Dockerizing PHP Apps

Aurelijus Banelis

ŠiauliaiPHP
2019-01-31

Aurelijus Banelis

Backend/DevOps

aurelijus.banelis.lt
aurelijus@banelis.lt

```
PGP    0x320205E7539B6203
130D C446 1F1A 2E50 D6E3
3DA8 3202 05E7 539B 6203
```

Adapting **PHP** applications to be used with **docker**

Introduction

- Why to dockerize
- What is docker
- How to dockerize

Development

- Context/Alternatives
- What I really liked
- What I do not like

Production

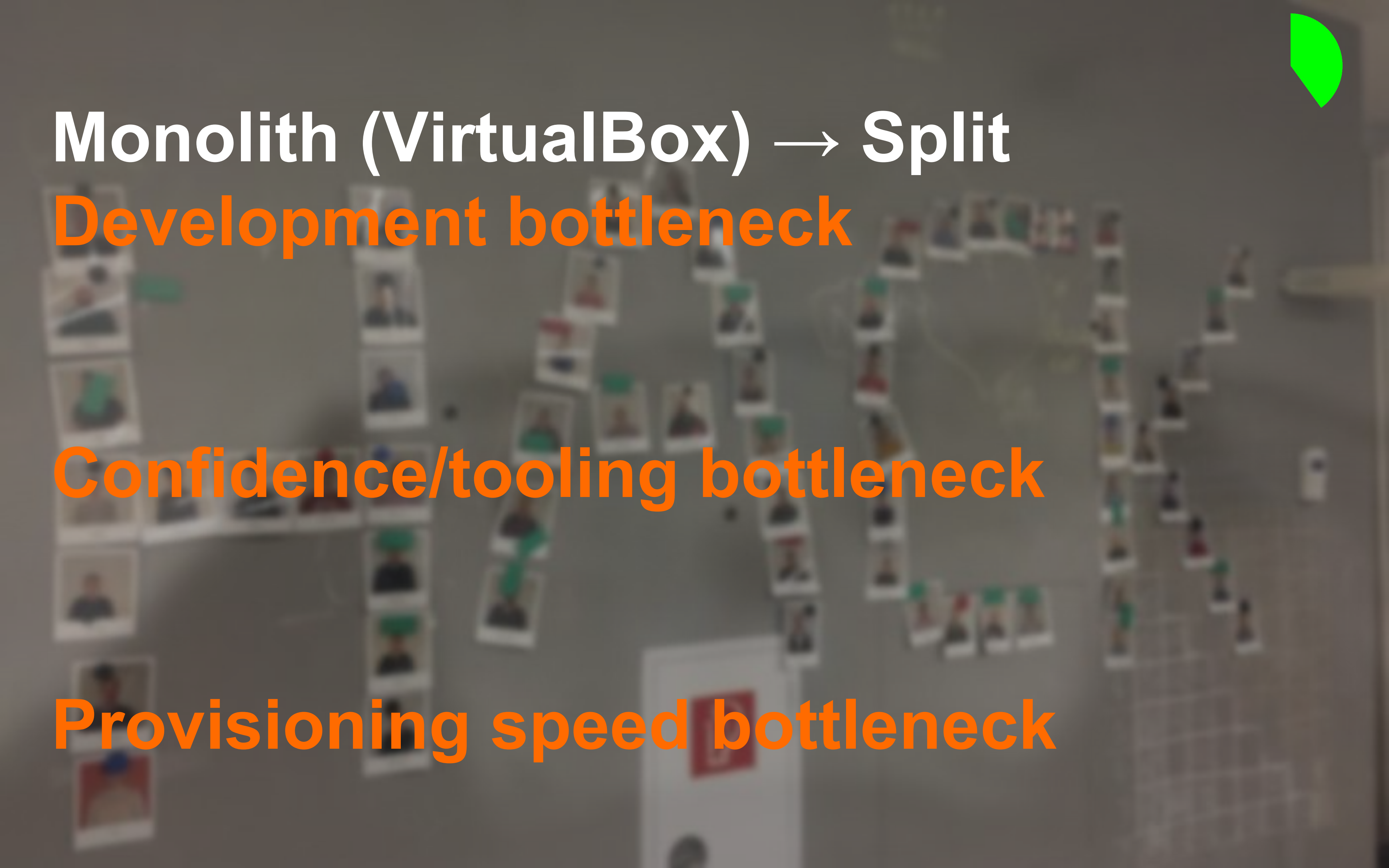
- Deploying to AWS
- Logging challenges

Introduction	Why to dockerize What is docker How to dockerize
Development	Context/Alternatives What I really liked What I do not like
Production	Deploying to AWS Logging challenges

Development bottleneck

Confidence/tooling bottleneck

Provisioning speed bottleneck



Monolith (VirtualBox) → Split

Development bottleneck

Confidence/tooling bottleneck

Provisioning speed bottleneck

Development bottleneck

Jenkins (time based) → CircleCI

Confidence/tooling bottleneck

Provisioning speed bottleneck

Development bottleneck

Confidence/tooling bottleneck

Release coordination → AWS services

Provisioning speed bottleneck

Bottleneck to grow

Monolith (VirtualBox) → Split
Development bottleneck

Jenkins (time based) → CircleCI
Confidence/tooling bottleneck

Release coordination → AWS services
Provisioning speed bottleneck

Monolith (VirtualBox) → Split

Development bottleneck

The need for better virtualization tools

Jenkins (time based) → CircleCI

Confidence/tooling bottleneck

Release coordination → AWS services

Provisioning speed bottleneck

Monolith (VirtualBox) → Split

Development bottleneck

Jenkins (time based) → CircleCI

Confidence/tooling bottleneck

Release coordination → AWS services

Provisioning speed bottleneck

The need for better virtualization tools

docker?

We migrated to docker

Using docker extensively at home24

org:Home24 filename:Dockerfile

Pull requests Issues Marketplace Explore

Repositories

Code

Commits

Issues

Marketplace

Topics

Wikis

Users

Languages

Dockerfile	157
PHP	

Advanced search Cheat sheet

Showing available code results

Home24/ Dockerfile

Dockerfile

Dockerfile

Home24 Dockerfile

Dockerfile

Dockerfile

Home24 Dockerfile

Dockerfile

Dockerfile

Home24 – Dockerfile

Dockerfile

Dockerfile

We migrated to docker

Using docker extensively at home24

I would do the same again

Right decision for current scale

Future opportunities

Introduction	Why to dockerize What is docker How to dockerize
Development	Context/Alternatives What I really liked What I do not like
Production	Deploying to AWS Logging challenges

Tool to run in isolated environment

Many ifs in kernel = cgroups

Not VirtualBox, not unikernel

Open source tool backed by Docker Inc

Container hosting and premium service

Improved by community (AWS, K8s)

Introduction

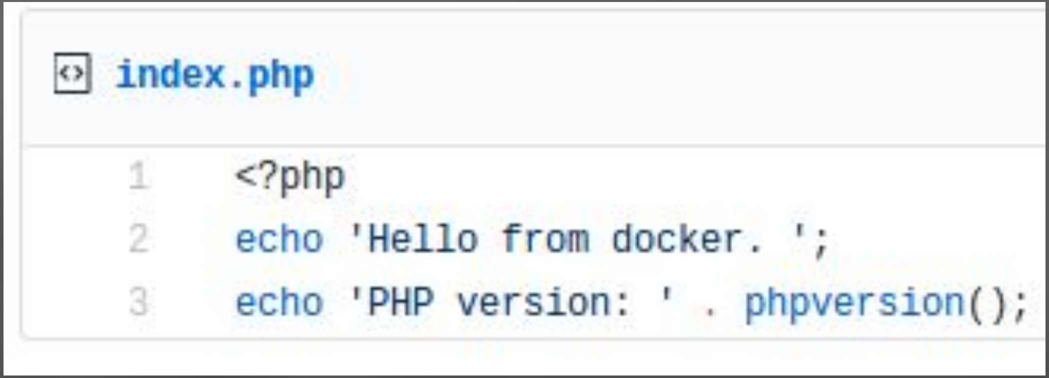
- Why to dockerize
- What is docker
- How to dockerize

Development

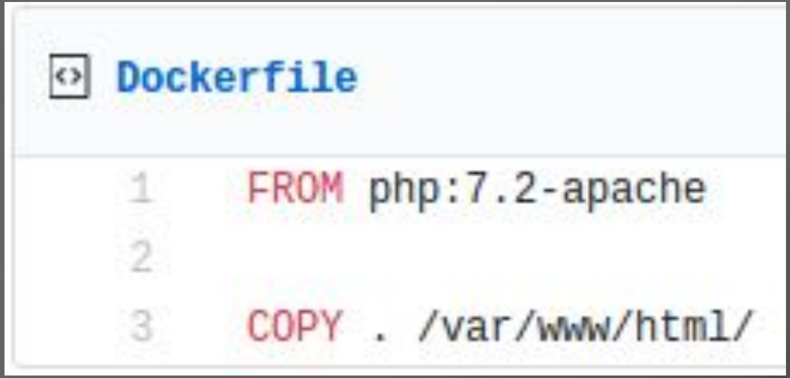
- Context/Alternatives
- What I really liked
- What I do not like

Production

- Deploying to AWS
- Logging challenges

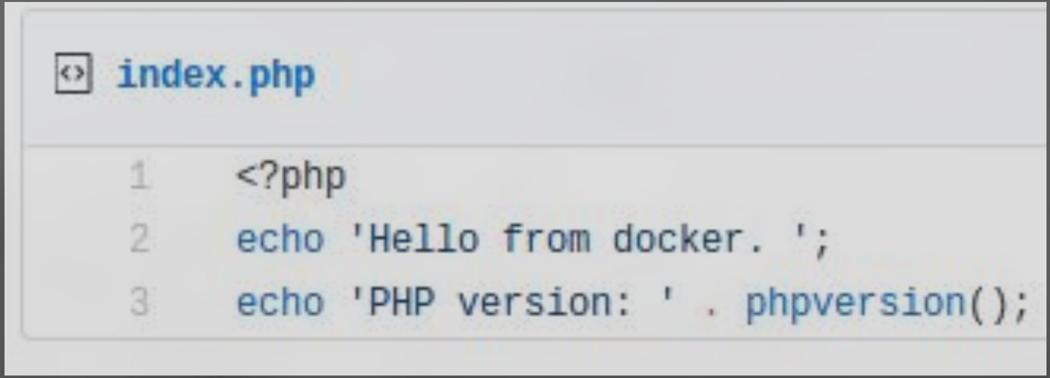

index.php

```php
<?php
echo 'Hello from docker. ';
echo 'PHP version: ' . phpversion();
```



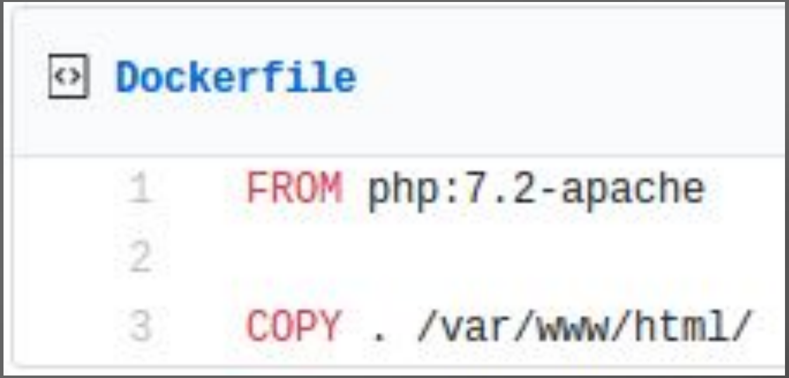
Dockerfile

```
FROM php:7.2-apache

COPY . /var/www/html/
```



