

En esta práctica vamos a montar un entorno de despliegue de aplicaciones modernas utilizando contenedores Docker.

Despliegue de Aplicaciones con Docker

Actividad 1.6

Cristóbal Suárez Abad
Administración de Sistemas Operativos
2º ASIR

Contenido

Instalación de Docker Compose.	2
1. Despliegue de Ruby on Rails	2
2. Despliegue de Prestashop	5
3. Despliegue de WordPress	7
4. Despliegue de Moodle:	9

Instalación de Docker Compose.

```
sudo apt-get update
```

```
sudo apt-get install docker-compose-plugin
```

Comprobamos version: docker compose version

1. Despliegue de Ruby on Rails

Crea un directorio a parte donde albergar el .yaml del docker compose. Dentro de ese directorio crea otro directorio llamado “myapps”.

```
mkdir -p /home/usuario/docker_compose/RubyOnRails/myapp
```

```
cd /home/usuario/docker_compose/RubyOnRails
```

- Genera un proyecto Rails dentro del contenedor:

```
docker run --rm -v $(pwd)/myapp:/app -w /app ruby:3.3 bash -c "gem install rails && rails new . --database=mysql"
```

- Edita myapp/config/database.yml y cambia la parte de development así:

development:

adapter: mysql2

encoding: utf8

database: myapp_development

username: rails

password: rails

host: mariadb

- docker-compose.yaml

```

services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb
    environment:
      - MARIADB_ROOT_PASSWORD=root
      - MARIADB_DATABASE=myapp_development
      - MARIADB_USER=rails
      - MARIADB_PASSWORD=rails
    volumes:
      - mariadb_data:/var/lib/mysql
    networks:
      - railsnet

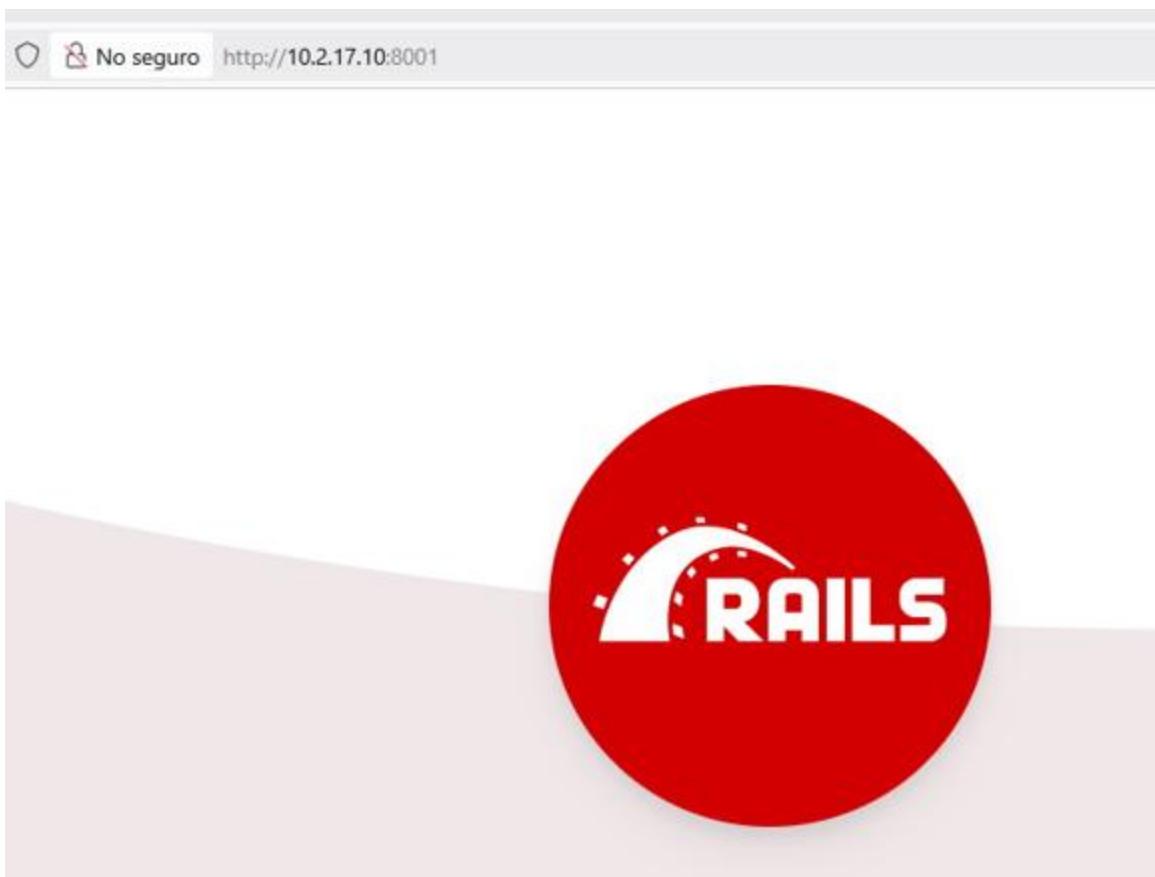
  myapp:
    image: ruby:3.3
    container_name: myapp
    ports:
      - "8001:3000"
    depends_on:
      - mariadb
    working_dir: /app
    volumes:
      - ./myapp:/app
    command: bash -c "bundle install && rails server -b 0.0.0.0"
    networks:
      - railsnet

  volumes:
    mariadb_data:

  networks:
    railsnet:
      driver: bridge


