

x-celerate

Tipps und Tricks im Umgang mit Docker

Nicholas Dille

Microsoft MVP & Docker Captain

Nicholas Dille

Ehemann, Vater, Geek, Autor, Aikidoka

DevOps Engineer @ Haufe-Lexware

Microsoft MVP seit 2010

Docker Captain seit 2017

<http://dille.name>

NicholasDille





CI/CD

docker-compose.yml

Dockerfile

Image

Registry

Repository

Orchestration

Service

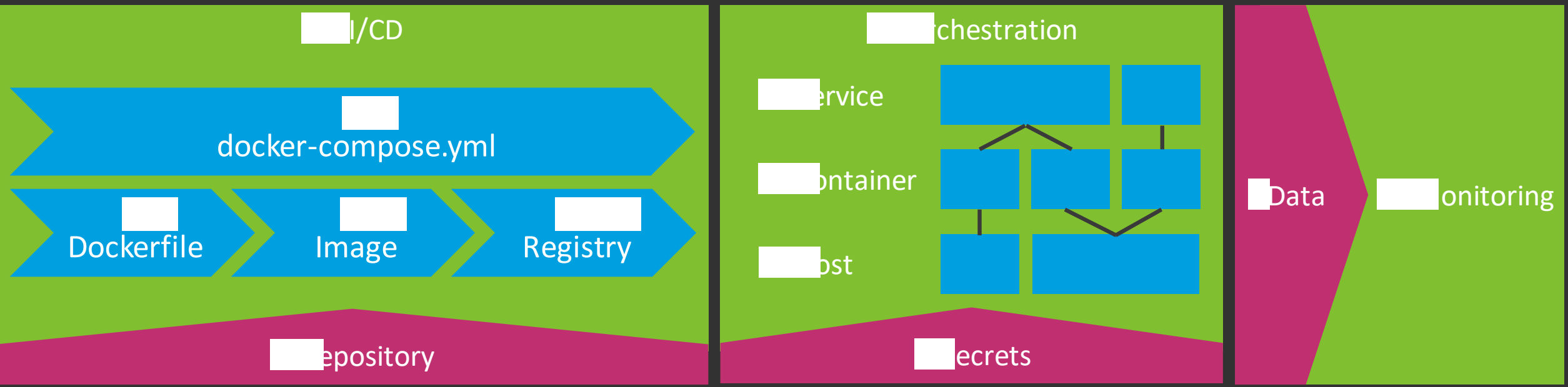
Container

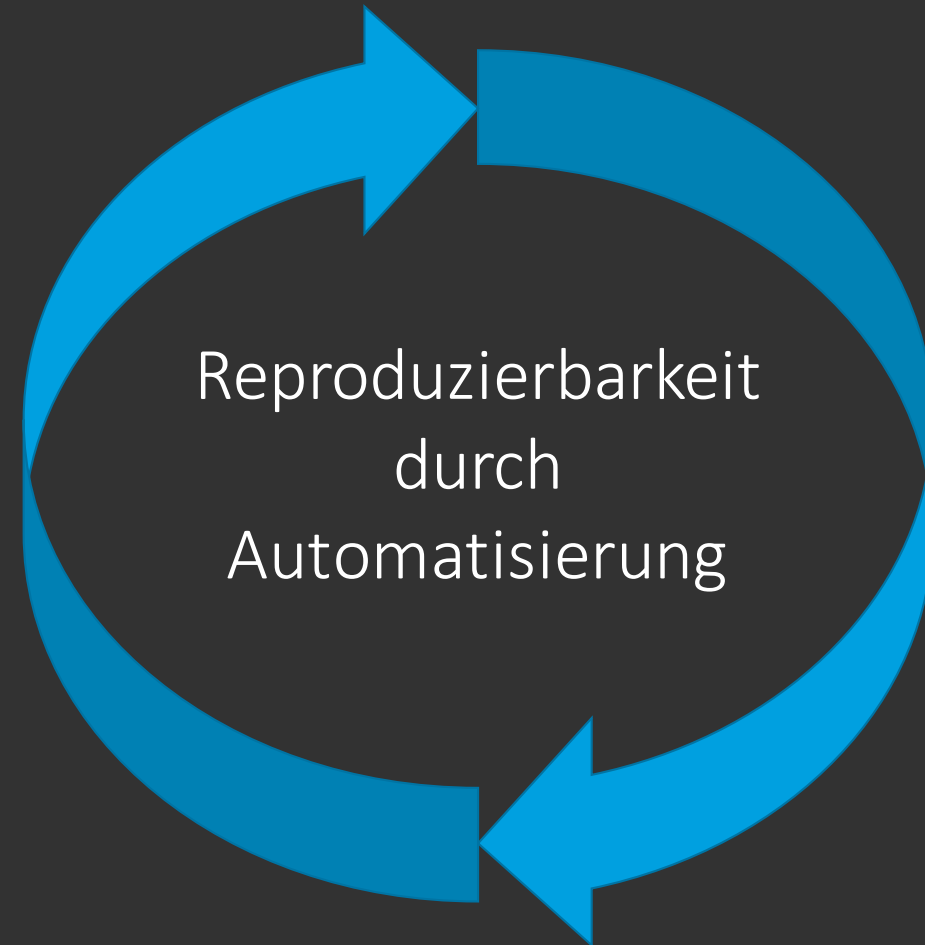
Host

Secrets

Data

Monitoring

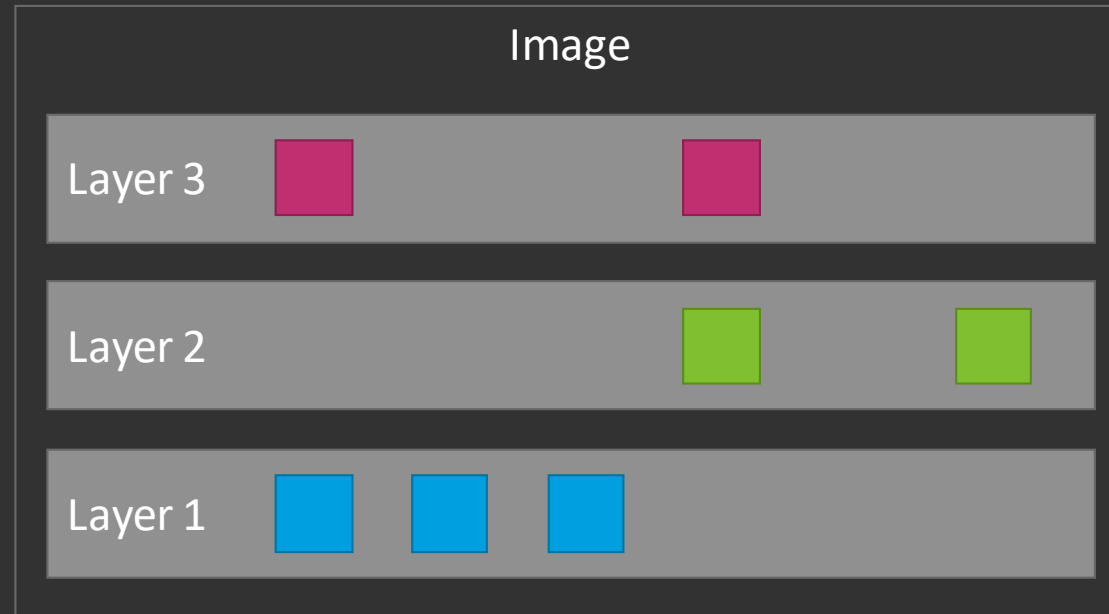




Dockerfile / docker-compose / docker-machine / CI/CD