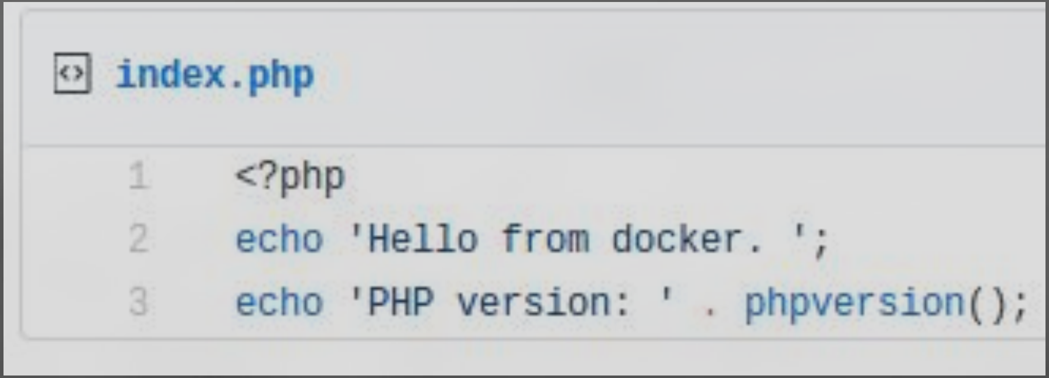
index.php

```
<?php
echo 'Hello from docker. ';
echo 'PHP version: ' . phpversion();
```



Dockerfile

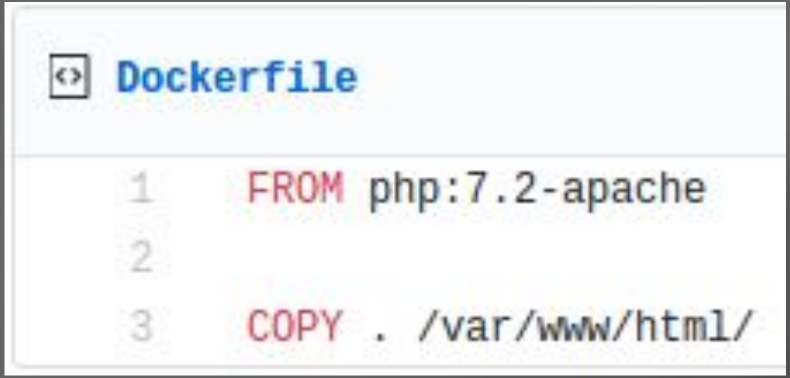
```
FROM php:7.2-apache

COPY . /var/www/html/
```



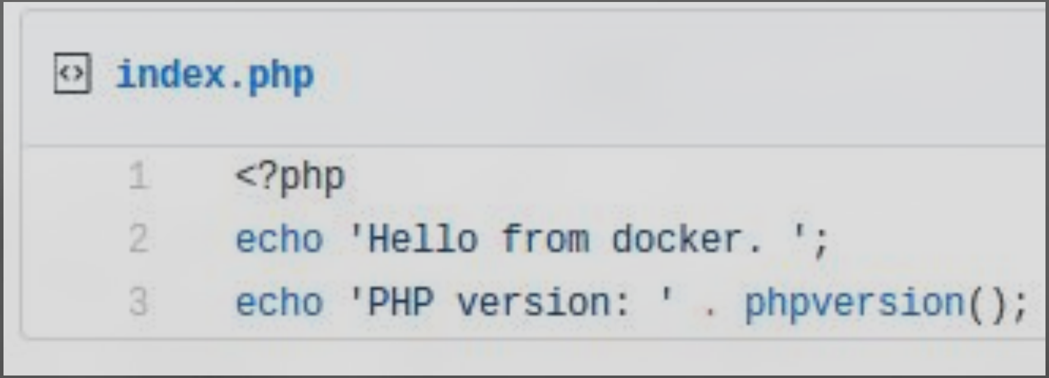
index.php

```
<?php
echo 'Hello from docker. ';
echo 'PHP version: ' . phpversion();
```

```
docker build . -t siauliaiphp-example
docker run -p 8080:80 siauliaiphp-example
```

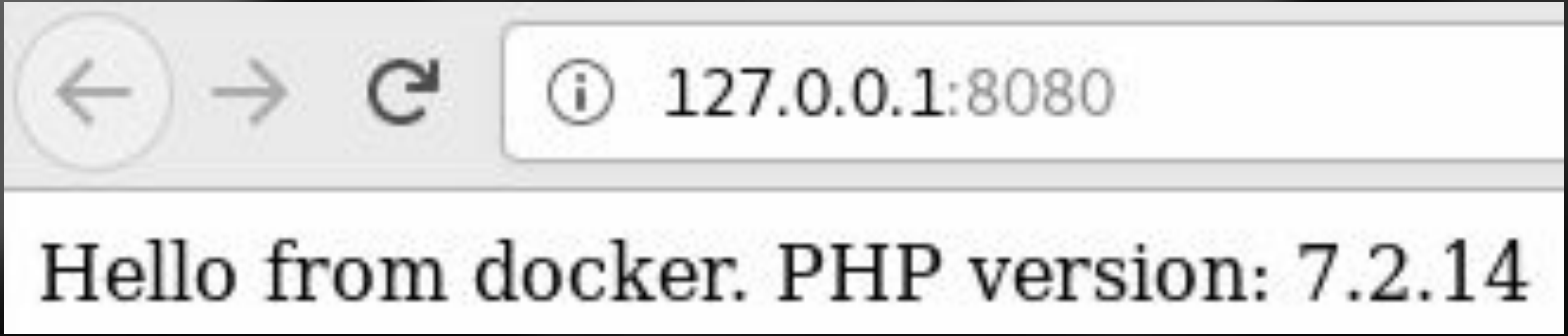

Dockerfile

```
FROM php:7.2-apache

COPY . /var/www/html/
```


index.php

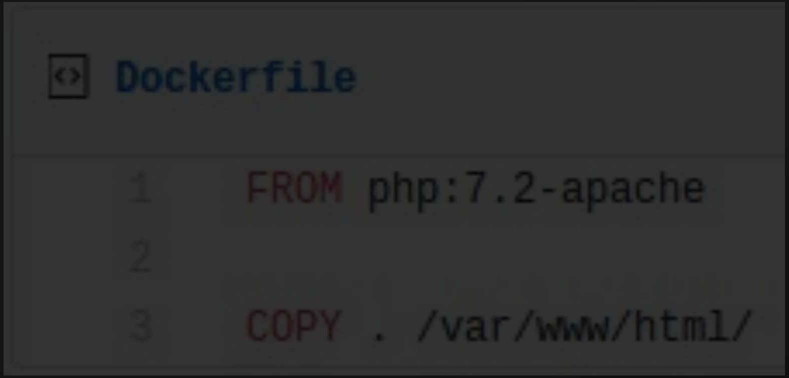
```
<?php
echo 'Hello from docker. ';
echo 'PHP version: ' . phpversion();
```

```
docker build . -t siauliaiphp-example
docker run -p 8080:80 siauliaiphp-example
```

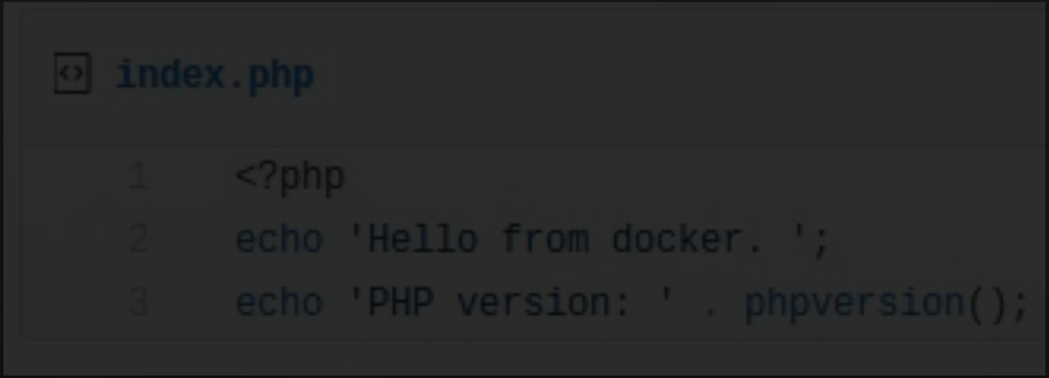

127.0.0.1:8080

Hello from docker. PHP version: 7.2.14

https://gist.github.com/aurelijusb/6f6b1c86c440ce527250193ffcae62d7


Dockerfile

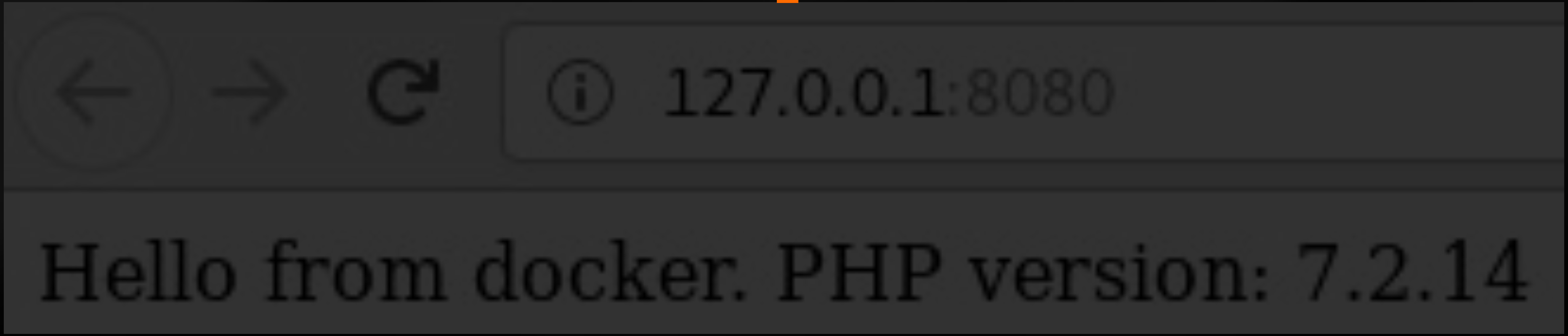
```
FROM php:7.2-apache

COPY . /var/www/html/
```


index.php

```php
<?php
echo 'Hello from docker. ';
echo 'PHP version: ' . phpversion();
```

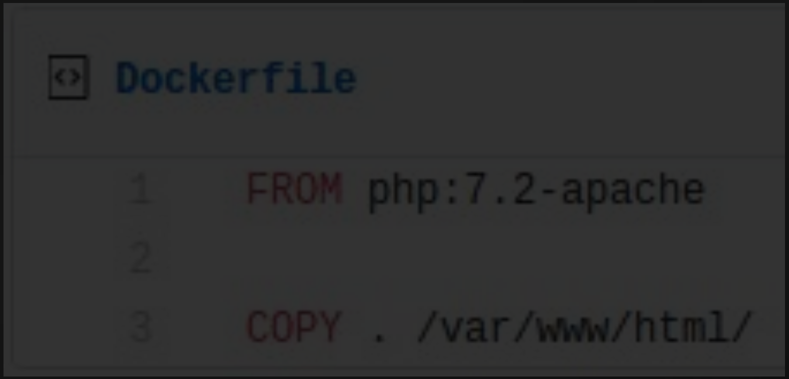
```
docker build . -t siauliaiphp-example
docker run -p 8080:80 siauliaiphp-example
```

Simple?