```

- Arranca: **docker compose up -d**
- <http://10.2.17.10:8001/>



2. Despliegue de Prestashop.

Usamos la guía:

<https://devdocs.prestashop-project.org/9/basics/installation/environments/docker/>

Creamos directorio:

```
mkdir -p /home/usuario/docker_compose/Prestashop  
cd /home/usuario/docker_compose/Prestashop
```

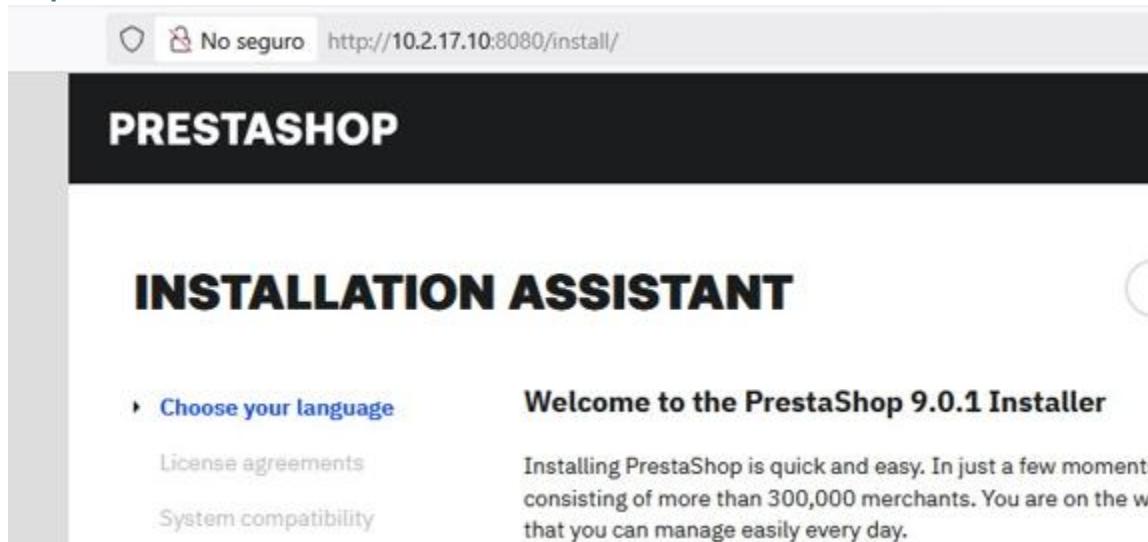
```
- docker-compose.yaml  
services:  
mariadb:  
  image: mariadb:latest  
  container_name: mariadb-prestashop  
  environment:  
    - MARIADB_ROOT_PASSWORD=root  
    - MARIADB_DATABASE=mitienda  
    - MARIADB_USER=pepe  
    - MARIADB_PASSWORD=pepe  
  volumes:  
    - mariadb_data:/var/lib/mysql  
  networks:  
    -prestashopnet  
  
prestashop:  
  image: prestashop/prestashop:latest  
  container_name:prestashop  
  depends_on:  
    - mariadb  
  ports:  
    - "8080:80"  
    - "8443:443"  
  environment:  
    - PRESTASHOP_HOST=127.0.0.1  
    - PRESTASHOP_DATABASE_HOST=mariadb  
    - PRESTASHOP_DATABASE_PORT_NUMBER=3306  
    - PRESTASHOP_DATABASE_NAME=mitienda  
    - PRESTASHOP_DATABASE_USER=pepe  
    - PRESTASHOP_DATABASE_PASSWORD=pepe  
    - PRESTASHOP_FIRST_NAME=Cristobal  
    - PRESTASHOP_LAST_NAME=Suarez
```

```
- PRESTASHOP_PASSWORD=admin123
- PRESTASHOP_EMAIL=admin@mitienda.local
volumes:
- prestashop_data:/bitnami/prestashop
networks:
- prestashopnet

volumes:
mariadb_data:
prestashop_data:

networks:
prestashopnet:
driver: bridge
```

