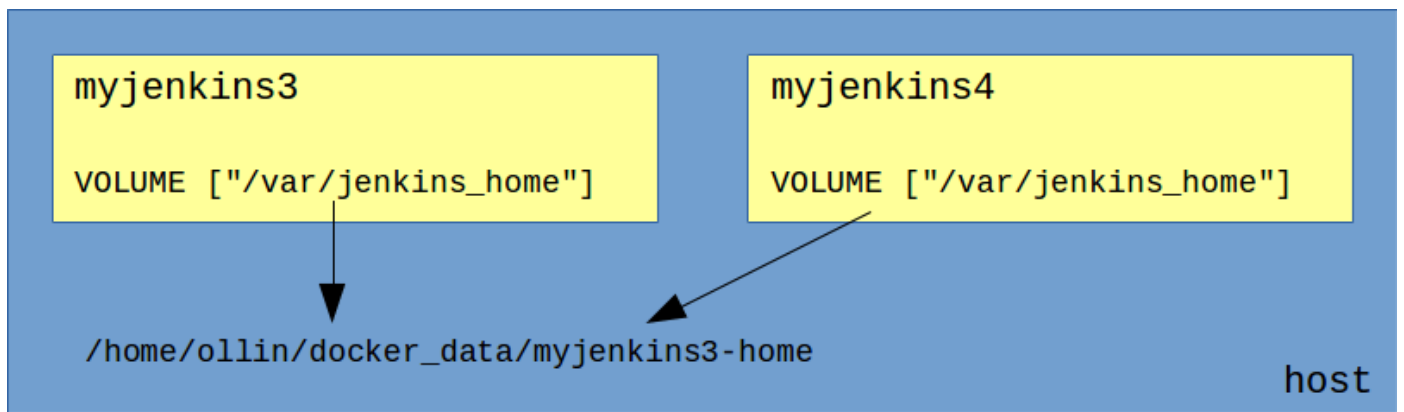
BASH

```
docker run -d -t --name myjenkins3 -p 8083:8080 -v
/home/ollin/docker_data/myjenkins3-home:/var/jenkins_home jenkins:latest
```

```
ls -al /home/ollin/docker_data/myjenkins3-home
```

```
docker stop myjenkins3 && docker rm myjenkins3
```

```
docker run -d -t --name myjenkins4 -p 8084:8080 -v
/home/ollin/docker_data/myjenkins3-home:/var/jenkins_home jenkins:weekly
```

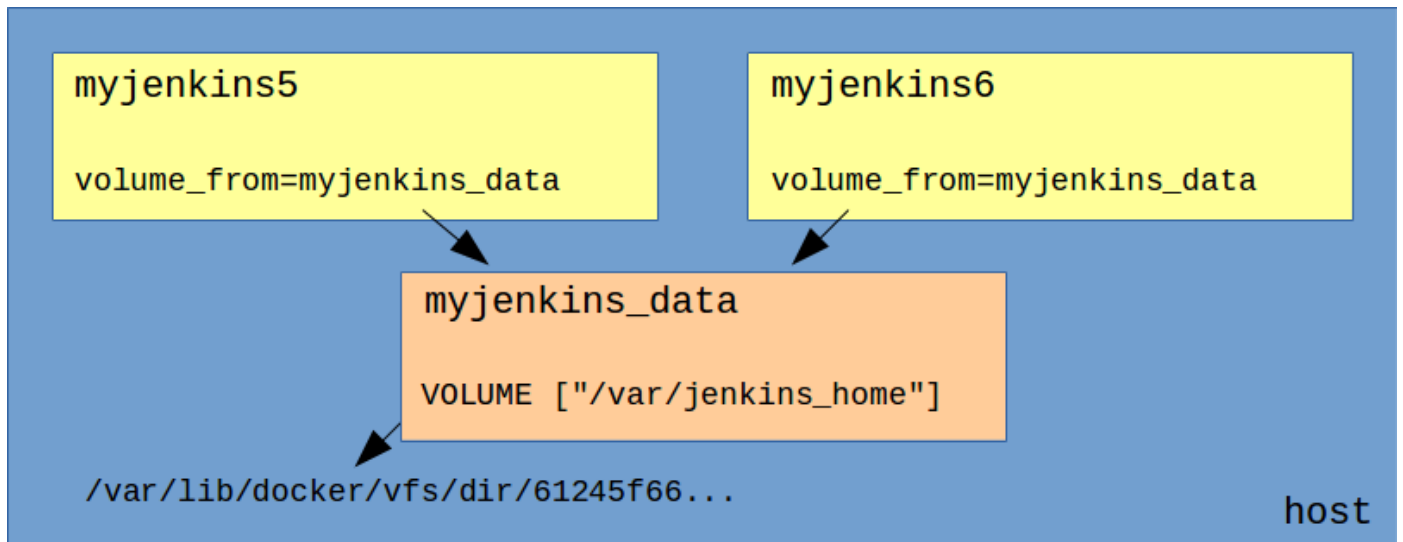


Data Container

```
docker run --name myjenkins_data -v /var/jenkins_home tianon/true
docker run -d -t --name myjenkins5 -p 8085:8080 -u root --volumes-
from="myjenkins_data" jenkins:latest
```

```
docker rm -f myjenkins5
```

```
docker run -d -t --name myjenkins6 -p 8086:8080 -u root --volumes-
from="myjenkins_data" jenkins:weekly
```