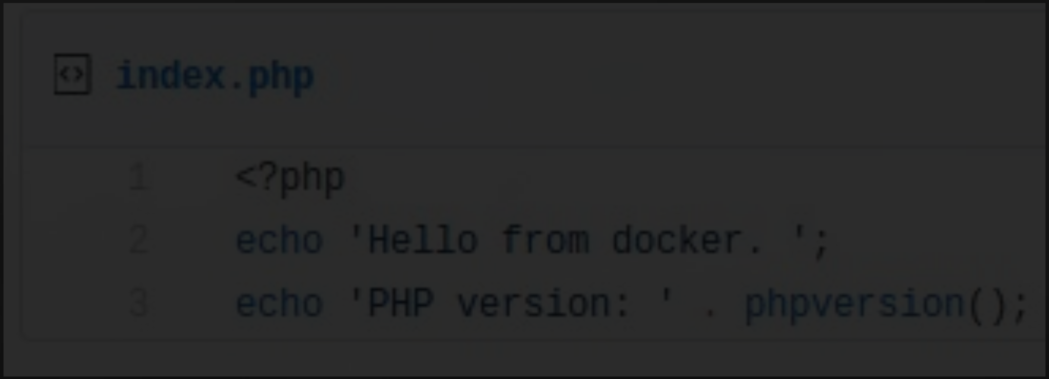
Hello from docker. PHP version: 7.2.14

https://gist.github.com/aurelijusb/6f6b1c86c440ce527250193ffcae62d7



Dockerfile

```
FROM php:7.2-apache

COPY . /var/www/html/
```



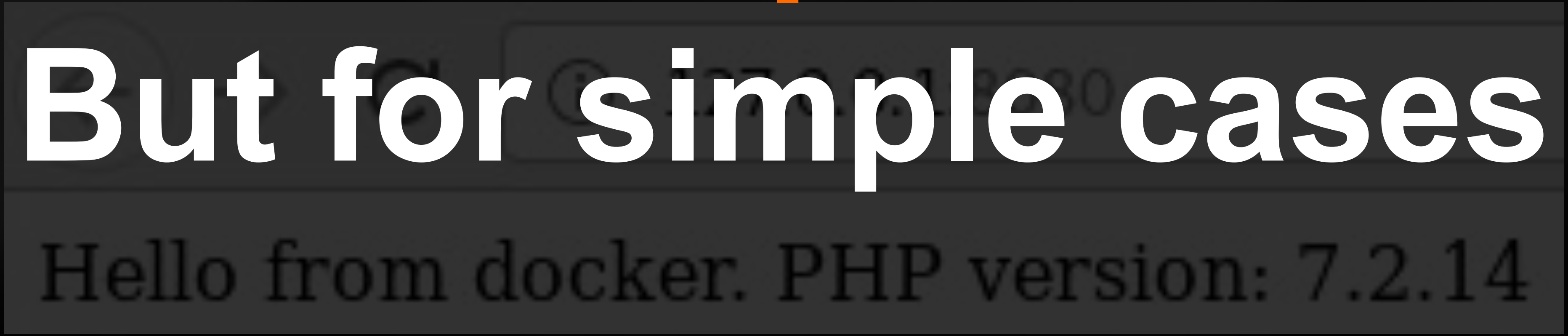
index.php

```php
<?php
echo 'Hello from docker. ';
echo 'PHP version: ' . phpversion();
```

```
docker build . -t siauliaiphp-example
docker run -p 8080:80 siauliaiphp-example
```

Simple?

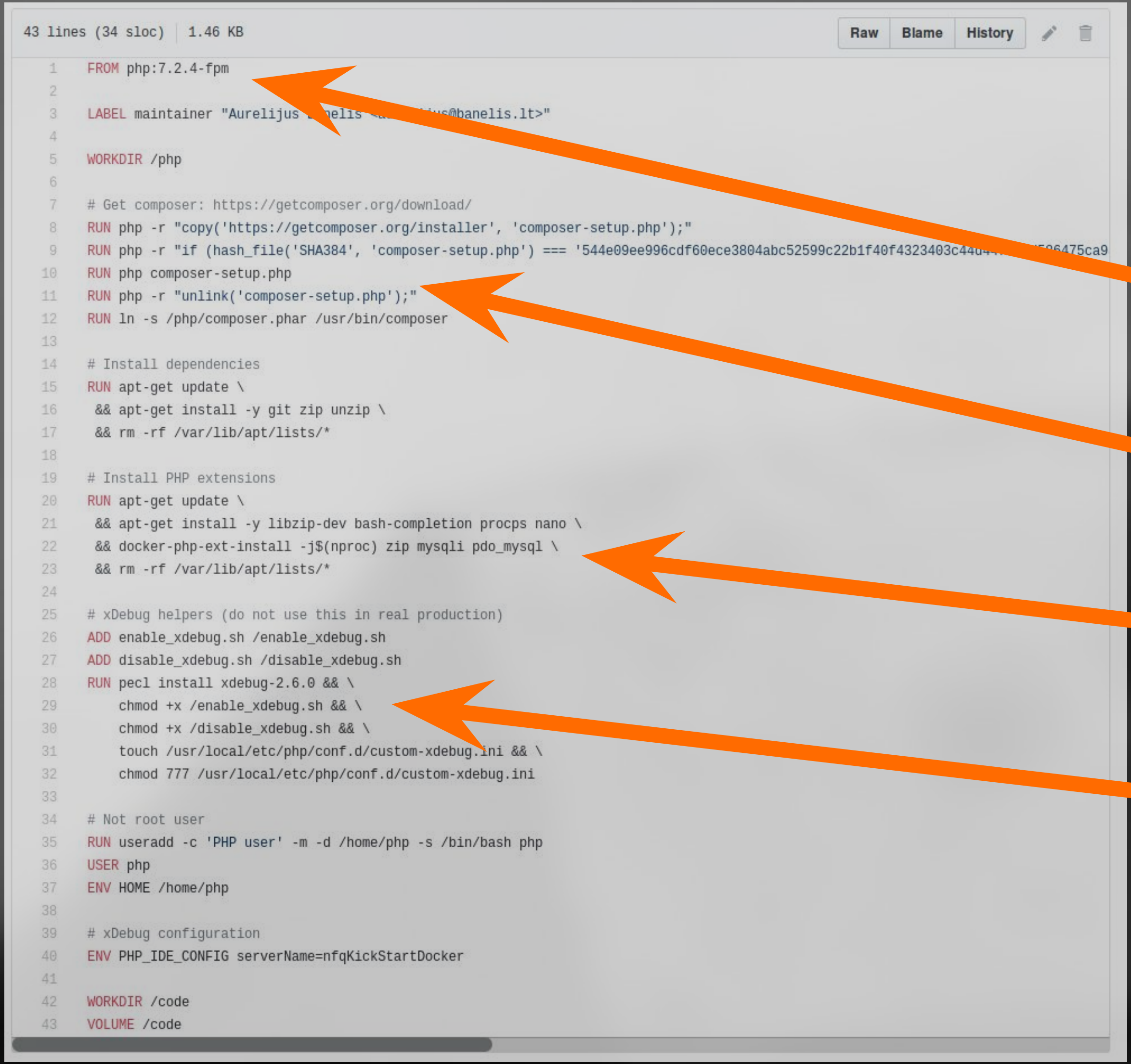
But for simple cases



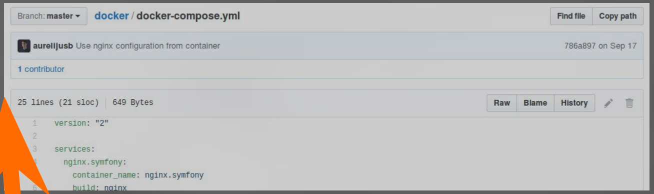
127.0.0.1:8080

Hello from docker. PHP version: 7.2.14

https://gist.github.com/aurelijusb/6f6b1c86c440ce527250193ffcae62d7



```
FROM php:7.2.4-fpm

LABEL maintainer "Aurelijus [illegible]nelis <a[illegible]ius@banelis.lt>"

WORKDIR /php

# Get composer: https://getcomposer.org/download/
RUN php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
RUN php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d4[illegible]f86475ca9
RUN php composer-setup.php
RUN php -r "unlink('composer-setup.php');"
RUN ln -s /php/composer.phar /usr/bin/composer

# Install dependencies
RUN apt-get update \
 && apt-get install -y git zip unzip \
 && rm -rf /var/lib/apt/lists/*

# Install PHP extensions
RUN apt-get update \
 && apt-get install -y libzip-dev bash-completion procps nano \
 && docker-php-ext-install -j$(nproc) zip mysqli pdo_mysql \
 && rm -rf /var/lib/apt/lists/*

# xDebug helpers (do not use this in real production)
ADD enable_xdebug.sh /enable_xdebug.sh
ADD disable_xdebug.sh /disable_xdebug.sh
RUN pecl install xdebug-2.6.0 && \
    chmod +x /enable_xdebug.sh && \
    chmod +x /disable_xdebug.sh && \
    touch /usr/local/etc/php/conf.d/custom-xdebug.ini && \
    chmod 777 /usr/local/etc/php/conf.d/custom-xdebug.ini

# Not root user
RUN useradd -c 'PHP user' -m -d /home/php -s /bin/bash php
USER php
ENV HOME /home/php

# xDebug configuration
ENV PHP_IDE_CONFIG serverName=nfqKickStartDocker

WORKDIR /code
VOLUME /code
```


Branch: master
docker / docker-compose.yml
Find file
Copy path
aurelijusb Use nginx configuration from container
786a897 on Sep 17
1 contributor
25 lines (21 sloc) | 649 Bytes
Raw
Blame
History

```
version: "2"

services:
  nginx.symfony:
    container_name: nginx.symfony
    build: nginx
```

