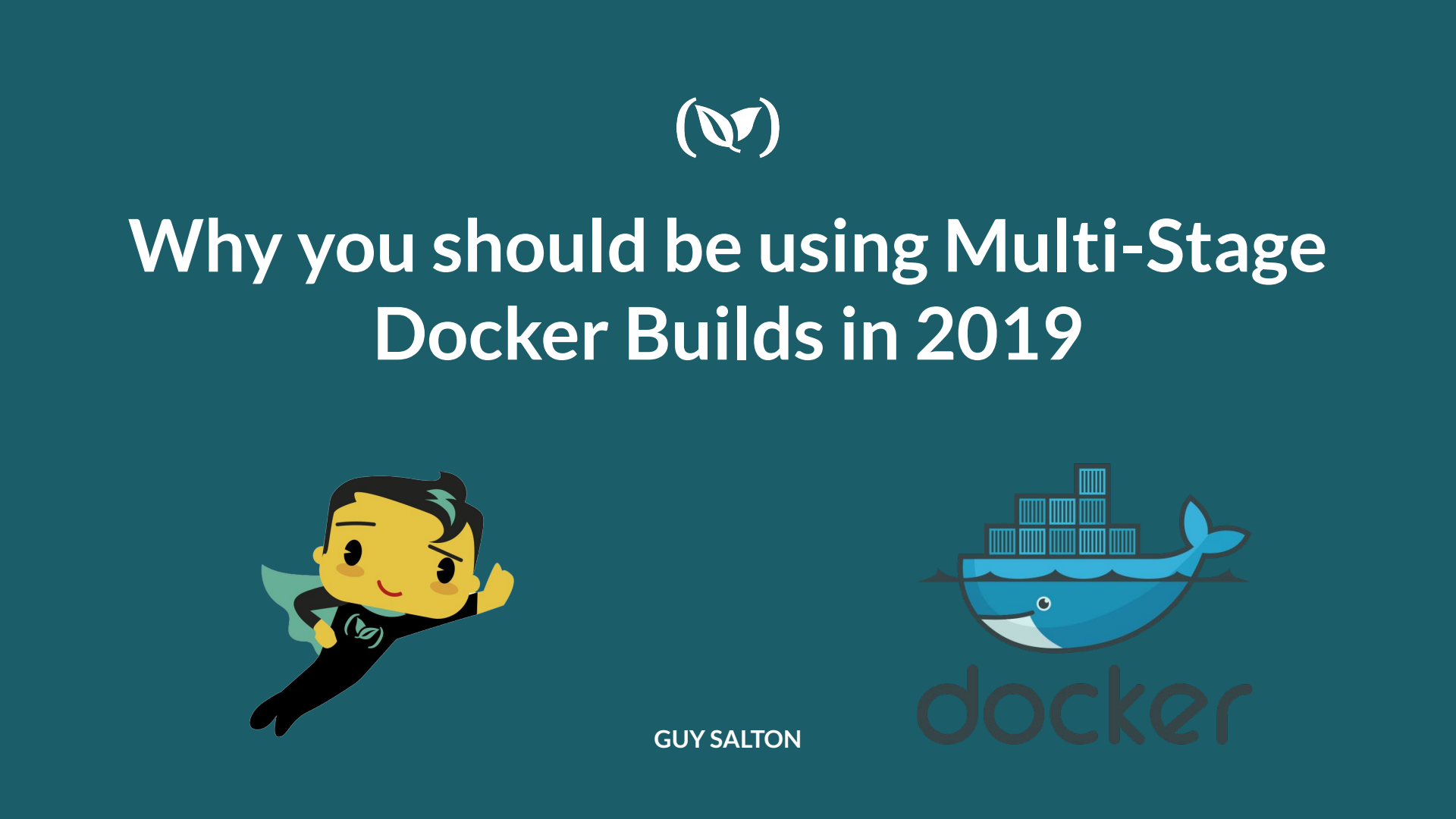


Why you should be using Multi-Stage Docker Builds in 2019

GUY SALTON

Guy Salton

Solutions Architect

codefresh

guy.salton@codefresh.io

Agenda

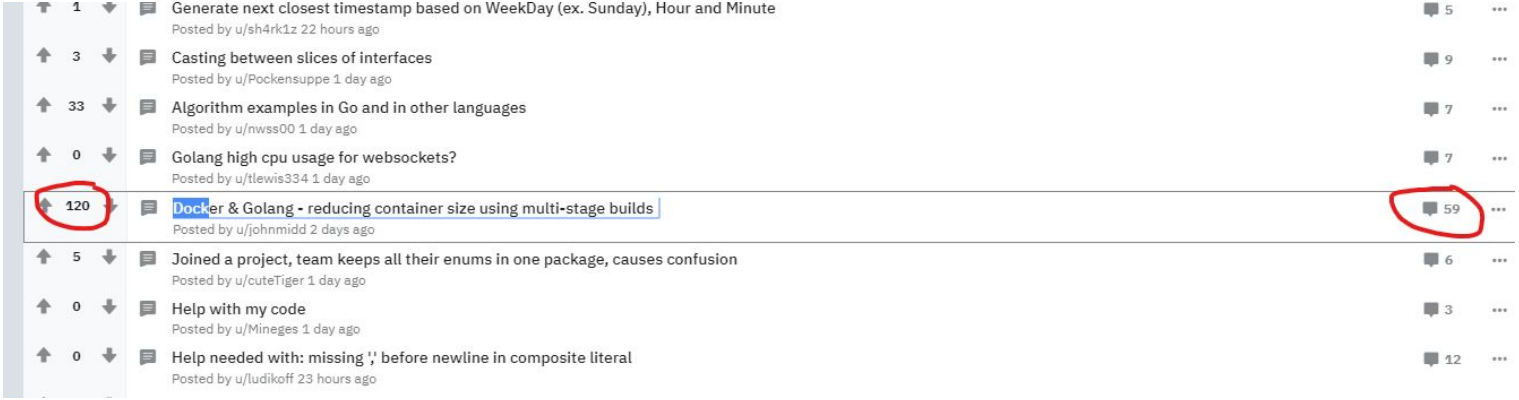
- Dockerfile and Docker build
- The problem with Docker build
- Solution: Docker multi-stage build
- Docker multi-stage build in CI/CD
- Summary

https://github.com/codefresh-contrib/golang-sample-app/

https://codefresh.io/docs/docs/learn-by-example/golang/golang-hello-world/

Docker multi-stage build

Multi-stage build is available starting from Docker 17.05 (released in 2017!) - so why now?



https://www.reddit.com/r/golang/comments/crkibq/docker_golang_reducing_container_size_using/

- “oh man! thank you! I've been fighting with build times on an image stack for weeks”
- “Wow, this is great for deploying tonnes of microservices!”
- “I've been meaning to use multistage builds, thanks for the walkthrough!”

Dockerfile and Docker build

- **Dockerfile** - imperative DSL that defines build commands
- Each **Docker build** command generates ONE image layer
- Complete **Docker build** execution generates ONE Docker image

Dockerfile and Docker build

```
1  FROM golang:1.7.1
2
3  # Copy everything from the src directory to /go/src directory inside the container
4  COPY src /go/src
5
6  # Build the Go app
7  RUN CGO_ENABLED=0 GOOS=linux go build -o bin/sample src/sample/trivial-web-server.go
8
9  # This container exposes port 8080 to the outside world
10 EXPOSE 8080
11
12 # Run the binary program
13 CMD ["./bin/sample"]
```

Demo 1:
Docker build on GO app

https://github.com/codefresh-contrib/helm-sample-app

The Problem with Docker build

Image we want

runtime
configuration
application

X (4..10)

Image we build

Compilers, debuggers, ...
Linters, tests, profilers, ...
code, build and test logs, ...
runtime
configuration
application

