

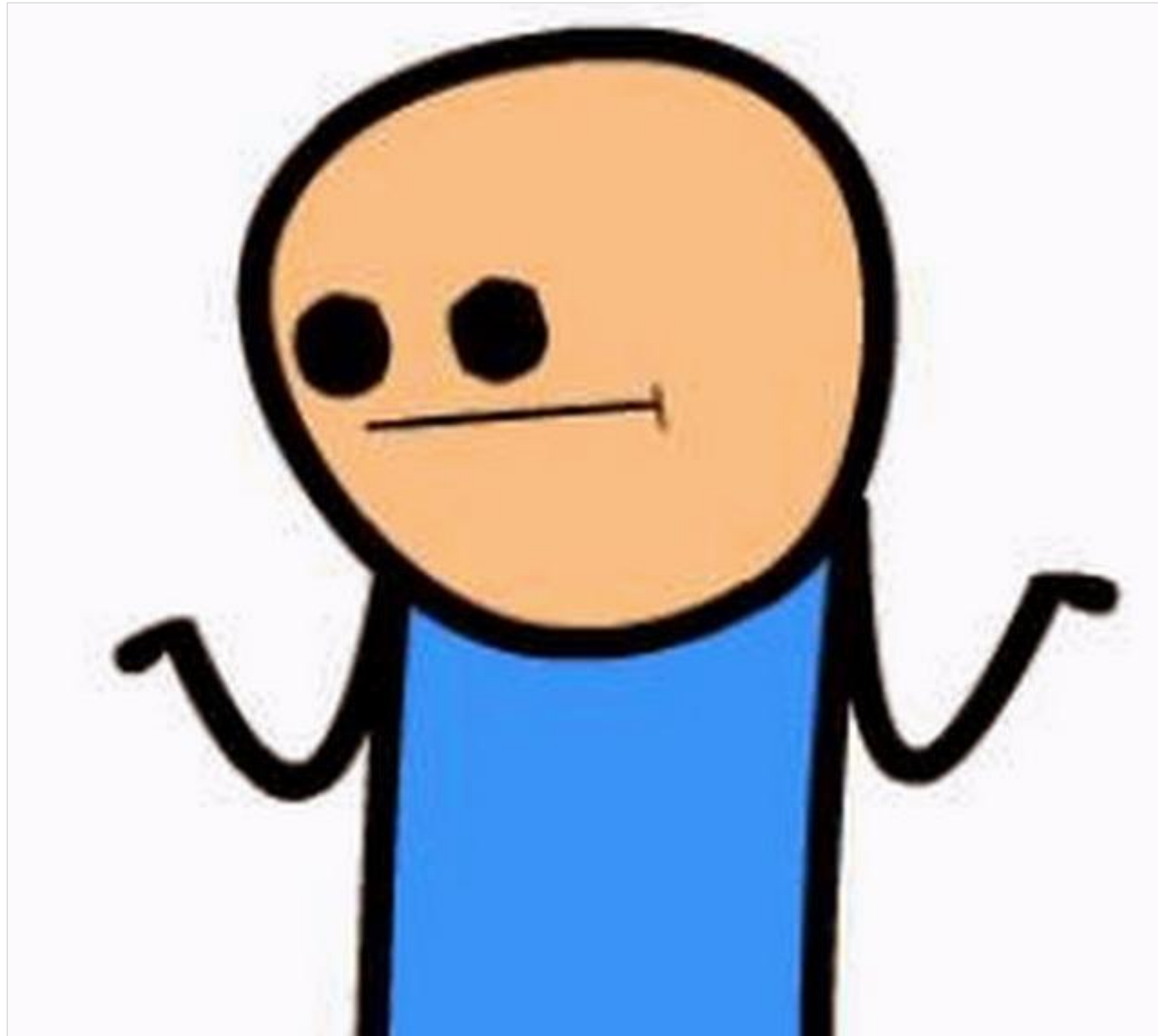


DEVOP-LESS DEPLOY WITH DOCKER

Davide Fantuzzi - Data Engineer

5 Maggio 2021 - **Più Docker per tutti**
#pugRomagna

WELCOME



Davide Fantuzzi

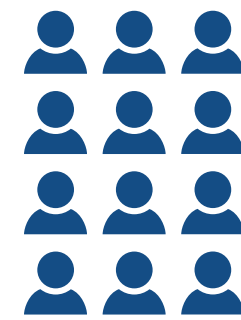
Data Engineer @ **LARUS Business Automation**