Multiple containers

Development tools

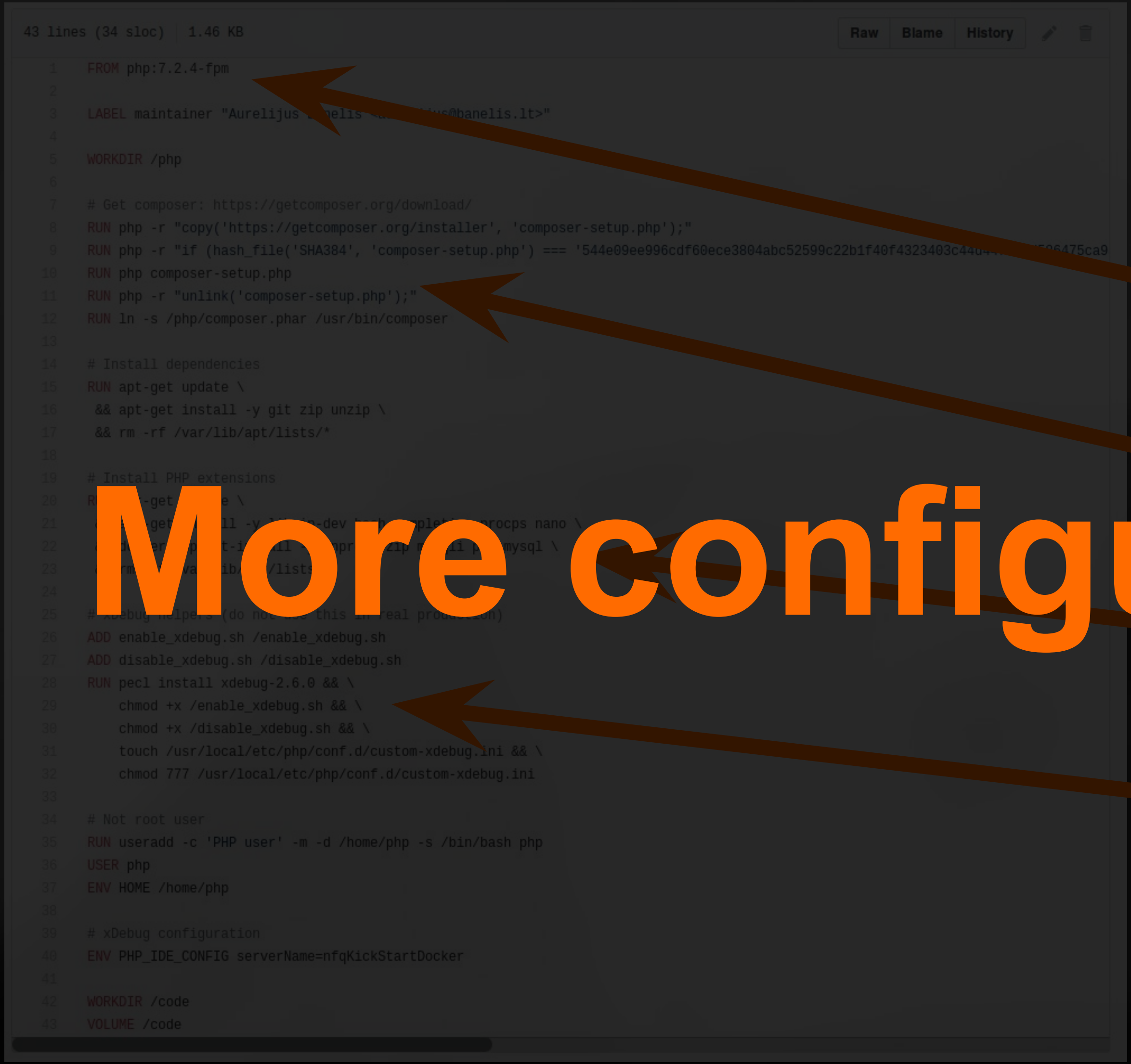
PHP extensions

xDebug support

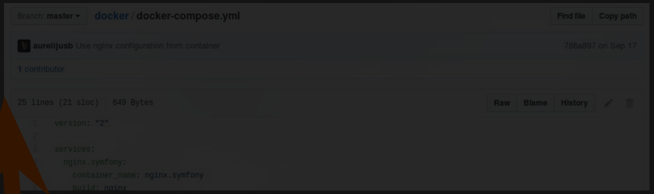
...

https://github.com/nfqakademija/docker/blob/master/php/Dockerfile

More configuration



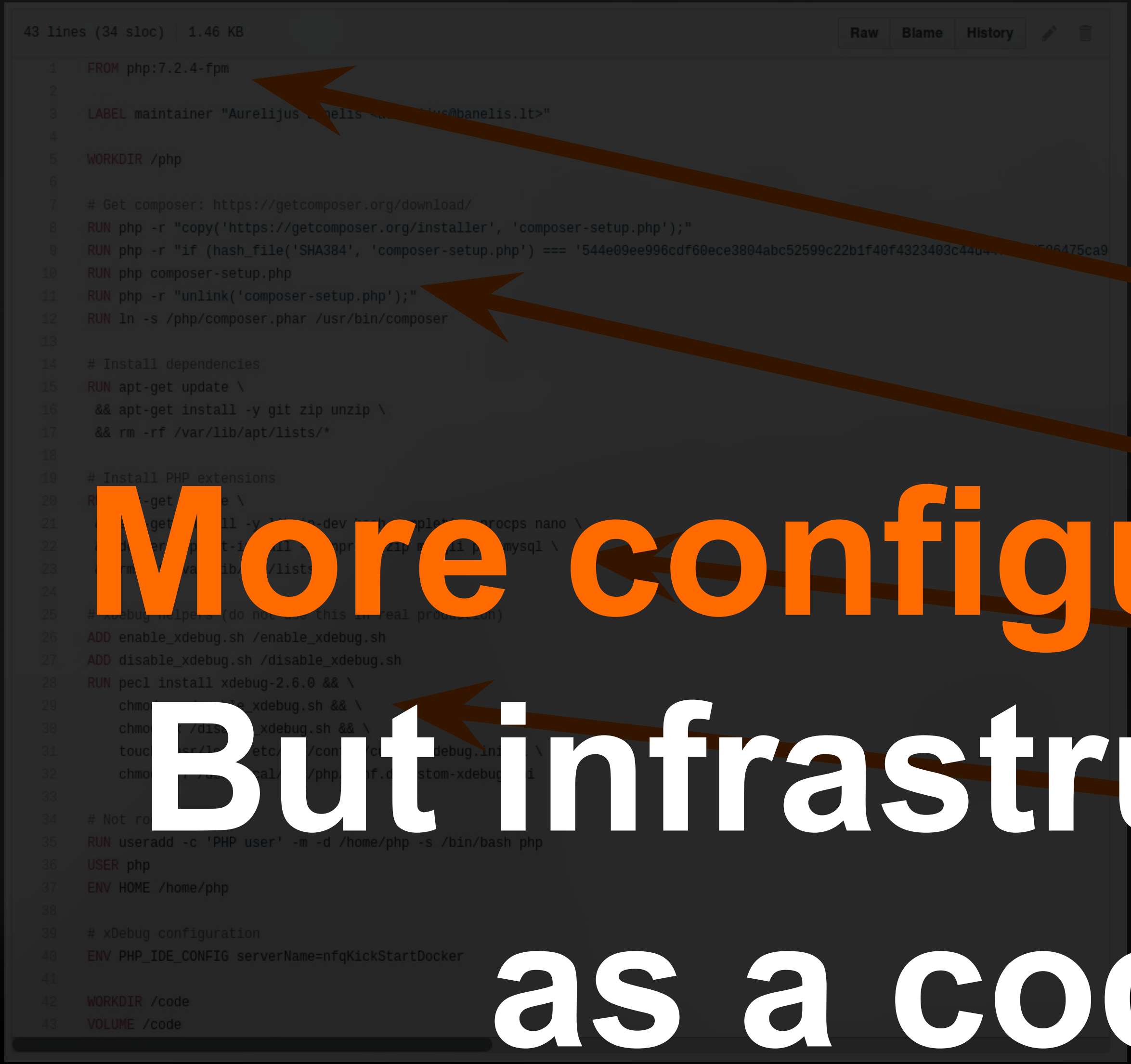
```
FROM php:7.2.4-fpm

LABEL maintainer "Aurelijus [illegible]nelis <a[illegible]jus@banelis.lt>"

WORKDIR /php

# Get composer: https://getcomposer.org/download/
RUN php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
RUN php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c440[illegible]586475ca9
RUN php composer-setup.php
RUN php -r "unlink('composer-setup.php');"
RUN ln -s /php/composer.phar /usr/bin/composer

# Install dependencies
RUN apt-get update \
 && apt-get install -y git zip unzip \
 && rm -rf /var/lib/apt/lists/*

# Install PHP extensions
R[illegible]-get [illegible]e \
 [illegible]-get [illegible]ll -y li[illegible]ip-dev b[illegible]h-[illegible]pleti[illegible] procps nano \
 [illegible] do[illegible]er[illegible]p[illegible]t-i[illegible]ll [illegible]pr[illegible] zip m[illegible]li p[illegible] mysql \
 [illegible] rm [illegible]va[illegible]ib/[illegible]/lists[illegible]

# xdebug helpers (do not use this in real production)
ADD enable_xdebug.sh /enable_xdebug.sh
ADD disable_xdebug.sh /disable_xdebug.sh
RUN pecl install xdebug-2.6.0 && \
    chmod +x /enable_xdebug.sh && \
    chmod +x /disable_xdebug.sh && \
    touch /usr/local/etc/php/conf.d/custom-xdebug.ini && \
    chmod 777 /usr/local/etc/php/conf.d/custom-xdebug.ini

# Not root user
RUN useradd -c 'PHP user' -m -d /home/php -s /bin/bash php
USER php
ENV HOME /home/php

# xDebug configuration
ENV PHP_IDE_CONFIG serverName=nfqKickStartDocker

WORKDIR /code
VOLUME /code
```


Branch: master
docker / docker-compose.yml
Find file
Copy path
aurelijusb Use nginx configuration from container
786a897 on Sep 17
1 contributor
25 lines (21 sloc) | 649 Bytes
Raw
Blame
History

```
version: "2"

services:
  nginx.symfony:
    container_name: nginx.symfony
    build: nginx
```

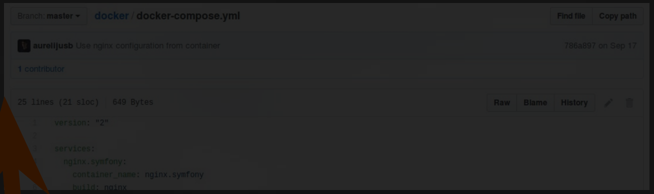
- Multiple containers
- Development tools
- PHP extensions
- xDebug support
- ...

https://github.com/nfqakademija/docker/blob/master/php/Dockerfile



43 lines (34 sloc) | 1.46 KB

```
FROM php:7.2.4-fpm

LABEL maintainer "Aurelijus [illegible]nelis <[illegible]us@banelis.lt>"

WORKDIR /php

# Get composer: https://getcomposer.org/download/
RUN php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
RUN php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d4[illegible]586475ca9
RUN php composer-setup.php
RUN php -r "unlink('composer-setup.php');"
RUN ln -s /php/composer.phar /usr/bin/composer

# Install dependencies
RUN apt-get update \
 && apt-get install -y git zip unzip \
 && rm -rf /var/lib/apt/lists/*

# Install PHP extensions
R[illegible]-get [illegible]e \
 [illegible]-get [illegible]ll -y [illegible]-dev [illegible]procps nano \
 [illegible]er[illegible]t-i[illegible]pr[illegible]zip [illegible]mysql \
 [illegible]rm[illegible]va[illegible]ib[illegible]/lists[illegible]

# [illegible] (do not [illegible] this in real production)
ADD enable_xdebug.sh /enable_xdebug.sh
ADD disable_xdebug.sh /disable_xdebug.sh
RUN pecl install xdebug-2.6.0 && \
    chmo[illegible]_xdebug.sh && \
    chmo[illegible] /dis[illegible]_xdebug.sh && \
    touc[illegible]etc[illegible]/con[illegible]debug.ini [illegible] \
    chmo[illegible]cal[illegible]/php[illegible]f.d[illegible]stom-xdebug[illegible]i

# Not ro[illegible]
RUN useradd -c 'PHP user' -m -d /home/php -s /bin/bash php
USER php
ENV HOME /home/php

# xDebug configuration
ENV PHP_IDE_CONFIG serverName=nfqKickStartDocker

WORKDIR /code
VOLUME /code
```



Branch: master docker / docker-compose.yml

aurelijusb Use nginx configuration from container 786a897 on Sep 17

1 contributor

25 lines (21 sloc) | 649 Bytes

```
version: "2"

services:
  nginx.symfony:
    container_name: nginx.symfony
    build: nginx
```

