



Docker image journey



How to shrink a docker image

Berlin CI/CD Meetup

2018-11-20



Lothar Schulz
@lothar_schulz



WE ARE CONSTANTLY INNOVATING TECHNOLOGY

**HOME-BREWED,
CUTTING-EDGE
& SCALABLE**

technology solutions



help our brand to
WIN ONLINE



> 2,000

employees at

8 international
tech locations

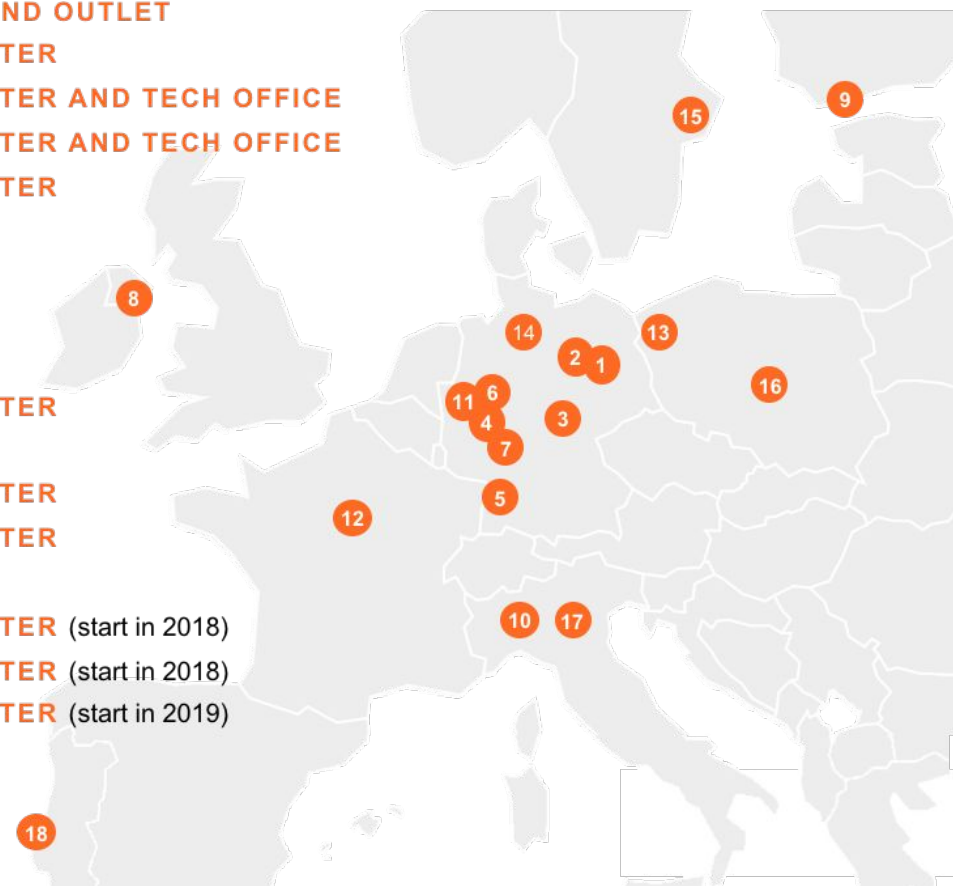


HQs
in Berlin



OUR FOOTPRINT AROUND EUROPE

1	BERLIN	HEADQUARTERS AND OUTLET
2	BRIESELANG	FULFILLMENT CENTER
3	ERFURT	FULFILLMENT CENTER AND TECH OFFICE
4	MÖNCHENGLADBACH	FULFILLMENT CENTER AND TECH OFFICE
5	LAHR	FULFILLMENT CENTER
6	DORTMUND	TECH HUB
7	FRANKFURT	OUTLET
8	DUBLIN	TECH HUB
9	HELSINKI	TECH HUB
10	MILAN (STRADELLA)	FULFILLMENT CENTER
11	COLOGNE	OUTLET
12	PARIS (MOISSY-CRAMAYEL)	FULFILLMENT CENTER
13	SZCZECIN (GRYFINO)	FULFILLMENT CENTER
14	HAMBURG	ADTECH LAB
15	STOCKHOLM (BRUNNA)	FULFILLMENT CENTER (start in 2018)
16	LODZ (GLUCHOW)	FULFILLMENT CENTER (start in 2018)
17	VERONA (NOGAROLE ROCCA)	FULFILLMENT CENTER (start in 2019)
18	LISBON	TECH-HUB



Me

Engineering Lead at Zalando

Code

- <https://github.com/zalando>
- <https://github.com/lotharschulz>

[@lothar_schulz](#)

lotharschulz.info







Why reduce docker image size ?

- Pulling and pushing docker images from and to remote docker registries is faster
- Security attack surface is often smaller



Proof?