Ableiten von Code nicht von Images / Ausnahme für offizielle Images



Ableiten von Code nicht von Images / Ausnahme für offizielle Images

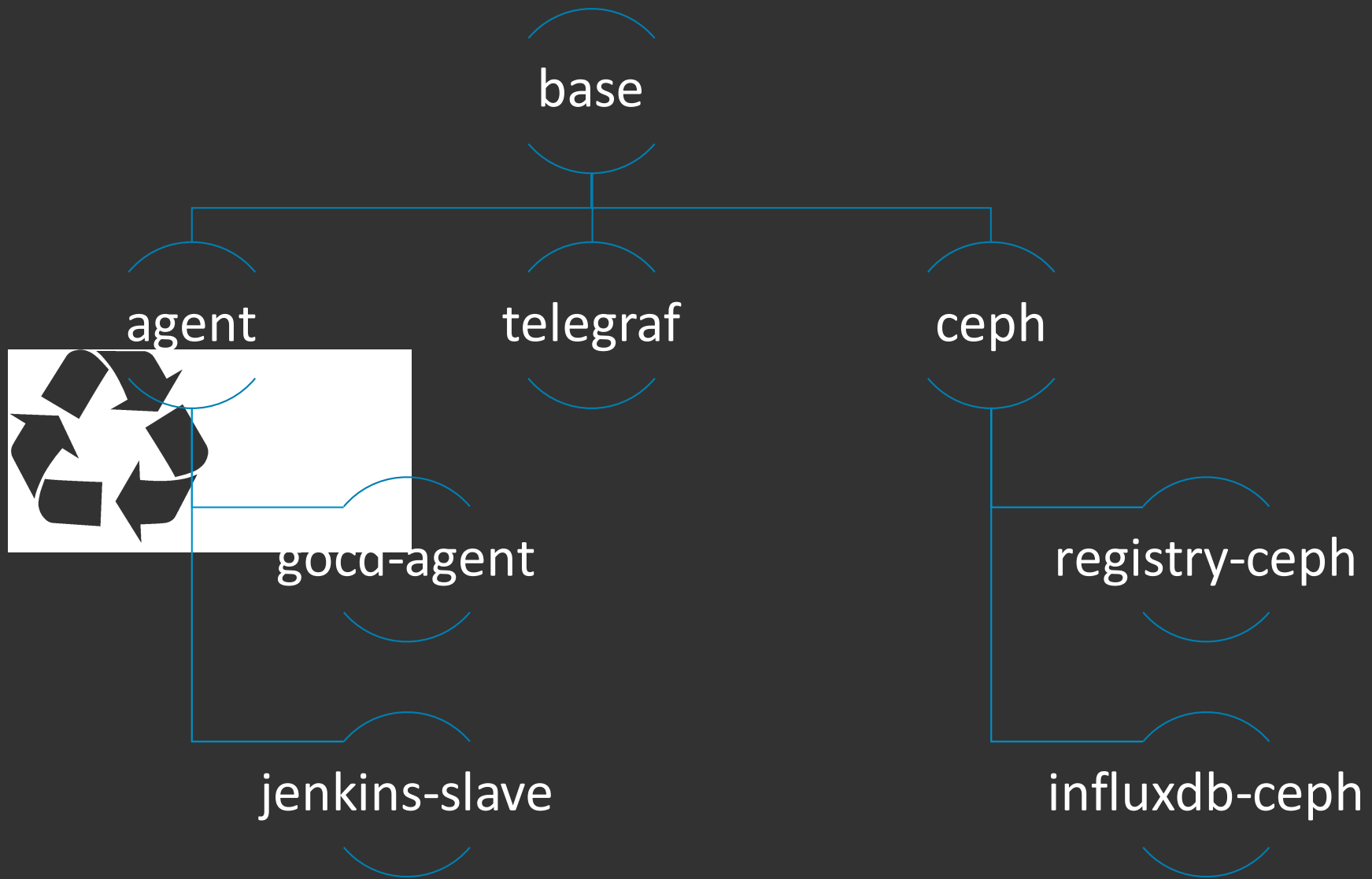
```
FROM ubuntu:xenial-20170915
ENV http_proxy="http://1.2.3.4:8080"
RUN apt update
```

```
docker build --build-arg http_proxy=...
docker run --env http_proxy=...
```