Multiple containers

Development tools

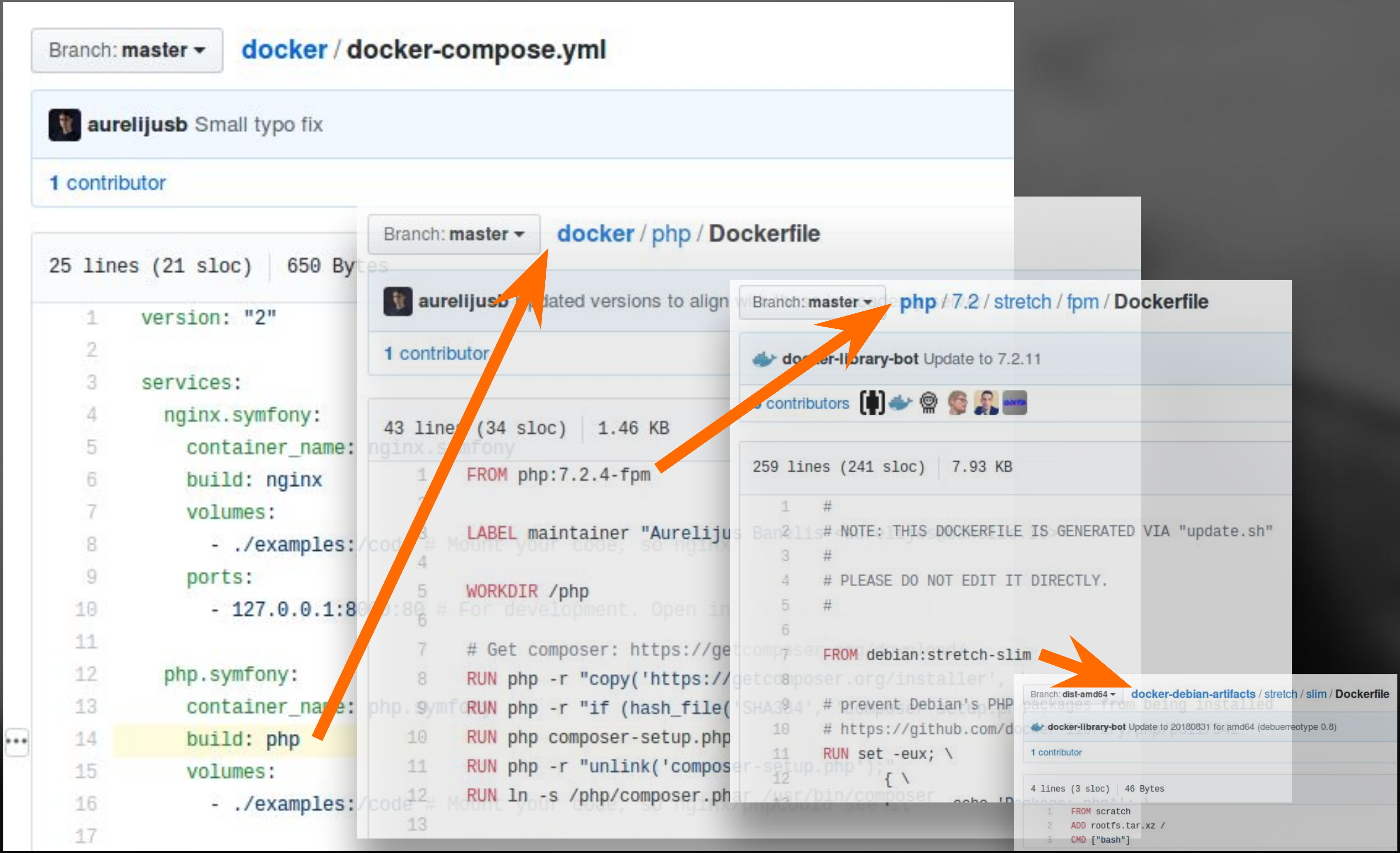
PHP extensions

xDebug support

More configuration

But infrastructure as a code

https://github.com/nfqakademija/docker/blob/master/php/Dockerfile



Branch: master docker / docker-compose.yml

aurelijusb Small typo fix

1 contributor

25 lines (21 sloc) | 650 Bytes

```
version: "2"

services:
  nginx.symfony:
    container_name:
    build: nginx
    volumes:
      - ./examples:
    ports:
      - 127.0.0.1:8
  
  php.symfony:
    container_name:
    build: php
    volumes:
      - ./examples:
```

Branch: master docker / php / Dockerfile

aurelijusb Updated versions to align

1 contributor

43 lines (34 sloc) | 1.46 KB

```
FROM php:7.2.4-fpm

LABEL maintainer "Aurelijus

WORKDIR /php

# Get composer: https://ge
RUN php -r "copy('https://
RUN php -r "if (hash_file(
RUN php composer-setup.php
RUN php -r "unlink('compos
RUN ln -s /php/composer.ph
```

Branch: master php / 7.2 / stretch / fpm / Dockerfile

docker-library-bot Update to 7.2.11

contributors

259 lines (241 sloc) | 7.93 KB

```
#
# NOTE: THIS DOCKERFILE IS GENERATED VIA "update.sh"
#
# PLEASE DO NOT EDIT IT DIRECTLY.
#

FROM debian:stretch-slim

# prevent Debian's PHP
# https://github.com/d
RUN set -eux; \
    { \
```

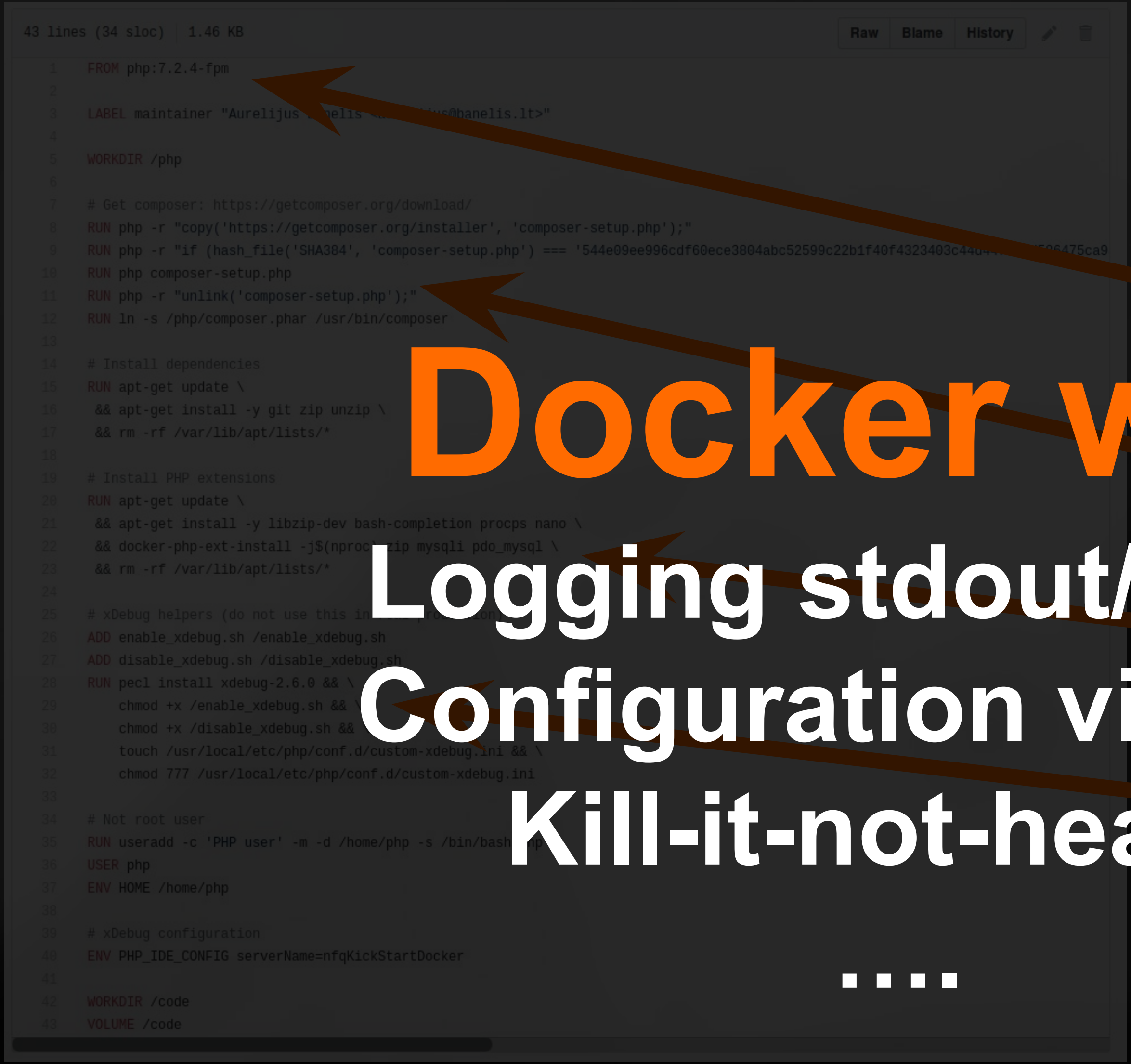
Branch: dist-amd64 docker-debian-artifacts / stretch / slim / Dockerfile

docker-library-bot Update to 20180831 for amd64 (debuerreotype 0.8)

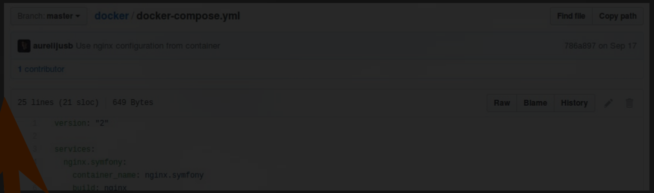
1 contributor

4 lines (3 sloc) | 46 Bytes

```
FROM scratch
ADD rootfs.tar.xz /
CMD ["bash"]
```


43 lines (34 sloc) | 1.46 KB

```
FROM php:7.2.4-fpm

LABEL maintainer "Aurelijus Banelis <aurelijus@banelis.lt>"

WORKDIR /php

# Get composer: https://getcomposer.org/download/
RUN php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
RUN php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d4[illegible]586475ca9
RUN php composer-setup.php
RUN php -r "unlink('composer-setup.php');"
RUN ln -s /php/composer.phar /usr/bin/composer

# Install dependencies
RUN apt-get update \
 && apt-get install -y git zip unzip \
 && rm -rf /var/lib/apt/lists/*

# Install PHP extensions
RUN apt-get update \
 && apt-get install -y libzip-dev bash-completion procps nano \
 && docker-php-ext-install -j$(nproc) zip mysqli pdo_mysql \
 && rm -rf /var/lib/apt/lists/*

# xDebug helpers (do not use this in [illegible]
ADD enable_xdebug.sh /enable_xdebug.sh
ADD disable_xdebug.sh /disable_xdebug.sh
RUN pecl install xdebug-2.6.0 && \
    chmod +x /enable_xdebug.sh && \
    chmod +x /disable_xdebug.sh && \
    touch /usr/local/etc/php/conf.d/custom-xdebug.ini && \
    chmod 777 /usr/local/etc/php/conf.d/custom-xdebug.ini

# Not root user
RUN useradd -c 'PHP user' -m -d /home/php -s /bin/bash php
USER php
ENV HOME /home/php

# xDebug configuration
ENV PHP_IDE_CONFIG serverName=nfqKickStartDocker

WORKDIR /code
VOLUME /code
```


Branch: master | docker / docker-compose.yml | Find file | Copy path

aurelijusb Use nginx configuration from container — 786a897 on Sep 17

1 contributor

25 lines (21 sloc) | 649 Bytes

```
version: "2"

services:
  nginx.symfony:
    container_name: nginx.symfony
    build: nginx
```

Multiple containers

Development tools

PHP extensions

xDebug support

...