- Currently **Scala, Neo4j, Apache Spark**.
- Previously **PHP, MySQL** and another one.
- **utnaf** on GitHub, Twitter, etc...
- Music, Drums, Inter

ABOUT LARUS



Founded in **2004**



Data Engineer, Data Architect, Data Scientist, Big Data certified experts team



HQ: **Venice**
Offices: **Pescara, Rome**



Leader in development of data-driven application based on NoSQL & Event Streaming Technologies.

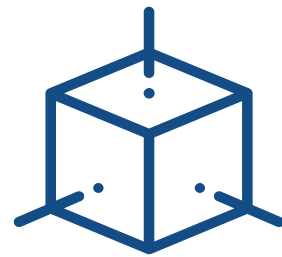


Global services
International projects

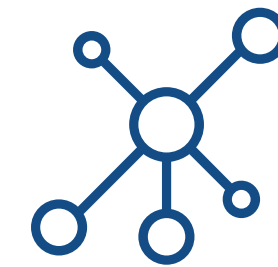


We help companies to become insight-driven organizations

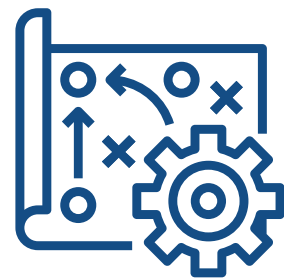
LARUS: OUR SPECIALTIES



Big Data Platform Design & Development (Java, Scala, Python, Javascript)



