

# Fejlesszünk Dockerrel!?

Palócz István  
<http://palocz.hu>

# Milyen egy “startup”

- Egy project
- Egy stack
- Ritkán változik
- na jó... 3-4 project... évente

# Milyen egy ügynökség

- Heti 3-4 project
- Nincs azonos stack
- Gyors váltások

# Docker mindent megold

- Különböző operációs rendszerek
- Biztonság
- Teljesítmény
- Skálázhatóság
- Mindenfajta plusz szaktudás nélkül is

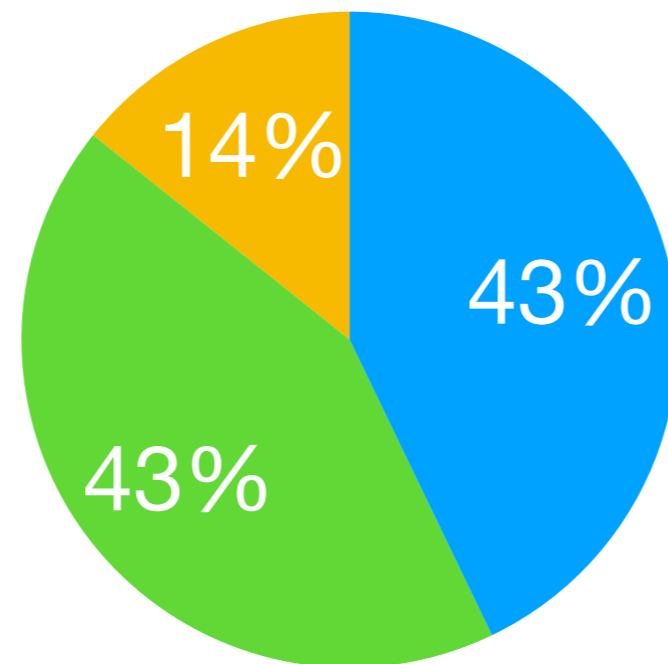


**Do not use in production**



# Saját preferenciák

● Windows ● Linux ● Mac



# Windows

- szimbolikus linkek
- futtatható jogosultság +x



# MAC

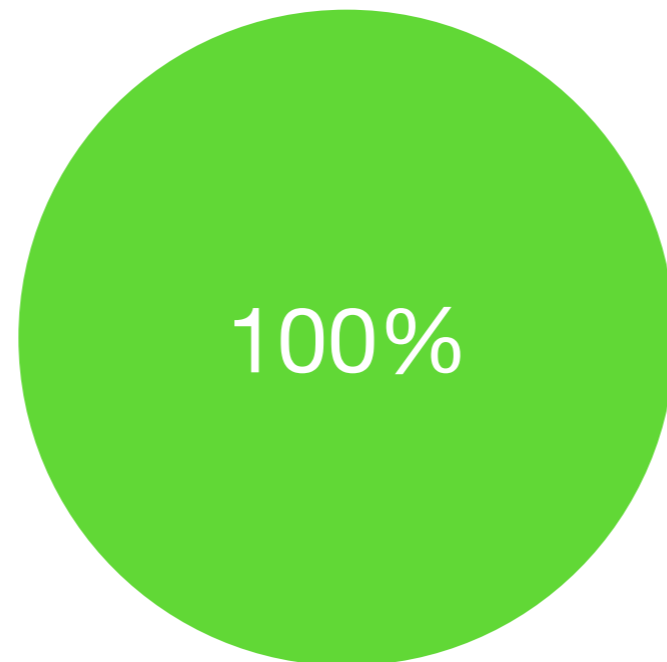
- Sebesség
- /Users

# Linux

- File tulajdonos

# Kissebb változások

● Windows ● Linux ● Mac



# Szolgáltatások

- PHP (web, cli, workerek, phpmyadmin)
- MySQL(standalone, Galera), PostgreSQL, MariaDB
- Node (Vue, React)
- Memcache, Redis (Cache)
- Elasticsearch, Solar
- Beanstalkd
- Mailcatcher
- Ruby, Go, stb

# Elasticsearch

- Kellenek a korlátok

# Node

- node\_modules
- Package-lock.json :)

# PHP

- web
- cli
- xdebug
- xdebug-cli

# docker-compose

- `docker-compose.yml` - repoban
- `docker-compose.override.yml`



# Helper scriptek

- Shell a web konténerhez
- npm
- mysql-console
- artisan, phpunit, phpcs stb.
- phpunit-xdebug

# shell

```
#!/bin/bash
```

```
if [[ ! $1 ]]
```

```
then
```

```
    docker-compose exec --user $(ls -lnd . | awk '{print $3}') web bash
```

```
else
```

```
    docker-compose exec --user $(ls -lnd . | awk '{print $3}') web $@
```

```
fi
```

# npm

```
#!/bin/bash
```

```
docker-compose run --rm --user $(ls -ln . | awk '{print $3}') node npm
```

# artisan

```
#!/bin/bash
```

```
./shell.sh php artisan $@
```

# Xdebug

- Honnan jön a kérés?
- ip címek
  - localhost: 127.0.0.1,
  - Docker hálózati cím: 192.168.65.x
  - Belső hálózati cím: 192.168.1.x
- Belső hálózati cím -> PHP-XDEBUG-IP
- Van jobb? Biztos.

# Xdebug-cli

```
#!/bin/bash
```

```
./shell.sh php \  
-dxdebug.remote_autostart=0n \  
-dxdebug.remote_enable=0n \  
-dxdebug.remote_host=$(cat ./PHP-XDEBUG-IP) \  
$@
```

# CI

- Minden ugyan az, mint a fejlesztő gépén
- Nem egészen :)
- shell runner
- fejlesztői gépeken is fut

# DOCKER\_COMPOSE\_ID

- **ci/run.sh**

```
docker-compose -p $DOCKER_COMPOSE_ID \  
exec --user $(stat -c '%u' .) -T web $@
```

- **gitlab-ci.yml**

**variables:**

```
DOCKER_COMPOSE_ID: "${CI_PIPELINE_ID}_${CI_JOB_ID}"
```



# Szolgáltatások leállítása

- ~~docker-compose stop; docker-compose rm~~
- docker-compose down

# Várakozni kell

## **waitingMysql**

```
#!/bin/bash
docker-compose -p $DOCKER_COMPOSE_ID exec -T web timeout 60s bash
waitingMysql.sh
```

## **waitingMysql.sh**

```
#!/bin/bash

while ! mysqladmin ping -h mysql -u dbuser -pdbpass --silent; do
sleep 1; echo "Waiting for MySQL";done
```

# Használjunk?

- Fejlesztéshez mindenképpen.
- Üzemeltetéshez?

15:30 Pásztor János: Practical Docker Operations  
(ugyanitt)