Docker way

Logging stdout/stderr

Configuration via ENV

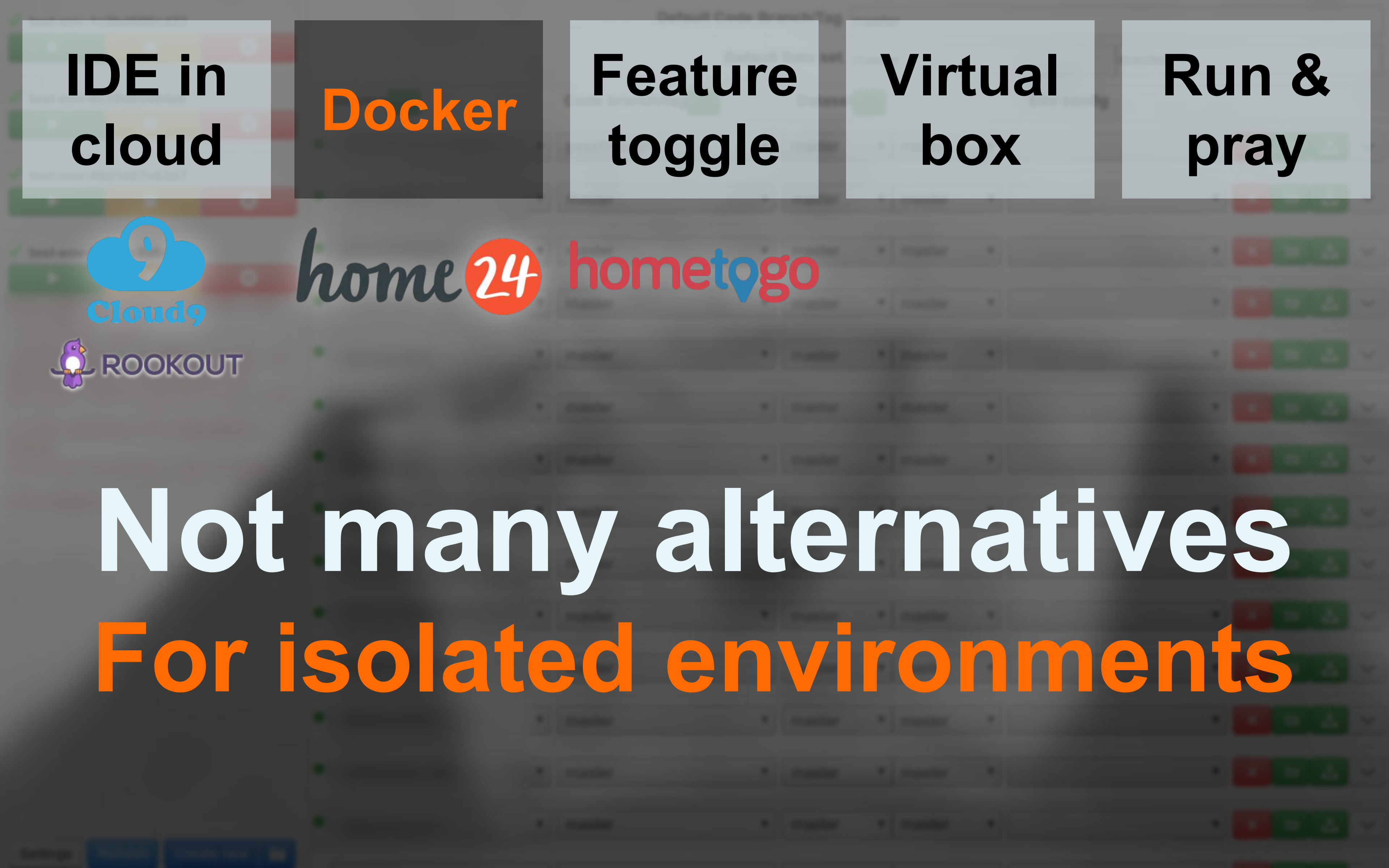
Kill-it-not-heal-it

....

https://github.com/nfqakademija/docker/blob/master/php/Dockerfile

Introduction	Why to dockerize What is docker How to dockerize
Development	Context/Alternatives What I really liked What I do not like
Production	Deploying to AWS Logging challenges

Introduction	Why to dockerize What is docker How to dockerize
Development	**Context/Alternatives** What I really liked What I do not like
Production	Deploying to AWS Logging challenges



IDE in cloud

Docker

Feature toggle

Virtual box

Run & pray

Cloud9

ROOKOUT

home24

hometogo

Not many alternatives

For isolated environments

Introduction

Why to dockerize
What is docker
How to dockerize

Development

Context/Alternatives
What I really liked
What I do not like

Production

Deploying to AWS
Logging challenges

Person

Community

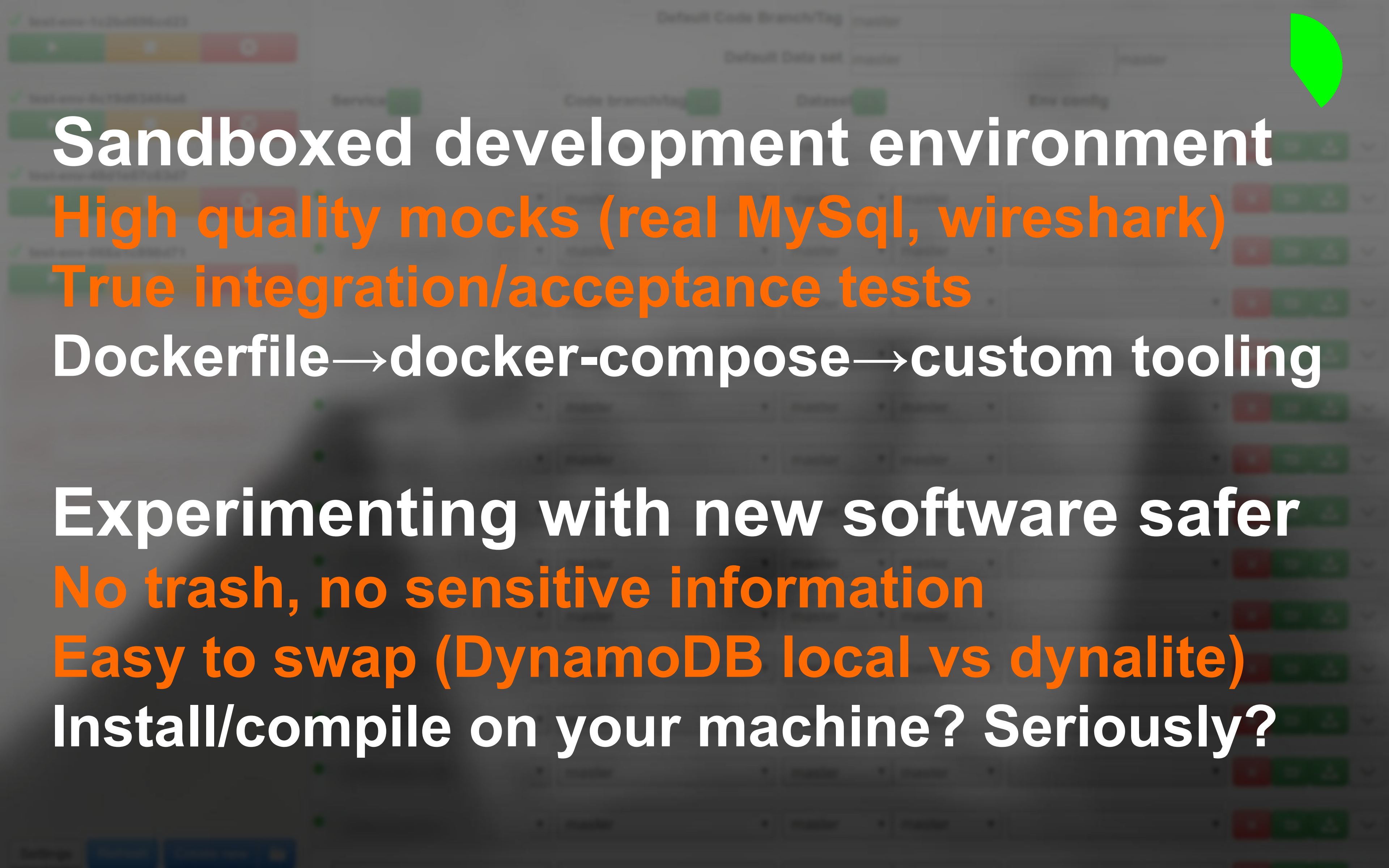
Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.

How Amazon ECS works

Amazon Elastic Container Registry
Amazon Elastic Container Service
Define your application
Amazon EC2
AWS Fargate
Manage containers

Team



Sandboxed development environment

High quality mocks (real MySql, wireshark)

True integration/acceptance tests

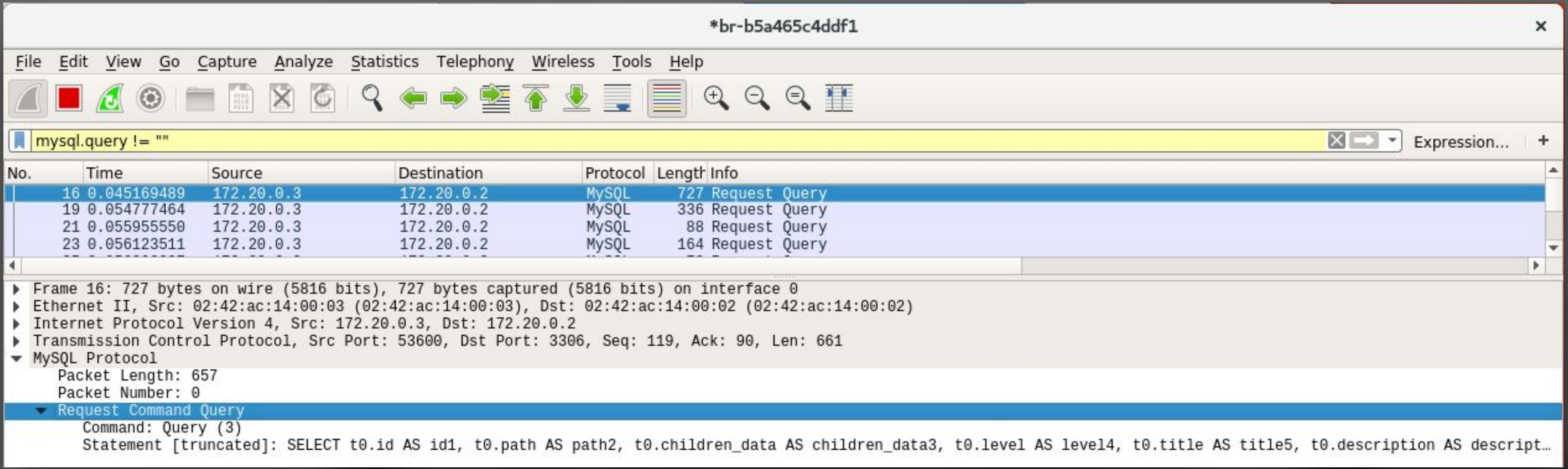
Dockerfile→docker-compose→custom tooling

Experimenting with new software safer

No trash, no sensitive information

Easy to swap (DynamoDB local vs dynalite)

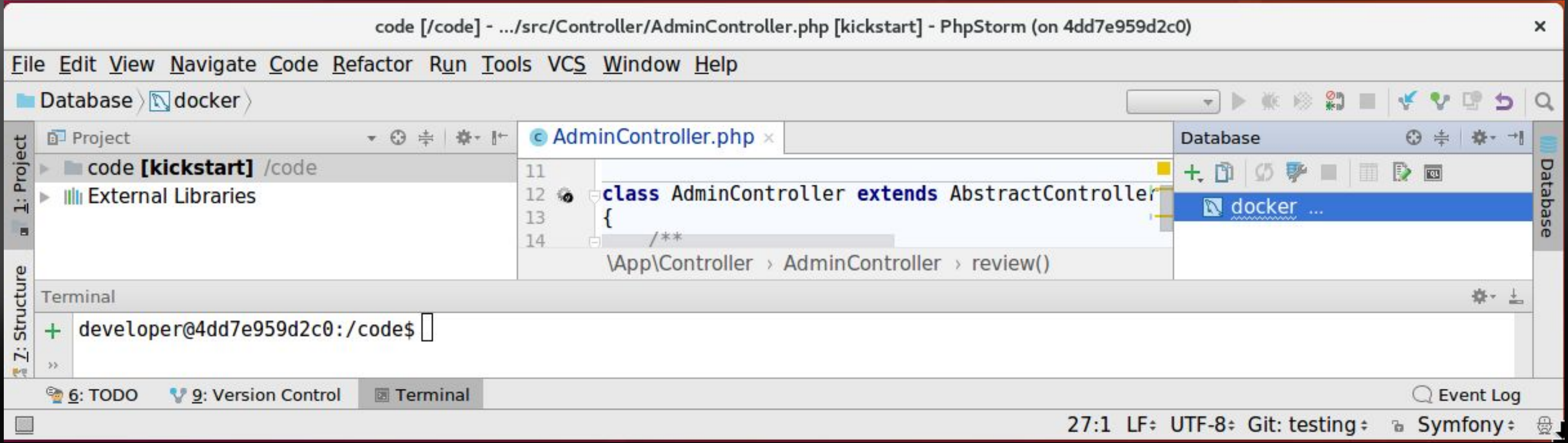
Install/compile on your machine? Seriously?