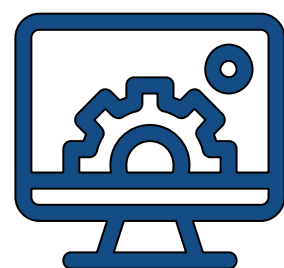
Data Science



Data Engineering



Machine Learning and AI
graph based technology

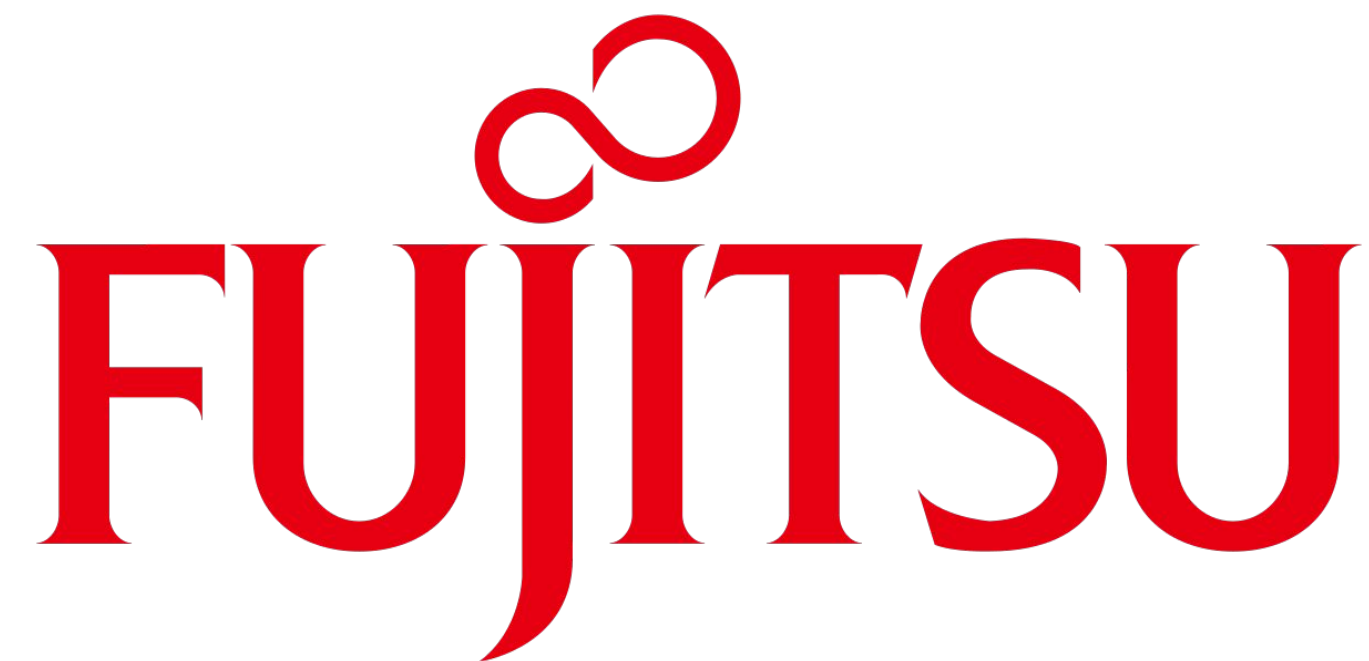


Graph Data Visualization



Strategic Advising for
Data-Driven Transformation
Projects

LARUS: OUR PARTNERS



WE'RE
HIRING



LARUS

Discover more:

<https://www.larus-ba.it/larus-jobs/>

AGENDA

- Problem
- Solution
- Hands on



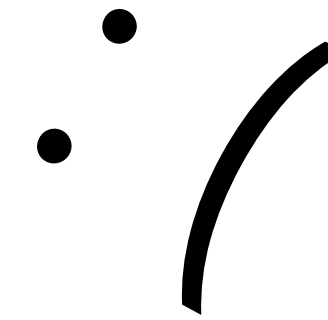
SCENARIO

- **Awesome (PoC / Demo / In-Progress)App**
- Need it online
- Need it online *NOW!*
- Need it done in the proper way
 - No git repository on the server
 - No publish our local environment
 - No FTP please!

Need for speed

SCENARIO (part 2)

- Don't know how to AWS / Azure / *Name it*
- Don't even know how to spell kubernetes
- *Money-Less Deploy* approach



HELP
Need for ~~speed~~



Step 1

Ask Devops team to make a simple deploy pipeline, please.

ff

No.

Devops answering any request for the
first time.



Step 2

Bring Devops team coffee and ask them to make a deploy pipeline by tomorrow, **please**.



Step 3



Step 4

Ping Devops for heads-up.

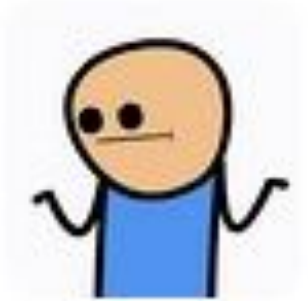
Step 5



Step 6

Ping again Devops for heads-up. **PLEASE!!**

Step 7



devop-buddy 6:24 PM

Sorry, something came up, didn't had time ^_(\ツ)_/-

Step 8



Let's prepare our own pipeline

Ingredients

- 1 Symfony WebApp
- 1 Remote Server with Docker and SSH
- 1 Dockerfile
- 1 DockerHub account

