

GABRIELE ROMANATO

Menu

WordPress e Docker: esempio d'uso

In questo tutorial vedremo come usare WordPress con Docker.

Per prima cosa scarichiamo il file compresso dell'ultima release di WordPress in una directory di lavoro e lo scompattiamo. Verrà creata la directory `wordpress` che volendo possiamo rinominare.

Useremo un `Dockerfile` per installare PHP, Apache e le estensioni di PHP richieste da WordPress e gestiremo il nostro progetto con Docker Compose installando anche l'immagine di MySQL con la sua configurazione.

L'installazione delle estensioni PHP è una procedura alquanto prolissa, quindi possiamo includere questo script per semplificare il processo.

Creiamo quindi il `Dockerfile`.

```
FROM php:8.2-apache
ADD https://github.com/mlocati/docker-php-extension-
installer/releases/latest/download/install-php-extensions
/usr/local/bin/
RUN chmod +x /usr/local/bin/install-php-extensions && \
    install-php-extensions gd mysqli pdo pdo_mysql
COPY site/ /var/www/html/
```

A questo punto possiamo creare il file `docker-compose.yml` per completare la configurazione aggiungendo MySQL.

```
version: "3.9"
services:
  wordpress:
    container_name: wordpress
    build: .
    ports:
```

```
- "8000:80"
volumes:
  - ./site:/var/www/html
db:
  image: mysql
  container_name: db
  ports:
    - "3307:3306"
  command: --default-authentication-
plugin=mysql_native_password
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: password
    MYSQL_DATABASE: wordpress
    MYSQL_USER: wordpress
    MYSQL_PASSWORD: wpassword
```

L'installazione di WordPress sarà raggiungibile dalla nostra macchina host all'indirizzo `http://localhost:8000`, mentre potremo collegarci all'istanza MySQL usando la porta 3307 e le credenziali fornite nella sezione `environment` (unicamente a scopo dimostrativo, poiché le credenziali non andrebbero mai esposte nei file di configurazione).

La sezione `volumes` del container `wordpress` ci permette di mappare la directory di WordPress presente sulla nostra macchina con la document root di Apache presente nel container. In questo modo le modifiche fatte al codice non richiederanno un restart del container per avere effetto.

Ora dobbiamo aggiungere le credenziali di accesso al database nel file `wp-config.php` di WordPress.

```
define( 'DB_NAME', 'wordpress' );
define( 'DB_USER', 'wordpress' );
define( 'DB_PASSWORD', 'wpassword' );
define( 'DB_HOST', 'db' );
```

Anche qui le credenziali sono inserite a titolo di esempio, perché in questo caso si dovrebbe attuare una strategia diversa (come le variabili di

ambiente).

db è il nome del container MySQL e viene risolto internamente da Docker a livello DNS in quanto entrambi i container condividono la stessa rete (network).

Ora possiamo lanciare i container con il seguente comando:

```
docker compose up -d
```

Infine, possiamo aprire nel browser l'indirizzo <http://localhost:8000/wp-admin/install.php> e finalizzare l'installazione.