

Kickass Development Environments with Docker

...

PHP Dorset

~

david@rawkode.com

@rawkode

github.com/rawkode

Software Consultant

Docker Glasgow Organiser

- PHP / Go / Elixir / Java / Obj-C
 - DevOps / CI / CD / Docker
 - CQRS & Event Sourcing
 - Domain-Driven Design
 - Framework agnostic code
 - TDD / BDD
-

let's travel through time ...

Development Environments circa 2000

```
$ tree
```

```
awesome-million-pound-project
```

```
├── src
│   ├── game.php
│   ├── game.php.bk-david
│   ├── game.php.bk-deano
│   ├── main.php
│   ├── main.php.maybe-fixed
│   ├── main.php.bk-1999-12-02
│   ├── main.php.bk-1999-12-02.2
│   ├── player.php
│   ├── player.php.orig
│   └── .swp.player.php
```

Production Environments circa 2000

```
$ tree
```

```
awesome-million-pound-project
```

```
├── src
```

```
│   ├── game.php
```

```
│   ├── game.php.bk-david
```

```
│   ├── game.php.bk-deano
```

```
│   ├── main.php
```

```
│   ├── main.php.maybe-fixed
```

```
│   ├── main.php.bk-1999-12-02
```

```
│   ├── main.php.bk-1999-12-02.2
```

```
│   ├── player.php
```

```
│   ├── player.php.orig
```

```
│   └── .swp.player.php
```

things got better ...

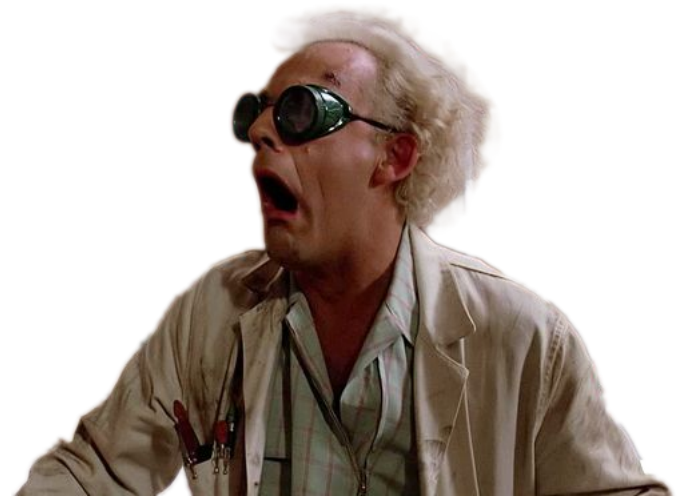
but not for another six years ...

**how did we survive those six years, you
ask?**



Let travel through time a little quicker ...

2006



**Dev/Prod parity becomes a twinkle in our
eye**

2009



CHEF™

DSL Hell



2011





Vagrant Problems

- You're creating and managing VMs for each project / service
- Those VM's are mutable / prone to error by user changes
- They're built JIT (most cases)
- How long goes your vagrant up take?

Let's check out another option

What is Docker?

...

What is Docker?

Docker allows you to package an application with all of its dependencies into a standardized unit for software development.

docker.com

What's the goal?

A single, runnable, distributable executable

The Docker Family

- Docker
- Docker Engine
- Docker Registry
- Docker Compose
- Docker Machine
- Docker Swarm

Docker Image

Layering

COPY ON WRITE

Official Images

<http://hub.docker.com>

- php (5.5, 5.6, 7.0)
- redis
- memcached
- elasticsearch
- mysql
- postgres
- mongo
- node
- rabbitmq
- ...

FROM php:7.0

```
ADD ./ /var/www
```

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

Warning: Be careful with ADD

WORKDIR */var/www*

ENTRYPOINT ["php"]

CMD [“-v”]

Docker Container

Demo



**CHALLENGES
AHEAD**

**Do I have to
rebuild
my container
when I update my
code?**



Docker Volumes

...

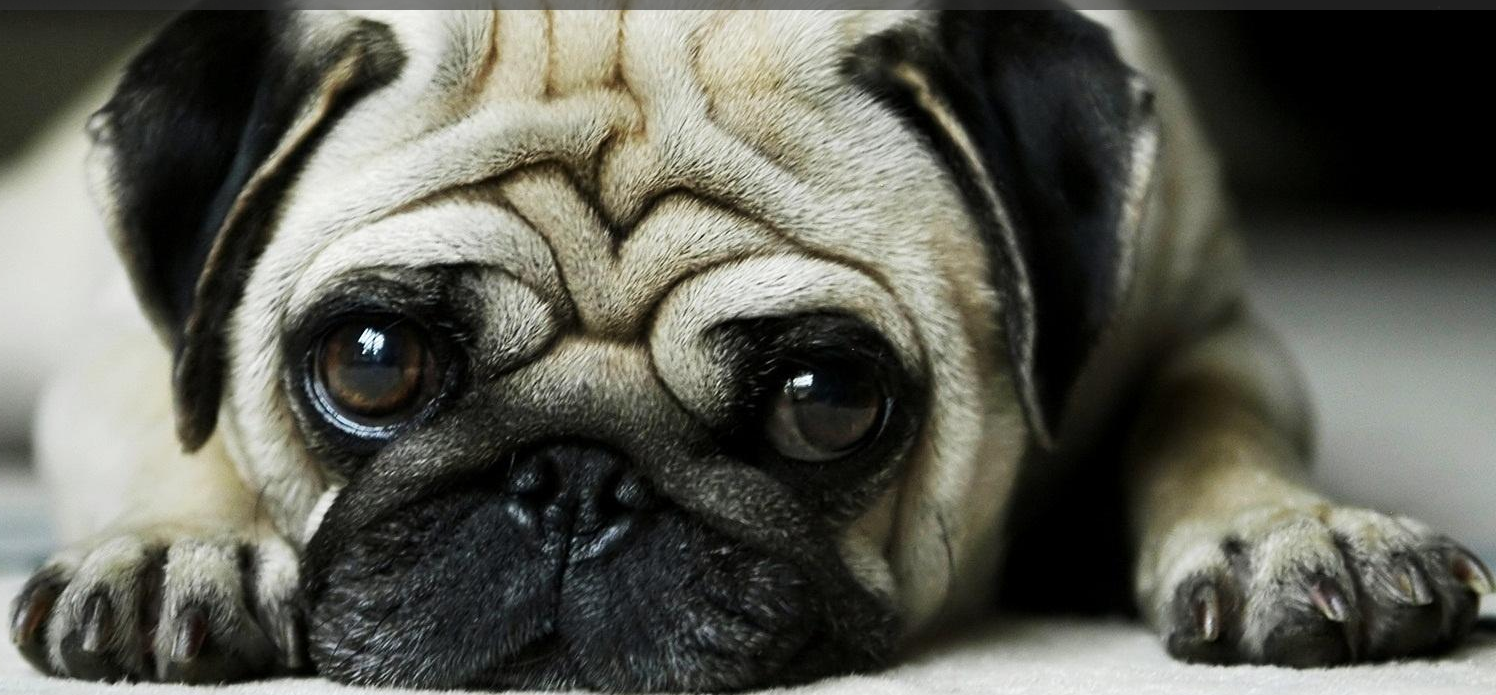
... -v `pwd` : /var/www

```
... -v /var/log/apache2
```