*br-b5a465c4ddf1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
mysql.query != ""
```

```
No.   Time         Source       Destination  Protocol  Length Info
   16 0.045169489  172.20.0.3   172.20.0.2   MySQL     727 Request Query
   19 0.054777464  172.20.0.3   172.20.0.2   MySQL     336 Request Query
   21 0.055955550  172.20.0.3   172.20.0.2   MySQL      88 Request Query
   23 0.056123511  172.20.0.3   172.20.0.2   MySQL     164 Request Query
```

```
▸ Frame 16: 727 bytes on wire (5816 bits), 727 bytes captured (5816 bits) on interface 0
▸ Ethernet II, Src: 02:42:ac:14:00:03 (02:42:ac:14:00:03), Dst: 02:42:ac:14:00:02 (02:42:ac:14:00:02)
▸ Internet Protocol Version 4, Src: 172.20.0.3, Dst: 172.20.0.2
▸ Transmission Control Protocol, Src Port: 53600, Dst Port: 3306, Seq: 119, Ack: 90, Len: 661
▾ MySQL Protocol
     Packet Length: 657
     Packet Number: 0
   ▾ Request Command Query
        Command: Query (3)
        Statement [truncated]: SELECT t0.id AS id1, t0.path AS path2, t0.children_data AS children_data3, t0.level AS level4, t0.title AS title5, t0.description AS descript…
```


code [/code] - .../src/Controller/AdminController.php [kickstart] - PhpStorm (on 4dd7e959d2c0)
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Database
docker
Project
code [kickstart] /code
External Libraries
AdminController.php
Database
docker ...
1: Project
7: Structure
Terminal
6: TODO
9: Version Control
Terminal
Event Log
27:1 LF÷ UTF-8÷ Git: testing÷ Symfony÷

```php
class AdminController extends AbstractController
{
    /**
```

\App\Controller › AdminController › review()

```
developer@4dd7e959d2c0:/code$
```

Switching between branches/tasks

Less issues with cache invalidation

Kill-it-not-heal-it

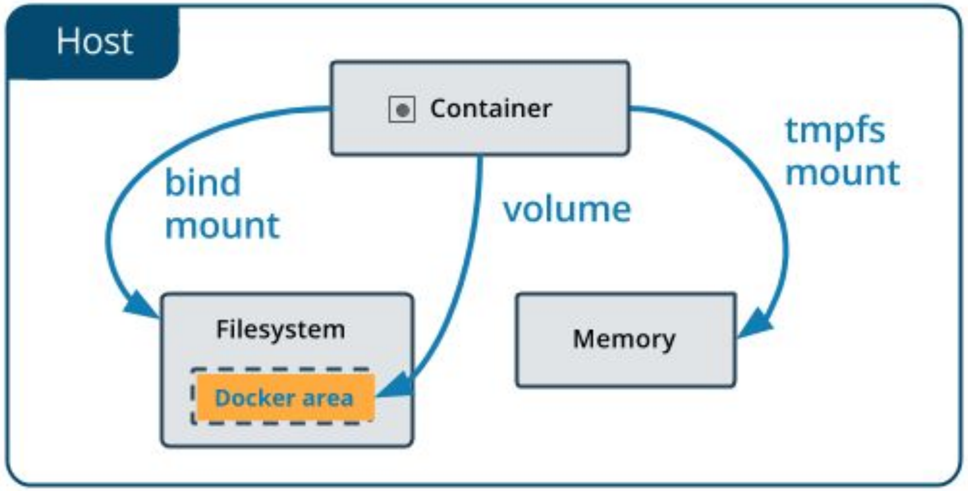
Data volumes = test data in branch

Infrastructure as a code

Fixed versions, fixed php.ini+extensions

Less “works on my machine” = reproducible

Many bash scripts, configuration via ENV



```
ide.symfony:
  build: ide
  container_name: ide.symfony
  environment:
    DISPLAY: $DISPLAY
    SSH_AUTH_SOCK: $SSH_AUTH_SOCK
  volumes:
    - /tmp/.X11-unix:/tmp/.X11-unix
    - ./examples:/code
    - ./.docker/ide-home:/home/developer # Ensure directory is writable, otherwise there will be code.path errors from PHPStorm
  ports:
    - 127.0.0.1:9000:9000
  entrypoint: "/ide/bin/phpstorm.sh"
```

Huge ecosystem

Pioniers are using and improving

Amazon, Google (K8s) is investing

Known issues and solutions in StackOverflow

Introduction

Why to dockerize
What is docker
How to dockerize

Development

Context/Alternatives
What I really liked
What I do not like

Production

Deploying to AWS
Logging challenges

High learning curve = blame docker

Misuse of the tool – isn't docker *silver bullet*?

Docker wraps – everyone blames the wrapper

Many ways to install docker incorrectly

Mounting vs copy-on-write operation

No Windows, more tooling/docs around

Projekto paleidimas

Pasileidžiant pirmą kartą būdavo įveliama daug klaidų, todėl padaryti *script'ai* dažniausiems atvejams.

- Pasileidžiama infrastruktūrą per `docker` į:

```
scripts/start.sh
```

- Įsidiegiame PHP ir JavaScript bibliotekas:

```
scripts/install-prod.sh
```

- Pasižiūrime, ar veikia. Naršyklėje atidarius `http://127.0.0.1:8000/` turėtų rašyti `NFQ Akademija
- Pabaigus, gražiai išjungiame:

```
scripts/stop.sh
```

Patogiai darbo aplinkai

- *Development* režimas (detalesnė informacija apie klaidas, automatiškai generuojami JavaScript/CSS):

```
scripts/install-dev.sh
```

Not mature tooling / edge cases

Download private dependency: github token?

Password protected SSH key? (Mac+Linux)

CircleCI limited “remote docker”

Host IP for xDebug

Even more bash scripts/docs

Mounting Folders

It is **not** possible to mount a folder from your job space into a container in Remote Docker (and vice versa). You may use the `docker cp` command to transfer files between these two environments. For example, to start a container in Remote Docker using a config file from your source code:

```
- run: |
    # create a dummy container which will hold a volume with config
    docker create -v /cfg --name configs alpine:3.4 /bin/true
    # copy a config file into this volume
    docker cp path/in/your/source/code/app_config.yml configs:/cfg
    # start an application container using this volume
    docker run --volumes-from configs app-image:1.2.3
```

In the same way, if your application produces some artifacts that need to be stored, you can copy them from Remote Docker:

```
- run: |
    # start container with the application
    # make sure you're not using `--rm` option otherwise the container will be killed after
    docker run --name app app-image:1.2.3

- run: |
    # after application container finishes, copy artifacts directly from it
    docker cp app:/output /path/in/your/job/space
```

Cache everything by design

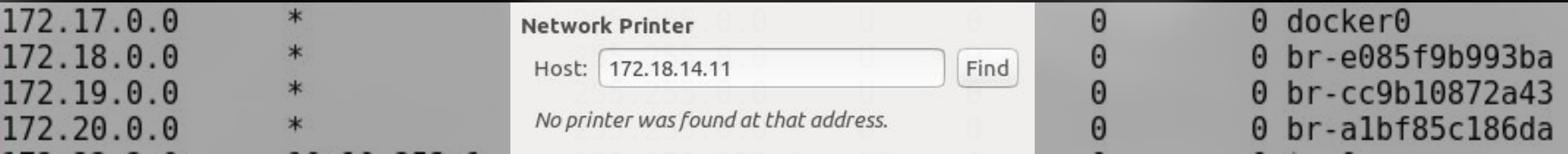
Full HDD because of docker images

Missing log rotation (no “log-opts” by default)

“Latest” tag, that is not immutable

Network unreachable, since used by docker

Docker-clean sh, docker pull, RTFM