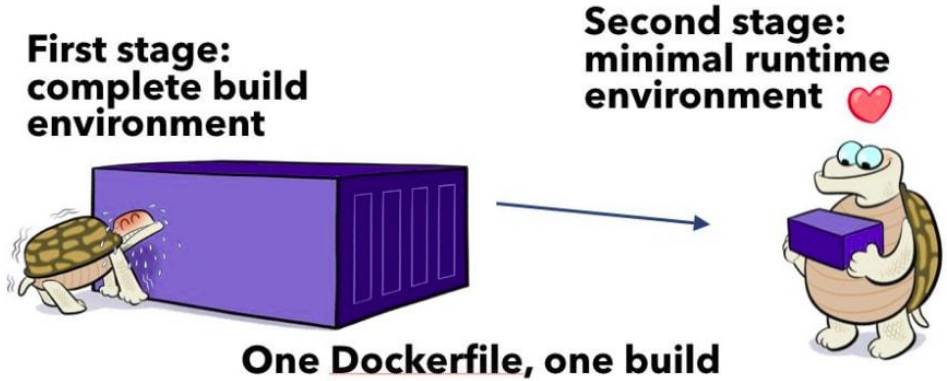
The Problem with Docker build

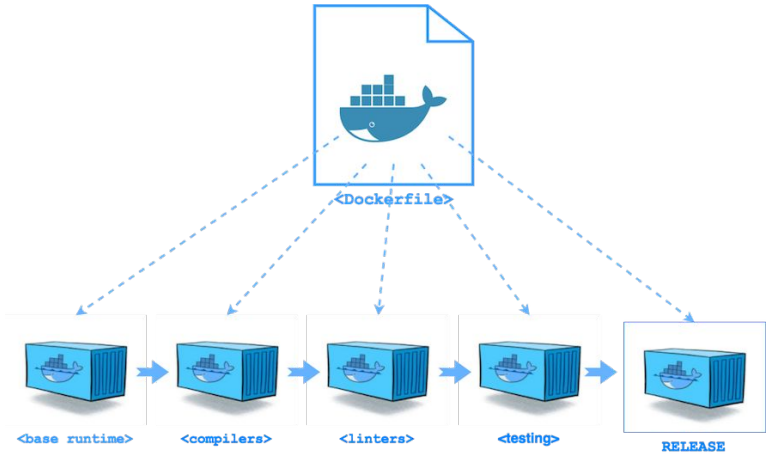
- **2 Dockerfiles**
  - 1st for build tools
  - 2nd for runtime

- **Drawbacks**
  - 2+ Dockerfiles
  - Orchestration needed: Bash, make, YAML, ...

Solution: Docker multi-stage build

- **Benefits**
  - One Dockerfile
  - One syntax to learn
  - Same build
    - Local and CI
  - Create multiple stages



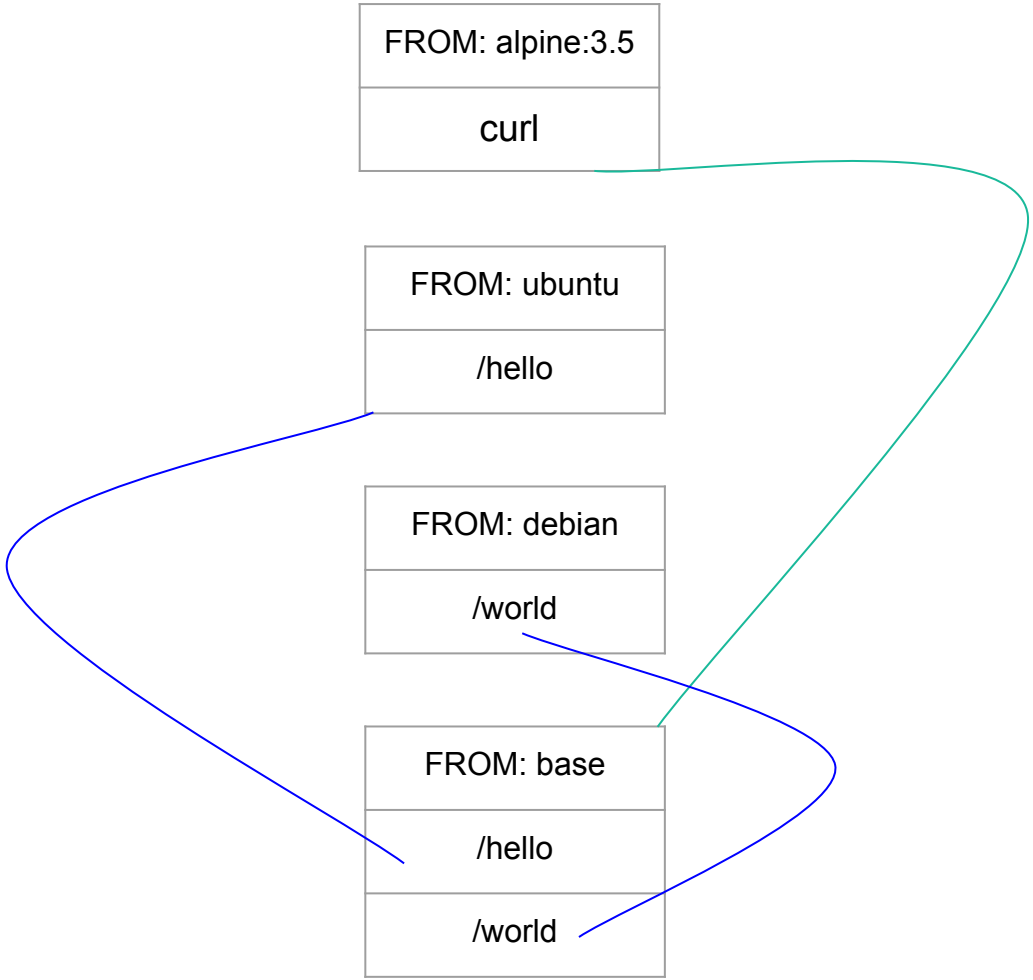


```dockerfile
# Base Image
FROM alpine:3.5 AS base
RUN apk add --no-cache curl

# Second Image
FROM debian AS second
RUN echo hello > /hello
LABEL image=second

# Third Image
FROM ubuntu AS third
RUN echo world > /world
LABEL image=third

# FINAL Image
FROM base
# Copy files from other images
COPY --from=second /hello /hello
COPY --from=third /world /world
RUN curl --version
```