... --volumes-from=app

Demo

I'm on OSX and it's really really slow!



Docker on OSX isn't native

VirtualBoxFs

...

VirtualBox Fileshare is slow ...




rsync!

github.com
/rawcode
/docker-developer-utils

```
docker-machine create -d  
virtualbox \  
--virtualbox-no-share \  
Development
```

```
docker-machine-rsync
```

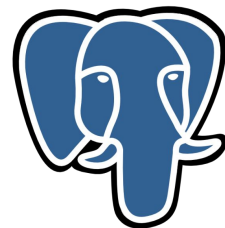
A black and white puppy is shown in profile, looking out of a window. The puppy has black fur with white markings on its chest and face. The background shows a wooden window frame and a view of a building outside.

**That's great, but
what about the
database,
cache,
elasticsearch,
etc**

Project Dependencies



Project Dependencies



redis

PostgreSQL

docker run memcached

docker run postgres

docker run elasticsearch

```
docker run -e  
APP_ENV=development -v .:  
/var/www -p 0.0.0.0:10000:80 -  
link elasticsearch:elasticsearch -  
link postgres:postgres -link  
memcached:memcached edpug:1
```

Docker Compose

...

version: '2'

services:

 application:

 image: php-7.0-apache

 redis:

 image: redis:version

Let's setup a project with:

- PHP 7
- Elasticsearch
- PostgreSQL
- memcached

... and let's do it in under one minute* ...

* docker pull commands done on the train and this text will be too small to read on a projector

Docker Compose Demo

Docker Tips

...

Composer

Keep it local

```
composer install --ignore-platform-reqs
```

- IDE AutoComplete
 - Cached to ~/.composer
 - Not passing keys or tokens to Docker container for satis / private repositories
-

Alpine Linux

~ 5 MiB

(with solid package management!)

FROM alpine:3.3

Base Image

ONBUILD FTW

```
# Dockerfile
```

```
ONBUILD COPY . /var/www
```

```
# docker-compose.yml
```

```
application:
```

```
  image: base-image
```

```
  volumes:
```

```
    - .:/var/www
```

```
# CI build
```

```
FROM base-image
```

Base Image

NIGHTLY BUILDS

cron / curl / wget / whatever!

There's no cascading builds
in Docker. Automate it.

FROM your-company:base:latest

FROM your-company:base:dev

FROM your-company:base:prod

Arbitrary Commands

github.com
/rawcode
/docker-developer-utils

```
docker-compose-run-command \  
php -v
```

Image Size

Keep it Lean!

```
RUN DEBIAN_FRONTEND=noninteractive \  
apt-get update \  
    && apt-get install -y \  
        --no-install-recommends apache2 \  
    && rm -rf /var/lib/apt/lists/*
```

Image Size

GOTCHAS

ADD `. /var/www`

RUN `chown www-data -R /var/www`

THIS WILL COST YOU TWICE THE
MiBs!

Do it at entrypoint, if really
needed

Image Size

If in doubt, check!

```
docker history myimage
```

Data Containers

Containerised logs and cache!

```
volumes:
```

- cache_and_logs

```
cache_and_logs:
```

- image: busybox
 - volumes:
 - /var/www/app/cache
 - /var/www/app/logs
-

Abuse Docker

WHA-EVA - I'LL DO WHAT I WANT!



```
testing-database-pdo:  
  extends:  
    service: testing-php-5.5  
  environment:  
    - DATABASE_USER=root  
    - DATABASE_PASS=testing  
    - DATABASE_NAME=testing
```

```
testing-database-mysql:  
  extends:  
    service: testing-database-pdo  
  links:  
    - mysql:mysql  
  environment:  
    - DATABASE_DRIVER=pdo_mysql  
    - DATABASE_HOST=mysql  
    - DATABASE_PORT=3306
```

Thanks for having me!

Questions?

Free advice, tweet:

[@rawkode](https://twitter.com/rawkode)

Not so free, but super-awesome, training for you and/or team?

david@rawkode.com

Send constructive (ish) feedback to:

joinind.in/talk/17628