Team Journey part 1

```
FROM golang:1.11
```

```
RUN mkdir /app
```

```
WORKDIR /app
```

```
COPY hello.go .
```

```
RUN go build -o hellogo .
```

```
RUN groupadd -g 99 appuser && useradd -r -u 99 -g appuser appuser
```

```
USER appuser
```

```
CMD ["/hellogo"]
```

```
$ docker build --rm -t ...
```

Team Journey part 1

Image size compressed: 299 MB

CircleCI build time: ~ 3 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 21 ses

pull to empty nodes, no layers cached

Team Journey part 1.1

```
FROM golang:1.11

RUN mkdir /app
WORKDIR /app
COPY hello.go .

RUN go build -o hellogo .

RUN groupadd -g 99 appuser && useradd -r -u 99 -g appuser appuser
USER appuser

CMD ["/hellogo"]
```

```
$ docker build --cache-from golang:1.11 -t ...
```

Team Journey part 1.1

previous step

Image size compressed: 299 MB

CircleCI build time: ~ 3 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 21 ses



current step

Image size compressed: 299 MB

CircleCI build time: ~ **0.5 sec**

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 21 ses

cache warm up in preparation step

Team Journey part 2 - builder pattern

#part one

FROM golang:1.11 as builder

workdir setup

COPY hello.go .

RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -ldflags '-w -extldflags "-static"' -o hellogodocker .

part two

FROM alpine:latest

user & workdir setup

COPY --from=builder /go/src/github.com/lotharschulz/hellogodocker/hellogodocker .

CMD ["/hellogodocker"]

\$ docker build --rm -t ...

use a static linked binary

Team Journey part 2 - builder pattern

previous step

Image size compressed: **299 MB**
CircleCI build time: ~ 0.5 sec
Dockerhub push time: ~ 4 sec
Dockerhub pull time: ~ **21 ses**



current step

Image size compressed: **5 MB**
CircleCI build time: ~ 0.6 sec
Dockerhub push time: ~ 4 sec
Dockerhub pull time: ~ **1.2 ses**

Team Journey part 2 - builder pattern - squash & compress

```
#part one
FROM golang:1.11 as builder
# workdir setup
COPY hello.go .

RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -ldflags '-w -extldflags
"-static"' -o hellogodocker .

# part two
FROM alpine:latest
# user & workdir setup ....
COPY --from=builder /go/src/github.com/lotharschulz/hellogodocker/hellogodocker .

CMD ["/hellogodocker"]
```

```
$ docker build --squash/--compress --rm -t ...
```

squash is an [experimental Docker daemon option](#)

Team Journey part 2 - builder pattern - squash & compress

previous step



current step

Image size compressed: 5 MB

CircleCI build time: ~ 0.6 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 1.2 ses

Image size compressed: 5 MB

CircleCI build time: ~ 0.7/1.1 sec

Dockerhub push time: ~ 4/4.9 sec

Dockerhub pull time: ~ 1.1/1.4 ses

Team Journey part 3 - alpine base

```
FROM alpine:latest  
# ... set up user
```

```
ADD hellogo /  
CMD ["/hellogo"]
```

```
$ docker build [--squash/--compress] --rm -t ...
```

static linked binary created within ci worker node outside of Dockerfile scope

Team Journey part 3 - alpine base

previous step



current step

Image size compressed: 5 MB

CircleCI build time: ~ 0.7/1.1 sec

Dockerhub push time: ~ 4/4.9 sec

Dockerhub pull time: ~ 1.1/1.4 ses

Image size compressed: 5 MB

CircleCI build time: ~ 0.6/1.1 sec

Dockerhub push time: ~ 3.8/4.6 sec

Dockerhub pull time: ~ 1.1 ses

Team Journey part 4 - FROM scratch

FROM scratch

ADD ca-certificates.crt /etc/ssl/certs/

ADD hellogo /

CMD ["/hellogo"]

```
$ docker build [--squash/--compress] --rm -t ...
```

static linked binary created within ci worker node outside of Dockerfile scope

Team Journey part 4 - FROM scratch

previous step

Image size compressed: 5 MB

CircleCI build time: ~ 0.6/1.1 sec

Dockerhub push time: ~ 3.8/4.6 sec

Dockerhub pull time: ~ 1.1 ses

current step



Image size compressed: 2 MB



CircleCI build time: ~ 16 sec

Dockerhub push time: ~ 3.5/3.9 sec

Dockerhub pull time: ~ 0.8 ses

What about Ops?