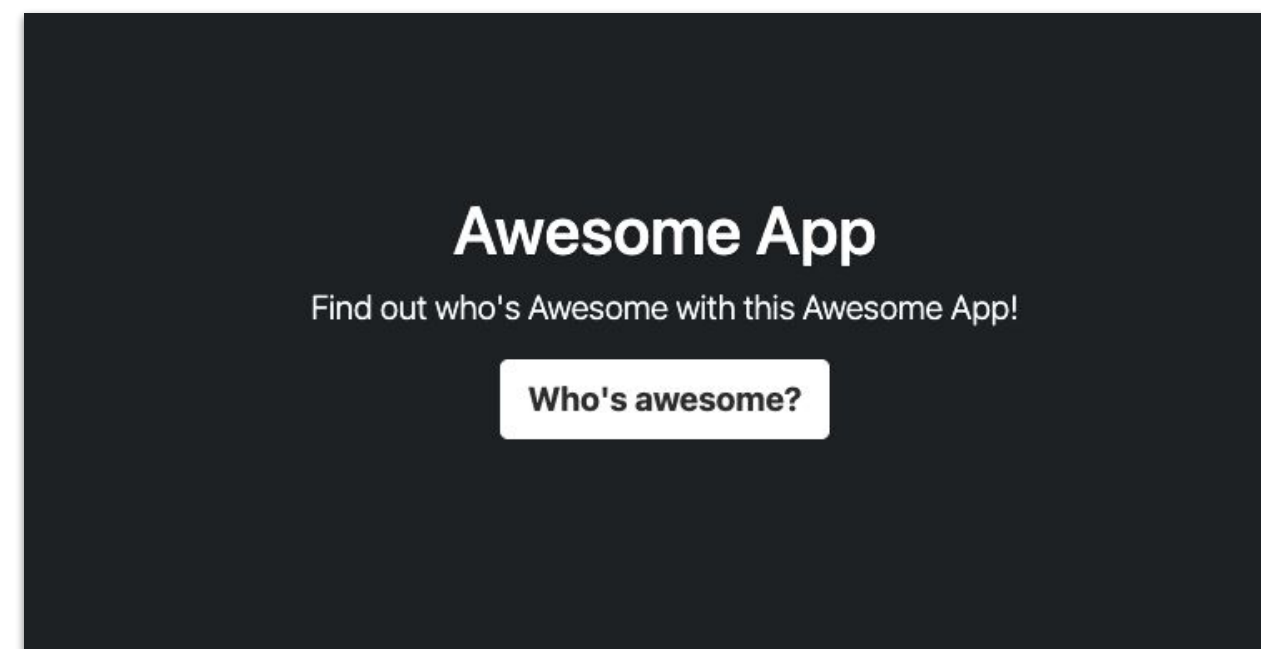
Preparation time: 20~ minutes

Cooking time: no cooking needed



GOAL

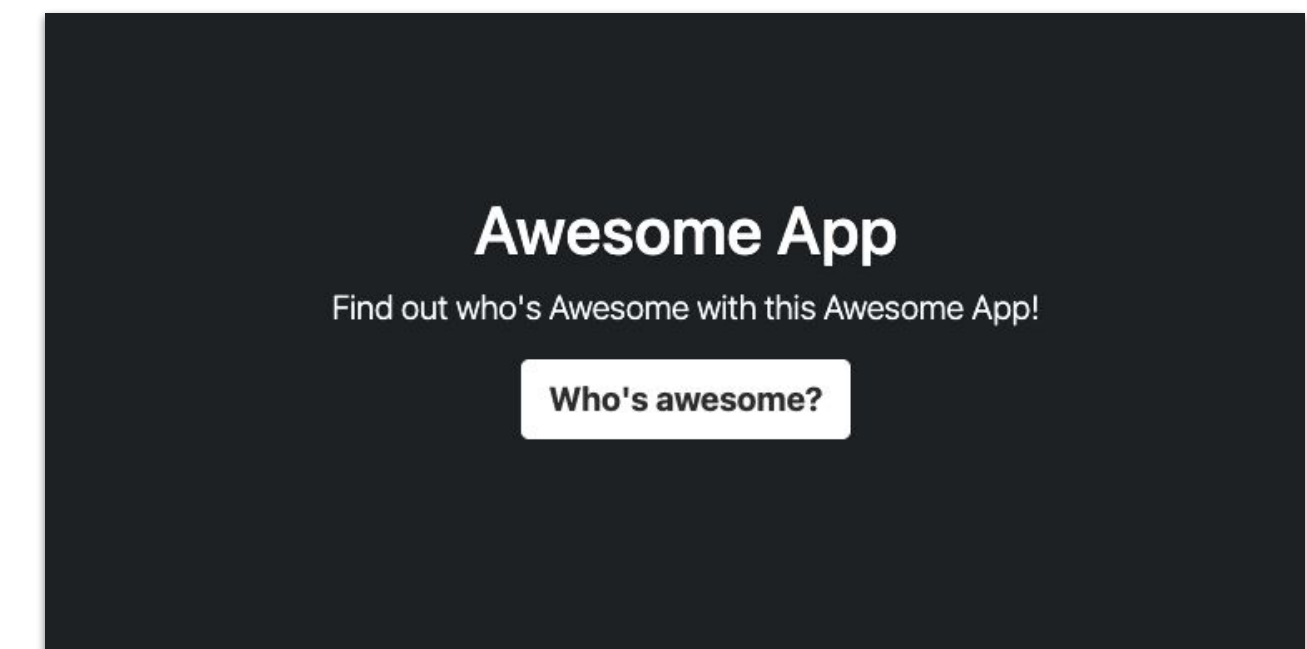
`http://localhost:8000`



Less manual step as possible



`http://awesome-app.prod`




HOW

- Create Docker image for our PHP application
 - Put Docker image *somewhere**
 - Allow the remote server to reach the image
 - Run the image on the server
-
- Easy to update
 - Configurable
 - Not-volatile


***somewhere?**


DOCKER HUB



[Explore](#) [Repositories](#) [Organizations](#) [Get Help](#)

utnaf





utnaf

[Edit profile](#)

