



# Docker image journey

## How to shrink a docker image

DevOps Pro Europe 2019  
2019-03-20

MOOVEL BECOMES

REACHNOW ✓

pic: © Moovel



MOOVEL BECOMES

REACHNOW ✓

# Me



## Engineering Manager

[github.com/lotharschulz](https://github.com/lotharschulz)

[@lothar\\_schulz](https://twitter.com/lothar_schulz)

[lotharschulz.info](https://lotharschulz.info)

[speakerdeck.com/lothar](https://speakerdeck.com/lothar)





pic: <https://pixabay.com/en/turtle-tortoise-swim-sea-turtle-863336>



pic: <https://www.flickr.com/photos/eyecatcherfotosde/41584801594>

# 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?*

*Well ...*



# Team Journey part 1

```
FROM golang:1.12
```

```
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 ...  
.....
```

<https://github.com/lotharschulz/hellogodocker/blob/master/Dockerfile>

# Team Journey part 1

Image size: 780 MB (compressed 299 MB)

CircleCI build time: ~ 0.5 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 21 sec

<https://circleci.com/gh/lotharschulz/hellogodocker/105>

pull to empty nodes, no layers cached

## Team Journey part 1.1

```
FROM golang:1.12
```

```
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:1.12 -t ...  
.....
```

<https://github.com/lotharschulz/hellogodocker/blob/master/Dockerfile>

# Team Journey part 1.1

## previous step

Image size compressed: 780 MB (c: 299)

CircleCI build time: ~ 0.5 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 21 sec

## current step

Image size compressed: 780 MB (c: 299)

CircleCI build time: ~ 0,6 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ 18 sec

<https://circleci.com/gh/lotharschulz/hellogodocker/105>

cache warm up in preparation step

## Team Journey part 2 - builder pattern

**#part one**

**FROM golang:1.12 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 ...**

---

<https://github.com/lotharschulz/hellogodocker/blob/master/DockerfileBuilder>

please note the static linked binary



## Team Journey part 2 - builder pattern

previous step

Image size compressed: **780 MB**

CircleCI build time: ~ 0.6 sec

Dockerhub push time: ~ 4 sec

Dockerhub pull time: ~ **18 sec**



current step

Image size compressed: **11.9 MB** (c: 5)

CircleCI build time: ~ 0.6 sec

Dockerhub push time: ~ 3.4 sec

Dockerhub pull time: ~ **1.2 sec**



<https://circleci.com/gh/lotharschulz/hellogodocker/105>

## Team Journey part 2 - builder pattern - squash & compress

```
#part one
FROM golang:1.12 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 ...
.....
```

<https://github.com/lotharschulz/hellogodocker/blob/master/DockerfileBuilder>

squash is an experimental feature

## Team Journey part 2 - builder pattern - squash & compress

previous step



current step

Image size compressed: 11.9 MB (c: 5)

CircleCI build time: ~ 0.6 sec

Dockerhub push time: ~ 3.4 sec

Dockerhub pull time: ~ 1.2 sec

Image size compressed: 11.9 MB (c: 5)

CircleCI build time: ~ 0.6/1.1 sec

Dockerhub push time: ~ 3.5/4.1 sec

Dockerhub pull time: ~ 1.1/1.3 sec

<https://circleci.com/gh/lotharschulz/hellogodocker/98>

## 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

<https://circleci.com/gh/lotharschulz/hellogodocker/105>

## Team Journey part 3 - alpine base

previous step



current step

Image size compressed: 11.9 MB (c: 5)

CircleCI build time: ~ 0.6/1.1 sec

Dockerhub push time: ~ 3.5/4.1 sec

Dockerhub pull time: ~ 1.1/1.3 sec

Image size compressed: 12.2 MB (c: 5)

CircleCI build time: ~ 0.6 sec

Dockerhub push time: ~ 3.5 sec

Dockerhub pull time: ~ 1.2 sec

<https://circleci.com/gh/lotharschulz/hellogodocker/105>

## 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

<https://github.com/lotharschulz/hellogodocker/blob/master/DockerfileAlpine>

## Team Journey part 4 - FROM scratch

previous step

Image size compressed: 12.2 MB

CircleCI build time: ~ 0.6 sec

Dockerhub push time: ~ 3.5 sec

Dockerhub pull time: ~ 1.2 sec



current step

Image size compressed: **4.8 MB** (c: 2)

CircleCI build time: ~ **15 sec**

Dockerhub push time: ~ 3.5 sec

Dockerhub pull time: ~ 0.7 sec

<https://circleci.com/gh/lotharschulz/hellogodocker/105>

# What about Ops?



```
$ docker run -p 1234:1234 --name hellogodocker lotharschulz/hello:build.docker-min--0.2.102
```

```
$ docker exec -it hellogodocker sh
```

```
OCI runtime exec failed: exec failed: container_linux.go:344:
```