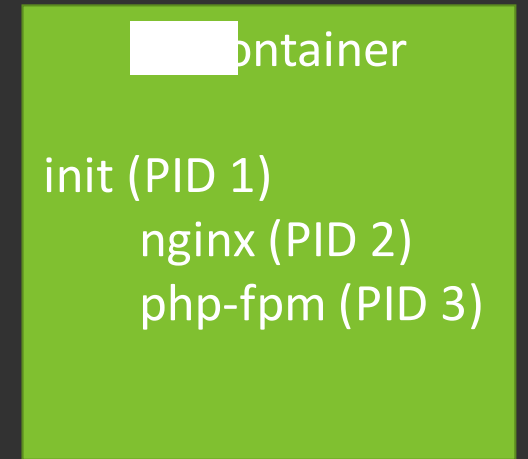
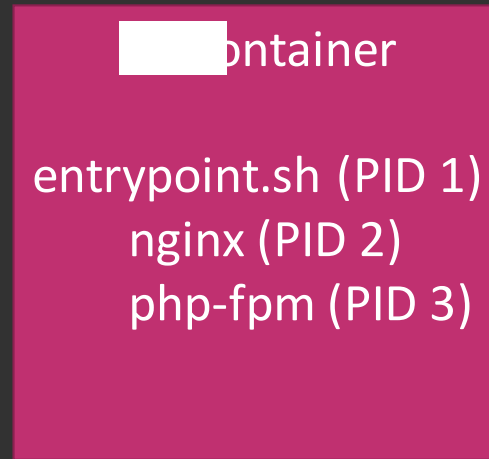
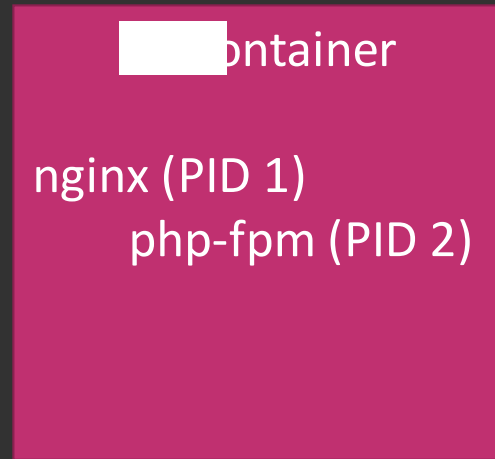
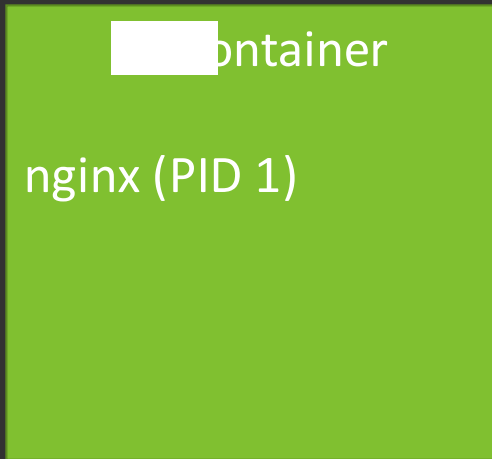
Keine hartkodierte Umgebungsspezifika

```
FROM ubuntu:xenial-20170915
ENV JAVA_VER="8u133"
RUN apt -y install \
    openjdk-jre-8=${JAVA_VER}*
```

Versionsnummern in Variablen definieren



Images wiederverwenden



Nur ein Dienst pro Image / Init-Prozess für mehrere Prozesse
(tini, dumb-init, supervisor, Windows Service Manager)



Rechte einschränken mit USER



Rückverfolgbarkeit durch Microlabeling

```
FROM ubuntu:xenial-20170915
LABEL \
  maintainer=team@x-celerate.de \
  org.label-schema.build-date=„${BUILD_DATE}" \
  org.label-schema.name=„${IMAGE_NAME}" \
  org.label-schema.description=„${IMAGE_DESCRIPTION}" \
  org.label-schema.url=„${PROJECT_URL}" \
  org.label-schema.vcs-ref=„${COMMIT_HASH}" \
  org.label-schema.vcs-url=„${REPO_URL}" \
  org.label-schema.vendor=„${VENDOR_NAME}" \
  org.label-schema.version=„${IMAGE_VERSION}" \
  org.label-schema.schema-version="1.0"

$ docker build \
--build-arg BUILD_DATE=`date +%s` ...
```