Community User


Joined January 9, 2016

Repositories

Starred

Contributed

Displaying 3 of 3 repositories



utnaf/zeppelin

By [utnaf](#) • Updated 22 days ago

Zeppelin image with Spark 2.4.5

Container

15

0

DownloadsStars

DOCKER HUB



`docker login`
`docker push utnaf/awesome.app`



`docker pull`
`utnaf/awesome.app`



`docker pull`
`utnaf/awesome.app`



`docker pull utnaf/awesome.app`



`docker pull`
`utnaf/awesome.app`



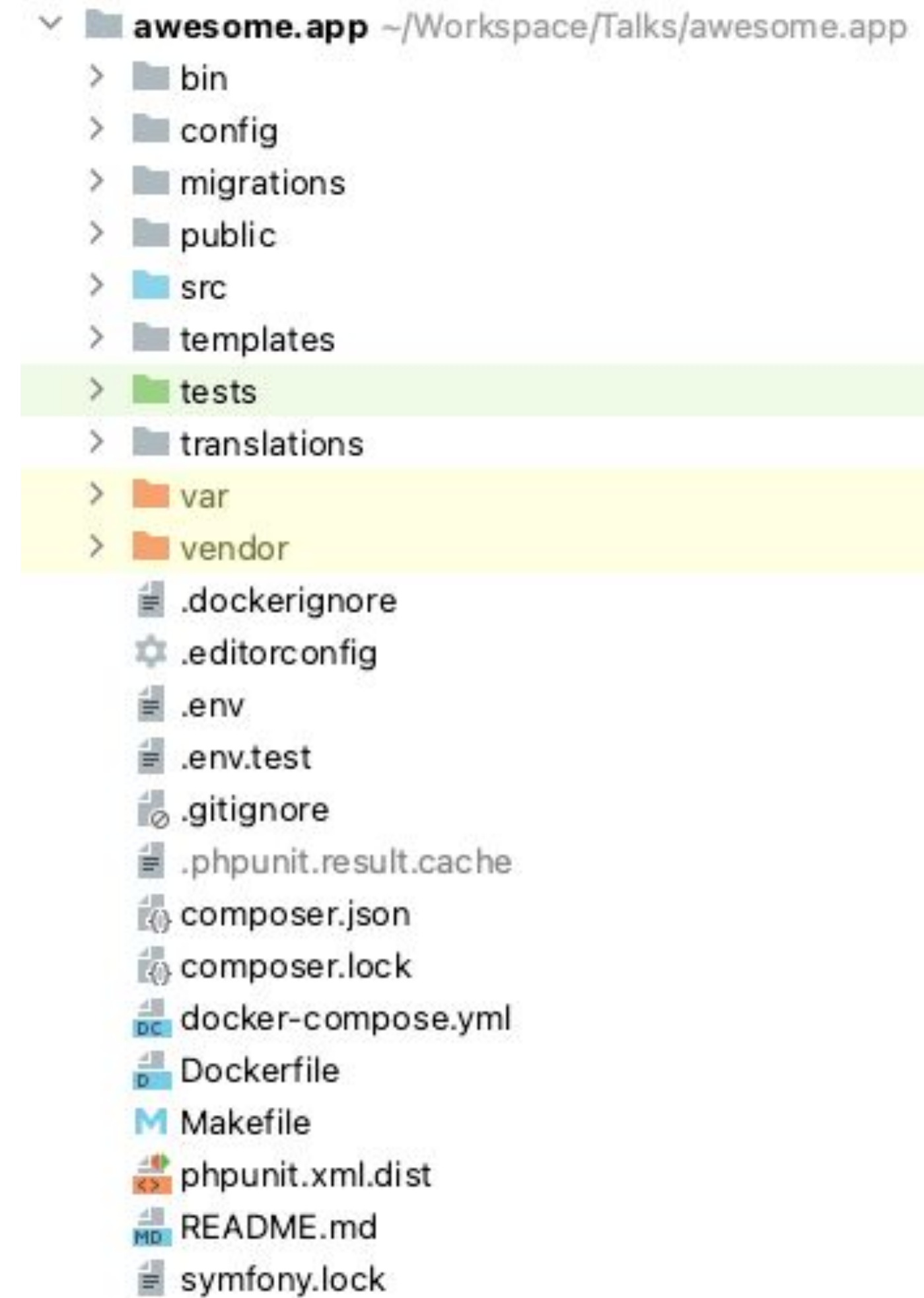
`docker pull`
`utnaf/awesome.app`

OUR AWESOME RECIPE

- Build the image
- Push it to DockerHub
- Access our server
- Create a simple `docker-compose.yml`
 - Our app image
 - Other images (MySQL, etc...)
 - Connect them with a network
 - Set up volumes for persistence
- Automate some operations with a script (migrations, cache clean, etc...)
- Execute the script from your dev environment

OUR AWESOME APP

- Symfony 5 App
- MySQL Database on docker-compose
- Dockerfile
- Makefile for laziness



OUR AWESOME DOCKERFILE

```
FROM php:8.0
```

```
WORKDIR /var/www
```

```
# Install PHP Stuff
```

```
RUN apt-get update \  
    && apt-get install -y zip libzip-dev # ...and some more
```

```
# Create `var` directory and add writing permissions
```

```
RUN mkdir var \  
    && chmod 755 -R var \  
    && chown www-data:www-data -R var
```

```
COPY . .
```

```
COPY --from=composer /usr/bin/composer /usr/bin/composer
```

```
RUN composer install --no-dev --optimize-autoloader --no-scripts
```

```
EXPOSE 8000
```

```
CMD php -S 0.0.0.0:8000 -t public
```

Let's do this!

Recap

OUR SOLUTION IS

- (Relatively) simple to implement
- Fast
- Easy to maintain
- Can be extended
- Dev friendly
- Devops friendly

Code available <https://github.com/utnaf/devopless-deploy>



Questions?

THANKS FOR YOUR ATTENTION

Davide Fantuzzi

Data Engineer

twitter.com/utnaf