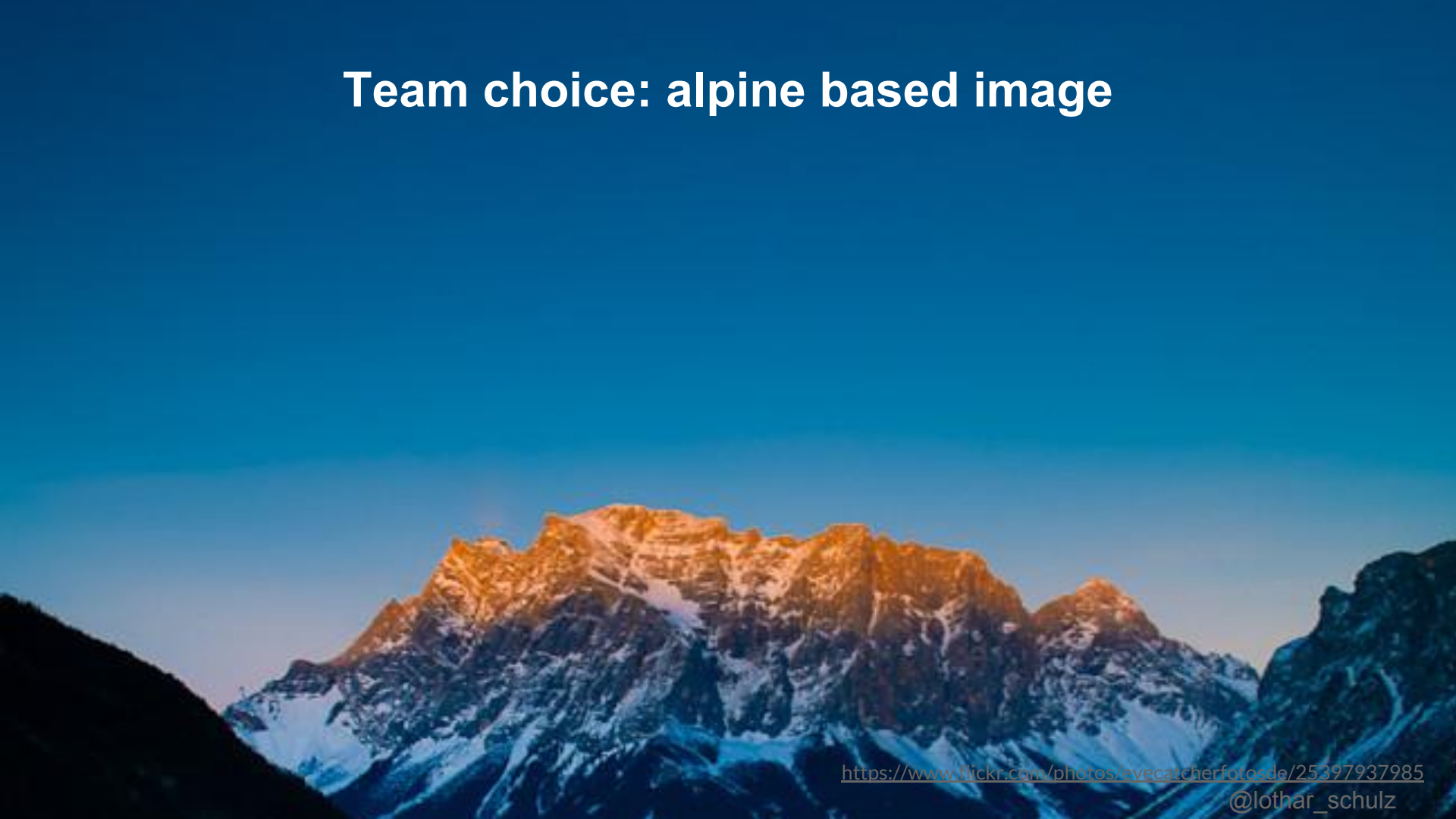
```
$ docker exec -it ghe-ms sh
```

```
OCI runtime exec failed: exec failed: container_linux.go:348:
```

```
starting container process caused
```

```
"exec: \"sh\": executable file not found in $PATH": unknown
```

Team choice: alpine based image



<https://www.flickr.com/photos/eyecatcherfoto-de/25397937985>

@lothar_schulz

Some numbers



image size compressed: 299 MB

Dockerhub pull time: ~ 20 sec

0.6 %
5 %



image size compressed: 2 MB

Dockerhub pull time: ~ 1 sec

attack surface - FROM golang:1.11

```
CLAIR_OUTPUT=High CLAIR_ADDR=http://192.168.99.100:30060 klar  
lotharschulz/hellogo:build.docker-cache--0.2.91
```

```
clair timeout 1m0s
```

```
docker timeout: 1m0s
```

```
no whitelist file
```

```
Analysing 10 layers
```

```
Got results from Clair API v1
```

Found 131 vulnerabilities

Unknown: 16

Negligible: 28

Low: 47

Medium: 31

High: 9

attack surface - FROM scratch

```
CLAIR_OUTPUT=High CLAIR_ADDR=http://192.168.99.100:30060 klar  
lotharschulz/hellogo:build.docker-min-compress--0.2.91  
clair timeout 1m0s  
docker timeout: 1m0s  
no whitelist file  
Analysing 2 layers  
Got results from Clair API v1  
Found 0 vulnerabilities
```

Why would you matter ?

- potentially reduced developer cycle time (build)
- potentially reduced container startup time (pull)
- potentially reduced attack surface

I'm sure you have questions

@lothar_schulz
lotharschulz.info

<https://www.flickr.com/photos/eyecatcherfoto/de/25397937985>