



Instituto Tecnológico del valle de Oaxaca



Manual de prácticas de la Materia de Framework para el desarrollo de aplicaciones Web

Presenta

Ambrosio Cardoso Jiménez

Práctica 2. Instalación y configuración de Laravel 8 en Docker

Objetivo: Configurar docker para nginx (latest), php (latest), mysql8 y laravel 8 (latest)

latest=> Se refiere a la última versión

Requisitos: Tener instalado los siguientes programas

- Docker 20.10.12
- Tener instalado docker-compose
- Tener instalado composer (gestor de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías)

Temas relacionadas.

2.1 Instalación

Condiciones.

- Contar con servicios de Internet para la descarga de la imagen de nginx, php, mysql y laravel
- Contar con equipo de al menos 4GB de Ram con un espacio en Disco Duro de 50 GB
- Preferentemente sistema operativo Linux (Fedora, Centos, AlmaLinux, Debian)

Procedimiento y guía ilustrativa

Paso 1: Crear una carpeta **dockers/lara**

Paso 2: Dentro de ella crear carpeta **nginx** y dentro de ella un archivo de texto **nginx.conf** cuyo contenido será la siguiente:



```
server {  
  
    listen 80;  
  
    root /var/www/html/proyecto_laravel/public;  
  
    index index.php index.html;  
  
    server_name localhost;  
  
  
    error_log /var/log/nginx/error.log;  
    access_log /var/log/nginx/access.log;  
  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
  
    location ~ \.php$ {  
        try_files $uri =404;  
        fastcgi_split_path_info ^(.+\.php)(/.+)$;  
        fastcgi_pass php8:9000;  
        fastcgi_index index.php;  
        include fastcgi_params;  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        fastcgi_param PATH_INFO $fastcgi_path_info;  
    }  
}
```



```
}
```

Paso 3: Crear la carpeta vacía **dockers/lara/www**

Paso 4: Crear el archivo **docker-compose.yml** en el directorio **lara** con el siguiente contenido (La sangría es sumamente importante):

```
version: '3.8'
```

```
networks:
```

```
  lara:
```

```
services:
```

```
  nginx:
```

```
    image: nginx:stable-alpine
```

```
    container_name: lara_nginx
```

```
    ports:
```

```
      - "8888:80"
```

```
    volumes:
```

```
      - "./www:/var/www/html"
```

```
      - "./nginx/nginx.conf:/etc/nginx/conf.d/default.conf"
```

```
    depends_on:
```

```
      - php8
```

```
      - mysql8
```

```
    networks:
```

```
      - lara
```



mysql8:

image: mysql:latest

container_name: mysql8

volumes:

- mysql8_data:/var/lib/mysql

command: ['--character-set-server = utf8mb4', '--collation-
server = utf8mb4_unicode_ci', '--default-authentication-
plugin = mysql_native_password']

tty: true

ports:

- "3333:3306"

environment:

MYSQL_DATABASE: bdlara

MYSQL_USER: webdev

MYSQL_PASSWORD: pwdWebDev

MYSQL_ROOT_PASSWORD: t0pS3cr3t

SERVICE_NAME: mysql8

networks:

- lara

php8:

build:

context: .



dockerfile: Dockerfile

container_name: php8

volumes:

- "./www:/var/www/html"

ports:

- "9999:9000"

networks:

- lara

volumes:

mysql8_data:

Paso 5: Crear el archivo **Dockerfile** en el directorio **lara** con el siguiente contenido:

#--- El repositorio docker de PHP

FROM php:8.0.5-fpm-alpine

RUN apk add --no-cache bash

RUN docker-php-ext-install pdo pdo_mysql mysqli

#--- Usar apk en lugar de apt-get para cuando se usa alpine

#--- nota de Cardoso 26-Ene-2022

RUN apk update && apk add \

curl \

wget \

zip \

unzip \

```
composer \
npm
```

Paso 6: Ejecutar la siguiente instrucción desde la carpeta **lara**

docker-compose build

NOTA: En ocasiones aparece el siguiente mensaje de error:

ERROR: Couldn't connect to Docker daemon at http+docker://localunixsocket - is it running?