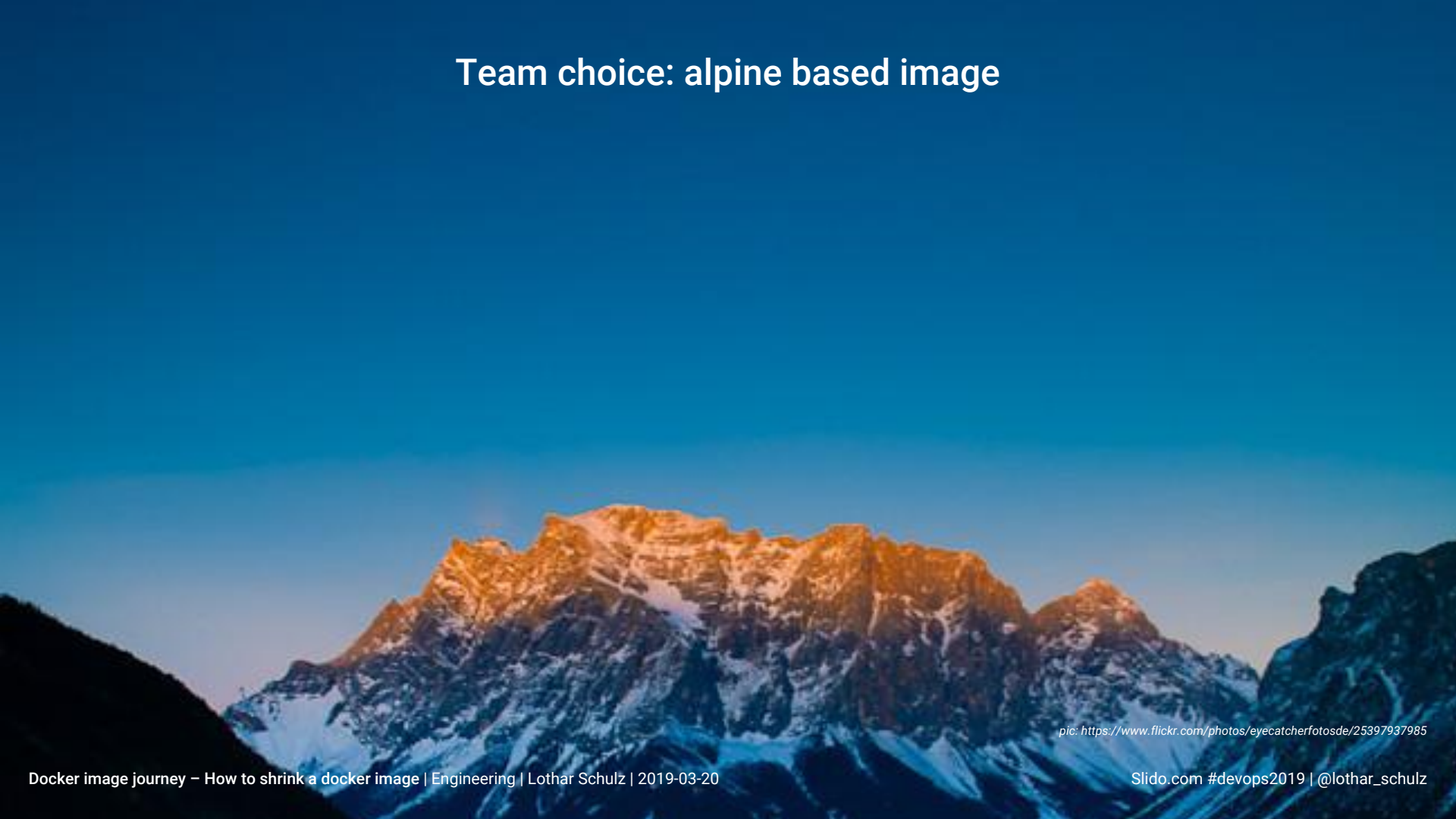
```
starting container process caused "exec: \"sh\":
```

```
executable file not found in $PATH": unknown
```

pic: <https://pixabay.com/en/caution-label-warning-red-mark-943376/>



# Team choice: alpine based image



pic: <https://www.flickr.com/photos/eyecatcherfotosde/25397937985>

## Some numbers



image size compressed: 780 MB

Dockerhub pull time: ~ 21 sec

**0.6%**

**5%**



image size compressed: 4,8 MB

Dockerhub pull time: ~ 0.7 sec

## attack surface - FROM golang:1.12

```
CLAIR_OUTPUT=High CLAIR_ADDR=http://192.168.99.100:30060 klar  
lotharschulz/hello:build.docker-cache--0.2.101  
clair timeout 1m0s  
docker timeout: 1m0s  
no whitelist file  
Analysing 11 layers  
Got results from Clair API v1  
Found 300 vulnerabilities  
Unknown: 8  
Negligible: 71  
Low: 95  
Medium: 107  
High: 19
```

## attack surface - FROM scratch

```
CLAIR_OUTPUT=High CLAIR_ADDR=http://192.168.99.100:30060 klar  
lotharschulz/hello:build.docker-min--0.2.101  
clair timeout 1m0s  
docker timeout: 1m0s  
no whitelist file  
Analysing 2 layers  
Got results from Clair API v1  
Found 0 vulnerabilities
```

## Conclusions

# Balance

- docker image size
- security considerations
- infrastructural effort

# Conclusions

- potentially reduced container startup time (pull)
- potentially reduced attack surface

*"Small Images are always better in terms of absolute performance"*

# I'm sure you have questions

lotharschulz.info

pic: <https://www.flickr.com/photos/eyecatcherfotosde/25397937985>

# Backup



[illegible]

## 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