- Arranca: docker compose up -d
- <http://10.2.17.10:8080>



3. Despliegue de WordPress.

Seguimos la guía:

<https://aulasoftwarelibre.github.io/taller-de-docker/docker-compose/#creacion-de-contenedores-automatizada>

- Creamos su propio directorio con “mkdir”.
- compose-docker-yaml

services:

db:

image: mariadb:latest

volumes:

- mariadb_wp_data:/var/lib/mysql

environment:

- MYSQL_ROOT_PASSWORD=12345

- MYSQL_DATABASE=wordpress

- MYSQL_USER=cristobal

- MYSQL_PASSWORD=12345

networks:

- wordpress

web:

image: wordpress:latest

depends_on:

- db

volumes:

- wp_data:/var/www/html

environment:

- WORDPRESS_DB_USER=cristobal

- WORDPRESS_DB_PASSWORD=12345

- WORDPRESS_DB_HOST=db

- WORDPRESS_DB_NAME=wordpress

ports:

- "8082:80"

networks:

- wordpress

volumes:

mariadb_wp_data:

wp_data:

networks:

wordpress:

- Arranca: **docker compose up -d**
- [**http://10.2.17.10:8082**](http://10.2.17.10:8082)



4. Despliegue de Moodle:

- Creamos su propio directorio:
- Dentro de ese directorio nos bajamos los ficheros de Moodle en una carpeta propia:

```
git clone -b MOODLE_404_STABLE git://git.moodle.org/moodle.git ./moodle-code  
sudo chown -R 33:33 ./moodle-code
```

- Creamos otro directorio para la carpeta “moodledata_local” que irá dentro del contenedor:

```
mkdir moodledata_local  
sudo chown -R 33:33 moodledata_local
```

- docker-compose.yaml

services:

mariadb:

... (Sin cambios)

image: mariadb:latest

container_name: mariadb-moodle

restart: always

environment:

MYSQL_ROOT_PASSWORD: rootpass

MYSQL_DATABASE: moodle

MYSQL_USER: cristobal

MYSQL_PASSWORD: 12345

volumes:

- mariadb_data:/var/lib/mysql

networks:

- moodle_network

moodle:

image: moodlehq/moodle-php-apache:8.4

container_name: moodle

restart: always

depends_on:

- mariadb

ports:

- "8083:80" # HTTP

- "8444:443" # HTTPS

environment:

```

MOODLE_DATABASE_TYPE: mysqli
MOODLE_DATABASE_HOST: mariadb
MOODLE_DATABASE_NAME: moodle
MOODLE_DATABASE_USER: cristobal
MOODLE_DATABASE_PASSWORD: 12345
MOODLE_USERNAME: cristobal
MOODLE_PASSWORD: 12345
MOODLE_SITE_NAME: "Moodle SUAREZ1"
MOODLE_EMAIL: admin@suarez1.abad2
volumes:
- ./moodle-code:/var/www/html
- ./moodledata_local:/var/www/moodledata #<-- Mapeo a carpeta local
networks:
- moodle_network

```

volumes:
mariadb_data:
Eliminados moodle_data y moodle_mariadb_data

networks:
moodle_network:

- Arranca: **docker compose up -d**
- <http://10.2.17.10:8083/install.php>



nullable type must be used instead in /var/www/html/lib/php-di/php-di/src/functions.
33

ated: Dl\autowire(): Implicitly marking parameter \$className as nullable is deprecated, the
nullable type must be used instead in /var/www/html/lib/php-di/php-di/src/functions.
46

Installation

Language

Choose a language

Please choose a language for the installation. This language will also
be used as the default language for the site, though it may be
changed later.

Language

English (en)

Next »