SOLUCION: iniciar docker (systemctl start docker para linux y simplemente ejecutarlo para Windows)

Volver a ejecutar la sentencia **docker-compose build**

Paso 7: Iniciar los contenedores ejecutando

docker-compose up -d (-d => para ejecutar en segundo plano)

Paso 8: Comprobar que los tres contenedores (nginx, mysql8 y php) estén corriendo

docker container ls

```
[cardoso@fedora lara]$ docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
fac22732ce95   nginx:stable-  "/docker-entrypoint..." About a minute Up About a minute 0.0.0.0:8888->80/tcp, :::8888->80/tcp lara_nginx
e44debe39330   lara_php8     "docker-php-entrypoi..." About a minute Up About a minute 0.0.0.0:9999->9000/tcp, :::9999->9000/tcp php8
ac122aeb9ee    mysql:latest   "docker-entrypoint.s..." About a minute Up About a minute 33060/tcp, 0.0.0.0:3333->3306/tcp, :::3333->3306/tcp mysql8
```

Figura 1. Contenedores en ejecución

Paso 9: Entrar al contenedor

docker exec -it php8 /bin/sh

Paso 10: Escriba composer. Debe mostrar la documentación de composer, si dice que no existe o no reconoce la sentencia entonces **ejecutar las siguientes sentencias**

```
# php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
# HASH="$ (wget -q -O - https://composer.github.io/installer.sig)"
# php composer-setup.php --install-dir=/usr/local/bin --filename=composer
# mv /usr/local/bin/composer.phar /usr/bin/composer
```

Paso 11: Estando en el contenedor cambiar al directorio */var/www/html*
`cd /var/www/html`

Paso 12: Crear el proyecto **proyecto_laravel** estando en la ruta */var/www/html*
ejecutar:

```
# composer create-project --prefer-dist laravel/laravel proyecto_laravel
```

esperar a que termine la descarga...

Paso 13: Editar el archivo */var/www/html/proyecto_laravel/.env* cambiando los datos resaltados

APP_NAME = lara

APP_ENV = local

APP_KEY = base64:82Z0EFT/ysqWDaaThBcTajemwOsVT1Pkov4pLkTJePw =

APP_DEBUG = true

APP_URL = http://localhost:8000

LOG_CHANNEL = stack

DB_CONNECTION = mysql

DB_HOST = 172.18.0.3

DB_PORT = 3306

DB_DATABASE = bdlara

DB_USERNAME = webdev

DB_PASSWORD = pwdWebDev

...

La **IP** se obtiene con la siguiente instrucción

```
docker inspect -f  
'{{range .NetworkSettings.Networks}}  
  {{.IPAddress}}{{end}}'  
mysql8
```

(En una sola línea, **mysql8** es el nombre del contenedor como se aprecia en la figura 1)

Paso 14: Probar desde el navegador <http://localhost:8888>

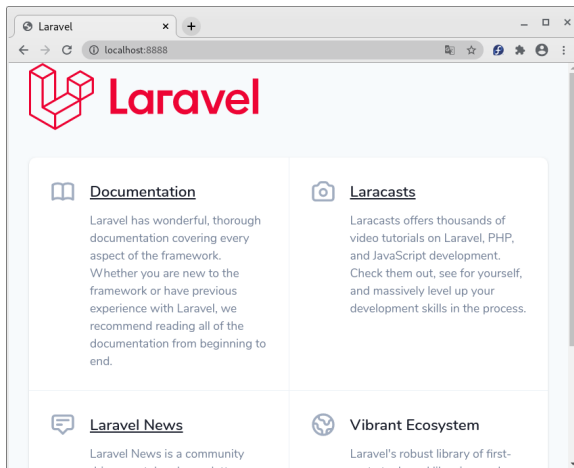


Figura 2. Laravel en ejecución

Consulta de rúbricas



https://plataforma.voaxaca.tecnm.mx/cardoso/rubricas/instalacion_laravel.pdf