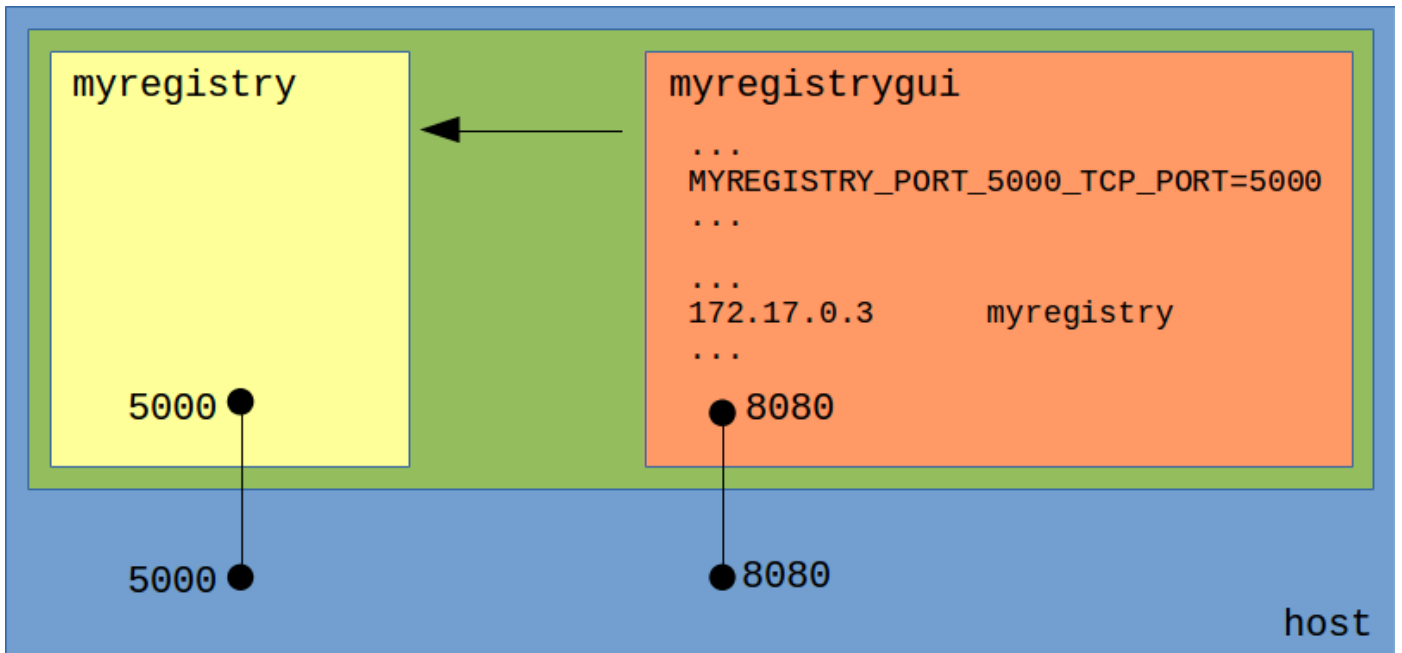
- Container, der nur ein oder mehrere Volumes definiert

Links

- Umgebungsvariablen
- /etc/hosts

```
docker run -d -t -p 5000:5000 --name myregistry registry:0.8.1
curl http://localhost:5000
```

```
docker run -d -t -P --name myregistrygui --link myregistry:myregistry -e
REG1=http://myregistry:5000/v1/ atcol/docker-registry-ui
```

BASH

```
docker exec -i -t myregistrygui bash
root@5baad6521a33:/tomcat7# env | sort
...
root@5baad6521a33:/tomcat7# cat /etc/hosts
...
```

Fig



Fig

(<http://www.fig.sh/>)

"Fast, isolated development environments using Docker."