Demo 2:
Docker multi-stage build

Docker multi-stage build

You can enjoy multi-stage build with every programming language (not only GO):

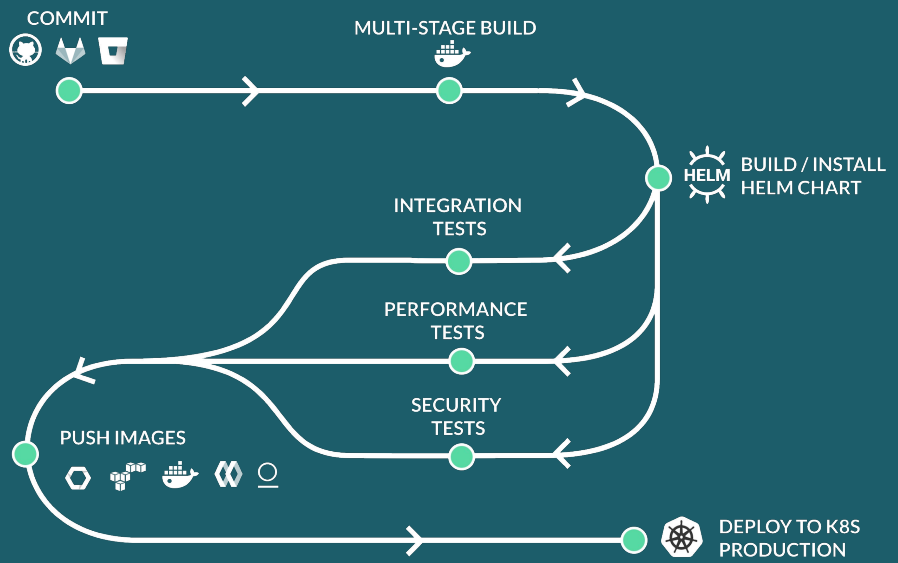
- GO example - https://codefresh.io/docs/docs/learn-by-example/golang/golang-hello-world/#create-a-multi-stage-docker-image-for-go 
- JAVA example - https://codefresh.io/docs/docs/learn-by-example/java/spring-boot-2/#spring-boot-2-and-docker-multi-stage-builds 
- Node example - https://codefresh.io/docs/docs/learn-by-example/nodejs/react/#react-and-docker-multi-stage-builds 
- PHP example - https://codefresh.io/docs/docs/learn-by-example/php/#the-example-php-project 

Docker anti-patterns https://codefresh.io/containers/docker-anti-patterns/

Codefresh

The 1st container-native CI/CD Platform for Microservices

- Container-native
- Intuitive & Robust
- Enterprise Ready
- Flexible Delivery



Demo 3:
Docker multi-stage build in CI/CD

Summary

- Using 1 Docker image for both build and production results in slow deployment and lots of CVE violations
- Multi-stage build to produce lean, secure and production ready Docker image
- On Codefresh, speedier builds thanks to caching across all images and layers

Thank You!

Build Fast, Deploy Faster

Signup for a FREE account with UNLIMITED builds

& schedule a 1:1 with our experts at https://codefresh.io

guy.salton@codefresh.io