```
172.17.0.0      *                        0         0 docker0
172.18.0.0      *                        0         0 br-e085f9b993ba
172.19.0.0      *                        0         0 br-cc9b10872a43
172.20.0.0      *                        0         0 br-a1bf85c186da
```

Network Printer

Host: 172.18.14.11 Find

No printer was found at that address.

Introduction

Why to dockerize
What is docker
How to dockerize

Development

Context/Alternatives
What I really liked
What I do not like

Production

Deploying to AWS
Logging challenges

Introduction

- Why to dockerize
- What is docker
- How to dockerize

Development

- Context/Alternatives
- What I really liked
- What I do not like

Production

- Deploying to AWS
- Logging challenges

PHP + Docker in production

Migrated Tier1 service – nobody noticed

AWS ECS

PHP 7.2

Docker

Apache

PHP + Docker in production

Migrated Tier1 service – nobody noticed

AWS ECS

PHP 7.2

Docker

Apache

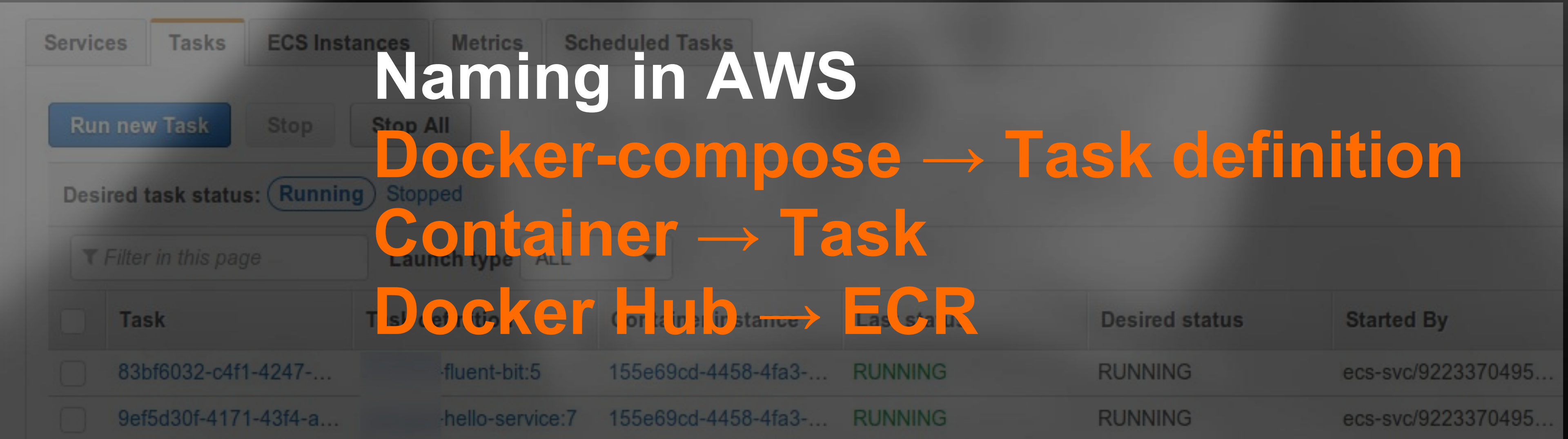
Amazon Web Services Elastic Container Service

AWS integration (E.g. spot-instances)

K8s on AWS then was not mature

Know-how (our Go apps in prod)

Deployment Id	Status	Desired count	Pending count	Running count	Created time	Updated time
ecs-svc/9223370495346757540	PRIMARY	1	0	0	2018-11-06 14:40:18 +0200	2018-11-06 14:40:18 +0200
ecs-svc/9223370495882532443	ACTIVE	[illegible]	[illegible]	[illegible]	2018-10-31 19:50:43 +0200	2018-11-06 14:40:18 +0200



Amazon Web Services Elastic Container Service

- AWS integration (E.g. spot-instances)
- K8s on AWS then was not mature
- Know-how (our Go apps in prod)

Naming in AWS

- Docker-compose → Task definition
- Container → Task
- Docker Hub → ECR

Apache mod_php

PHP 7.2 log trimming in Nginx+FPM

Single process container – easier

No visible performance impact

Internal service (no slow loris attack)

Introduction

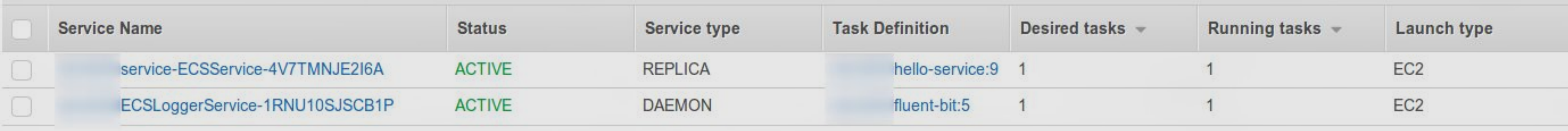
- Why to dockerize
- What is docker
- How to dockerize

Development

- Context/Alternatives
- What I really liked
- What I do not like

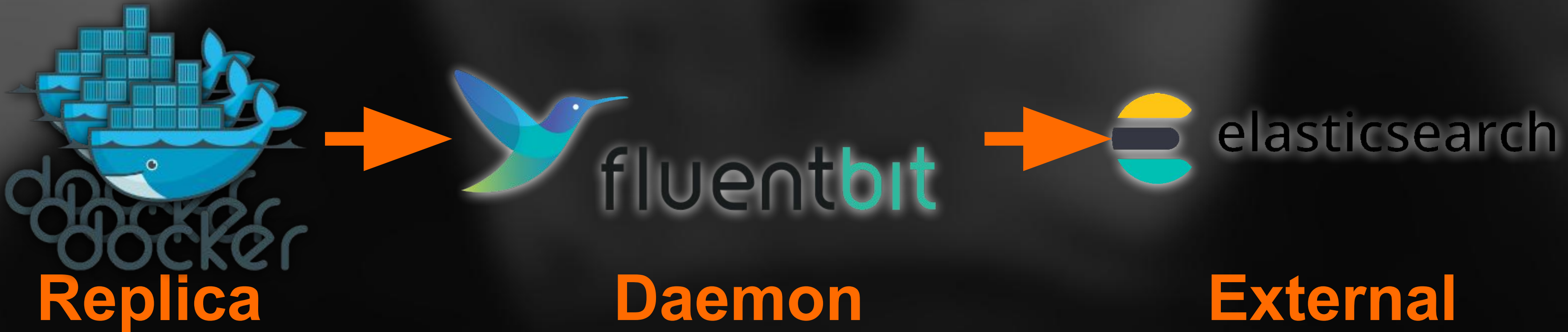
Production

- Deploying to AWS
- Logging challenges



	Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type
☐	[redacted]service-ECSService-4V7TMNJE2I6A	ACTIVE	REPLICA	[redacted]hello-service:9	1	1	EC2
☐	[redacted]ECSLoggerService-1RNU10SJSCB1P	ACTIVE	DAEMON	[redacted]fluent-bit:5	1	1	EC2

ECS Service types

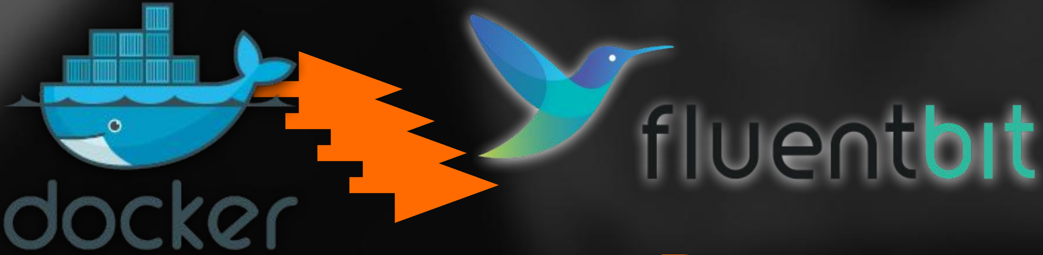


docker

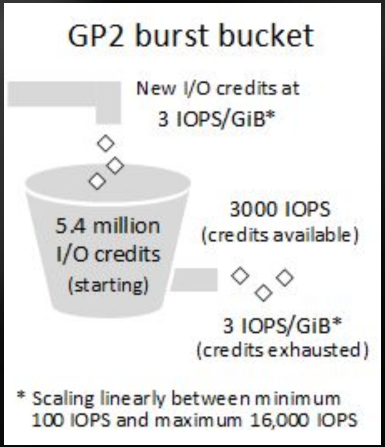
fluentbit

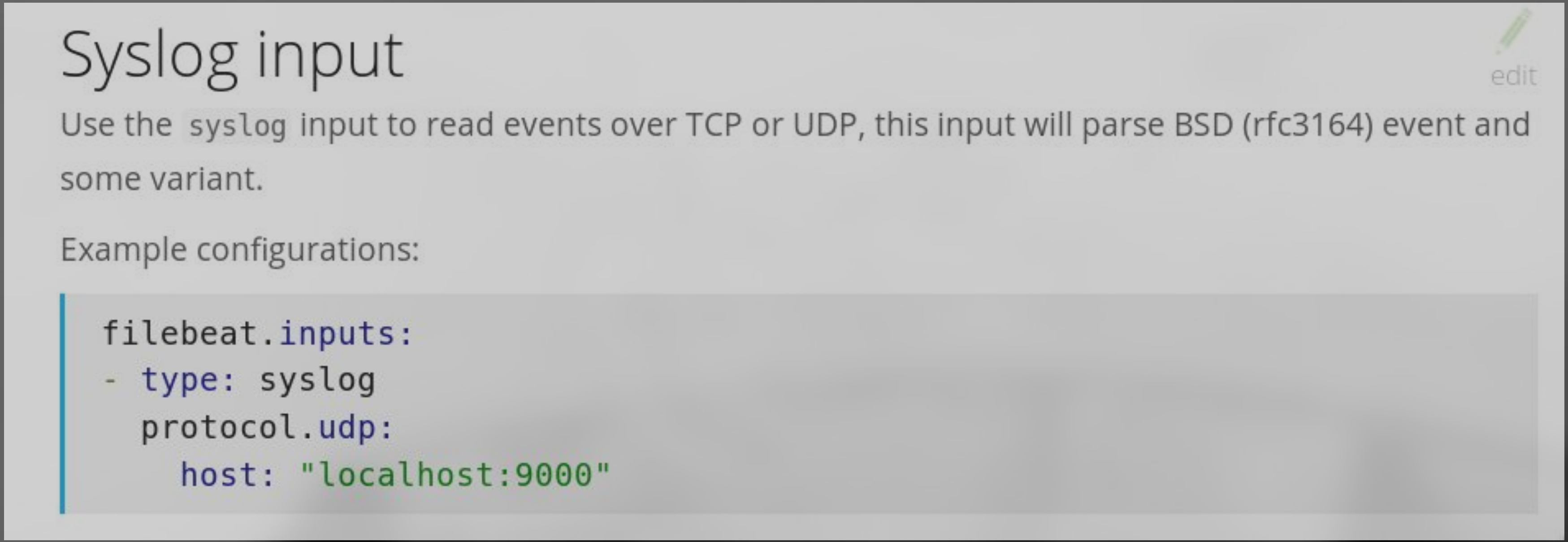
Daemon

```
LogConfiguration:
 LogDriver: fluentd
 Options:
   fluentd-async-connect: "true"
   fluentd-buffer-limit: "32MB"
   mode: "non-blocking"
```



Daemon





Syslog input

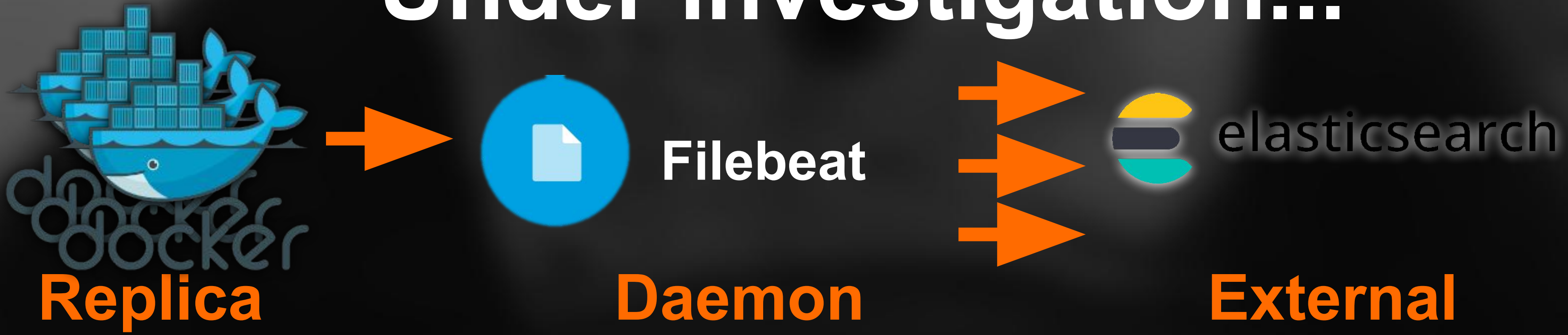
edit

Use the `syslog` input to read events over TCP or UDP, this input will parse BSD (rfc3164) event and some variant.

Example configurations:

```
filebeat.inputs:
- type: syslog
  protocol.udp:
    host: "localhost:9000"
```

Under investigation...



Introduction	Why to dockerize What is docker How to dockerize
Development	Context/Alternatives What I really liked What I do not like
Production	Deploying to AWS Logging challenges

Feature toggle | Docker | IDE in cloud | Virtual box | Run & pray

Docker:

Good to understand

Use on demand

Dockerizing PHP Apps

Thank you

Questions?

Aurelijus Banelis

ŠiauliaiPHP
2019-01-31

Further reading/references

- https://www.docker.com/
- https://hub.docker.com/
- https://aws.amazon.com/ecs/
- https://aws.amazon.com/ecr/
- https://fluentbit.io/
- https://www.home24.de/
- https://home24.tech.blog/category/aws/
- https://aws.amazon.com/blogs/opensource/network-load-balancer-support-in-kubernetes-1-9/
- https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-volumes.html
- https://d1.awsstatic.com/whitepapers/microservices-on-aws.pdf
- https://justi.cz/security/2019/01/22/apt-rce.html
- https://www.elastic.co/guide/en/beats/filebeat/master/filebeat-input-syslog.html
- https://d1.awsstatic.com/whitepapers/DevOps/running-containerized-microservices-on-aws.pdf