- Entwicklungsumgebungen (z.B. Tools)

- Produktion
- oder auch für Präsentationen :)

BASH

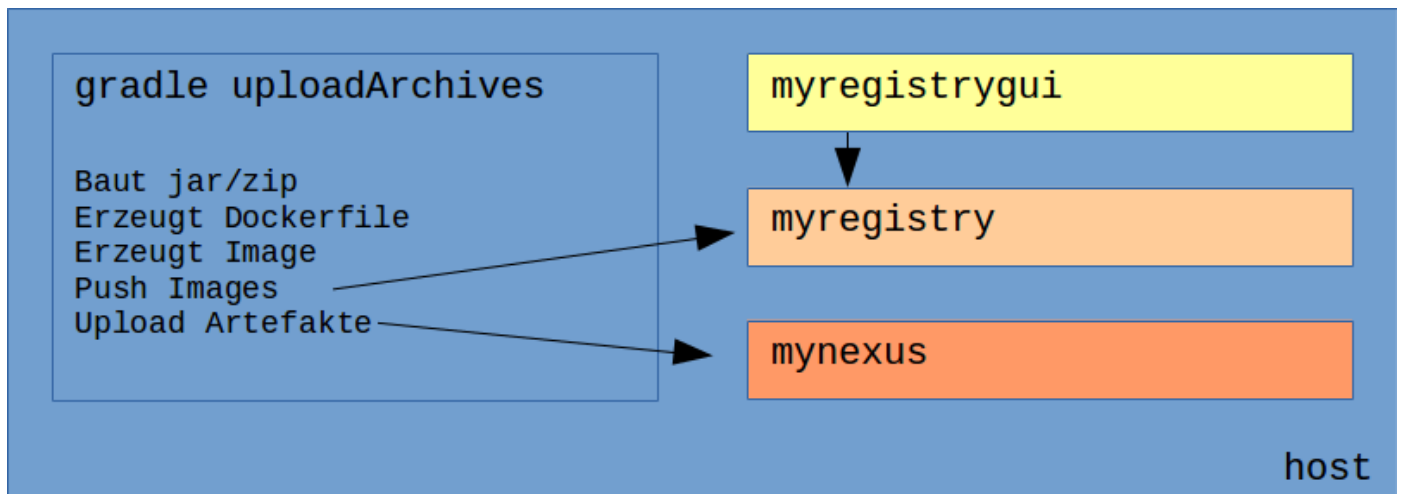
```
cd docker-handson/fig/
less fig.yml
fig up -d
```

Veröffentlichung eines Images

BASH

```
issumy/gradlew uploadArchives
```

```
docker run --rm -i -t -p 9090:8080 myregistry:5000/ollin/issumy
curl http://localhost:9090/issues
```



Geschichte

- internes Projekt bei dotCloud
- erster Release 3.3.2013
- September 2013 redhat Ankündigung ([OpenShift v3](https://blog.openshift.com/openshift-v3-platform-combines-docker-kubernetes-atomic-and-more/)
(<https://blog.openshift.com/openshift-v3-platform-combines-docker-kubernetes-atomic-and-more/>)
)
- Oktober 2013 dotCloud Inc. → Docker Inc.
- Juni 2014 dockercon
 - keynotes from IBM, google, rackspace

- Juli 2014 Windows Azure + Docker
- aktuell Version 1.3.x (exec, create, signing)
- 13'000+ dockerisierte Applikationen
- 2014 15M + 40M

Docker - weitere Themen

- [Docker Documentation](https://docs.docker.com/) (https://docs.docker.com/)
- [CoreOs - Linux for Massive Server Deployments](https://coreos.com/) (https://coreos.com/)
- [Project Atomic - Deploy and Manage Your Docker Containers.](http://www.projectatomic.io/)
(http://www.projectatomic.io/)
- [Kubernetes - Container Cluster Managment](https://github.com/GoogleCloudPlatform/kubernetes)
(https://github.com/GoogleCloudPlatform/kubernetes)
- ...

Diskussion

- Docker - Java?
- Ein Prozess und viele Anwendungen (JEE Modell) vs. viele Prozesse?

Vielen Dank für die Teilnahme!

Präsentation unter: <https://github.com/ollin/pres-docker-dev-view>

email: oliver.nautsch@gmail.com (mailto:oliver.nautsch@gmail.com)

twitter: [@ollispieps](https://twitter.com/ollispieps) (https://twitter.com/ollispieps)



(<http://creativecommons.org/licenses/by/4.0/>)

FAQ

1. *Gibt es die Möglichkeit Applikationen mit graphischen Oberflächen mit Docker zu starten? Im Vortrag sind nur headless Applikationen gezeigt worden.*

Ja, diese Möglichkeit gibt es tatsächlich. Hier ein Beispiel für Firefox und Netbeans: [Running GUI apps with Docker by Fábio Rehm](http://fabiorehm.com/blog/2014/09/11/running-gui-apps-with-docker/)
(<http://fabiorehm.com/blog/2014/09/11/running-gui-apps-with-docker/>).