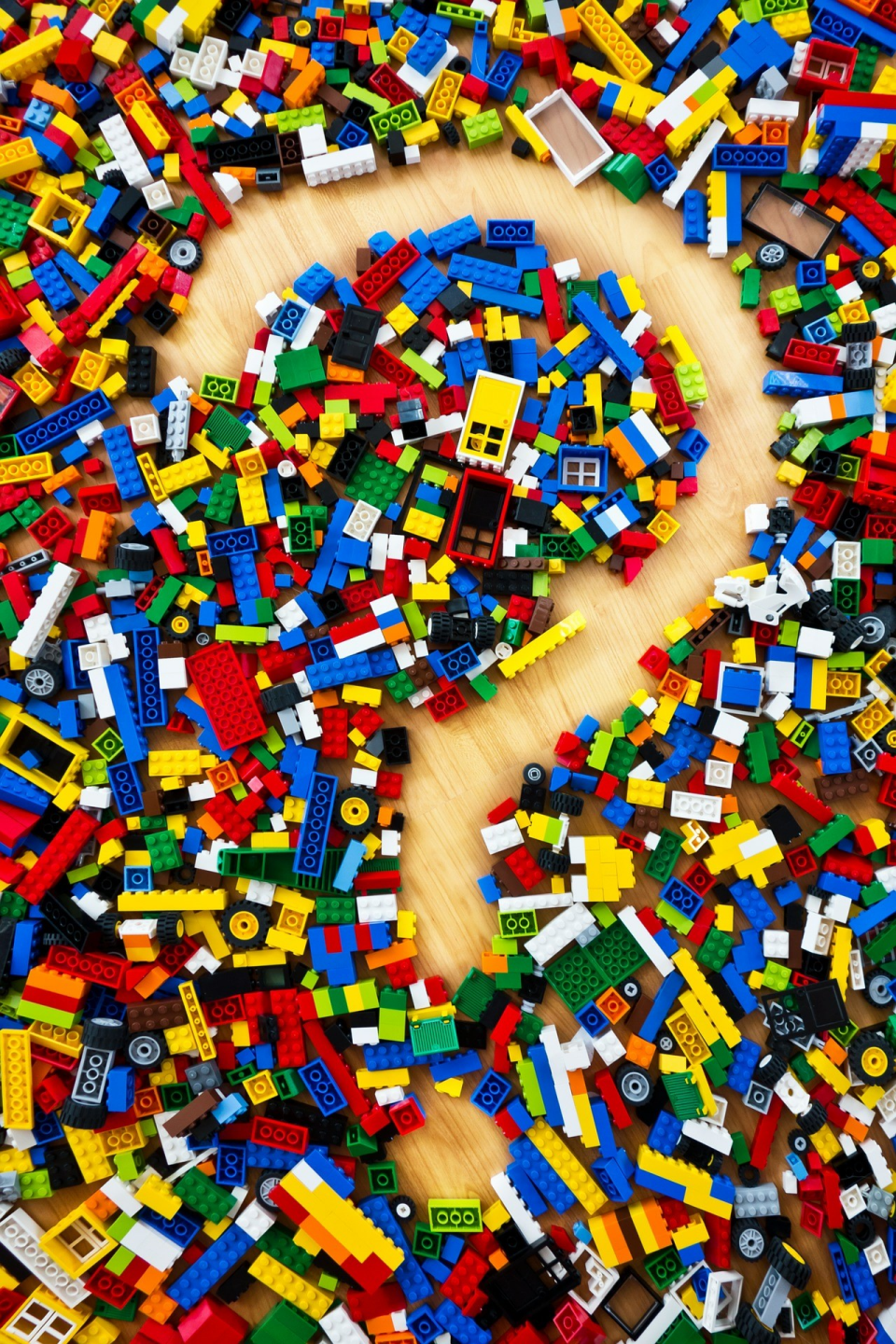
Rückverfolgbarkeit durch Microlabeling / label-schema.org



Continuous Integration / Image prüfen / Test-Deployments



Signierte Pakete / HTTPS / Prüfsummen



Images aktualisieren

```
docker build --pull
```

Unterdrücken des ENTRYPOINT

```
docker run -d --entrypoint bash ubuntu \  
-c 'while true; do sleep 5; done'
```

Ausgabe formatieren

```
docker ps --format \  
"table {{.Names}}\t{{.Image}}\t{{.Status}}"
```



Schnelles Testen

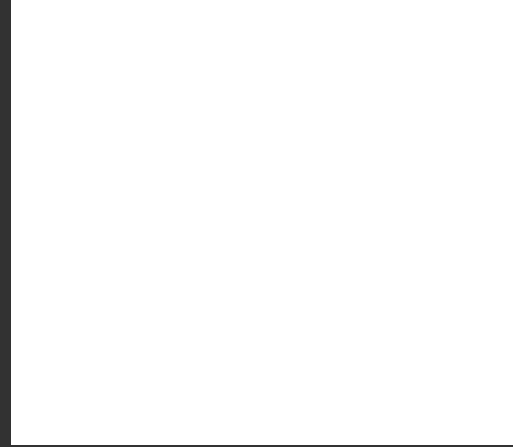
```
docker run -it -rm
```

Aufräumen

```
docker system prune
```

Aufräumen für Image-Builds

```
docker images -q | xargs docker rmi -f  
docker images -q | % { docker rmi -f $_ }
```

Informiert eigene Best Practices entwickeln