

GABRIELE ROMANATO

Come scrivere uno script di inizializzazione per Laravel in Docker con Docker Compose

Utilizzare Docker per eseguire Laravel è una scelta comune per chi vuole standardizzare l'ambiente di sviluppo. Uno script di inizializzazione consente di impostare correttamente l'ambiente Laravel al primo avvio, automatizzando operazioni come installazione delle dipendenze, migrazioni e generazione delle chiavi di sicurezza.

Struttura di base del progetto

Assumiamo che tu abbia una struttura di progetto simile alla seguente:

```
.
├─ docker-compose.yml
├─ init.sh
├─ app/
│   └─ artisan
│   └─ ... (file Laravel)
```

Il file docker-compose.yml

Ecco un esempio base di docker-compose.yml per Laravel:

```
services:
  app:
    image: php:8.2-fpm
    container_name: laravel_app
    working_dir: /var/www
    volumes:
      - ./app:/var/www
```

```
    - ./init.sh:/init.sh
command: ["sh", "/init.sh"]
depends_on:
  - mysql

web:
  image: nginx:alpine
  container_name: laravel_web
  ports:
    - "8000:80"
  volumes:
    - ./app:/var/www
    - ./nginx.conf:/etc/nginx/conf.d/default.conf
  depends_on:
    - app

mysql:
  image: mysql:8.0
  container_name: laravel_db
  restart: always
  environment:
    MYSQL_DATABASE: laravel
    MYSQL_USER: laravel
    MYSQL_PASSWORD: secret
    MYSQL_ROOT_PASSWORD: root
  ports:
    - "3306:3306"
```

Lo script init.sh

Questo script viene eseguito all'avvio del container app. Serve a preparare Laravel:

```
#!/bin/sh

set -e

# Installa Composer
EXPECTED_SIGNATURE=$(php -r
"copy('https://composer.github.io/installer.sig',
```

```
'php://stdout');")
php -r "copy('https://getcomposer.org/installer', 'composer-
setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') !==
'$EXPECTED_SIGNATURE') { echo 'Invalid installer';
unlink('composer-setup.php'); exit(1); }"
php composer-setup.php --install-dir=/usr/local/bin --
filename=composer
rm composer-setup.php

cd /var/www
composer install

php artisan key:generate

php artisan migrate --force

php-fpm
```

Ricorda di rendere eseguibile lo script con il comando:

```
chmod +x init.sh
```

Considerazioni finali

Questo script semplifica molto l'onboarding in team o la distribuzione locale di un'app Laravel. Puoi estenderlo aggiungendo supporto per la cache delle configurazioni, seeding del database o installazione di package globali.

Una buona pratica è anche usare un'immagine personalizzata che abbia già Composer installato, evitando così di scaricarlo ogni volta.