

Wordpress + Traefik

Requerimientos previos

- Docker instalado
- Instancia de Traefik instalada y corriendo
- Red de Docker creada para conectar contenedor con Traefik
- Opcionalmente, usa un [middleware configurado en Traefik](#):
 - autoDetectContentType@file
 - permissionsPolicy@file
 - referrerPolicy@file
 - cspWordpress@file
 - secHeaders@file

Pasos

1. Preparar dominio/subdominio de la instancia a configurar
2. Crear archivos de configuración
3. Cambiar valores de acuerdo a las necesidades de la instancia
4. Levantar contenedor (docker composer up)
5. Visitar aplicación y comprobar que esté funcionando correctamente

Configuración previa

Crear secretos en Docker para acceso a base de datos

```
# Creación de secretos para credenciales de base de datos
printf "palabra secreta 1" | docker secret create db_root_password -
printf "palabra secreta 2" | docker secret create db_user -
printf "palabra secreta 3" | docker secret create db_password -
```

Archivos de configuración

.env

```
INSTANCE_URL=ejemplo.com
DB_NAME=nombre_base_de_datos
DB_TABLE_PREFIX=prefijo_
UID=1000
GID=1000
GID_FILES=33
VOLUME_PATH=./web
```

compose.yaml

En la configuración de `entrypoints` y `certresolver` cambiar de acuerdo a los valores configurados en Traefik.

```
services:
  bd:
    image: mariadb:latest
    container_name: bd
    volumes:
      - ${VOLUME_PATH}/wordpress-db:/var/lib/mysql
      - ./entrypoint-initdb:/docker-entrypoint-initdb.d:ro
    environment:
      - UID=${UID}
      - GID=${GID}
      - MYSQL_ROOT_PASSWORD=/run/secrets/db_root_password
      - MYSQL_USER=/run/secrets/db_user
      - MYSQL_PASSWORD=/run/secrets/db_password
      - MYSQL_DATABASE=${DB_NAME}
    restart: unless-stopped
    networks:
      - traefik
  wp:
    image: wordpress:latest
    container_name: wp
    volumes:
      - ${VOLUME_PATH}/wordpress-files:/var/www/html
      - ./uploads.ini:/usr/local/etc/php/conf.d/uploads.ini
    environment:
```

```

- UID=${UID}
- GID=${GID_FILES}
- WORDPRESS_DB_HOST=bd
- WORDPRESS_DB_NAME=${DB_NAME}
- WORDPRESS_TABLE_PREFIX=${DB_TABLE_PREFIX}
- WORDPRESS_DB_USER=/run/secrets/db_user
- WORDPRESS_DB_PASSWORD=/run/secrets/db_password
depends_on:
- bd
restart: unless-stopped
labels:
- "traefik.enable=true"
- "traefik.http.routers.wordpress-http.rule=Host(`${INSTANCE_URL}`)"
- "traefik.http.routers.wordpress-http.entrypoints=http"
- "traefik.http.routers.wordpress-http.middlewares=http-to-https"
- "traefik.http.middlewares.http-to-https.redirectscheme.scheme=https"
- "traefik.http.routers.wordpress.rule=Host(`${INSTANCE_URL}`)"
- "traefik.http.routers.wordpress.entrypoints=https"
- "traefik.http.routers.wordpress.tls=true"
-
"traefik.http.routers.wordpress.middlewares=permissionsPolicy@file,referrerPolicy@file,cspWord
press@file,secHeaders@file,autoDetectContentType@file"
- "traefik.http.routers.wordpress.tls.certresolver=tls"
networks:
- traefik
networks:
traefik:
external: true

```

uploads.ini (opcional)

Configuración especial para PHP, para permitir subir archivos de hasta 100MB

```

file_uploads = On
memory_limit = 256M
upload_max_filesize = 100M
post_max_size = 100M
max_execution_time = 600

```

Agregar Redis a la configuración

Para agregar Redis, como objeto de cache, para optimizar la instancia de Wordpress se debe añadir lo siguiente a los archivos de configuración:

.env

```
REDIS_HOST_PASSWORD=ContraseñaDeRedisAquí
```

compose.yaml

A la configuración del servicio de WordPress se debe agregar:

```
environment:
  - WORDPRESS_CONFIG_EXTRA=
    define('WP_REDIS_HOST', 'nombre_del_servicio_de_redis');
    define('WP_REDIS_PASSWORD', '${REDIS_HOST_PASSWORD}');
```

y la configuración del servicio de redis sería la siguiente:

```
redis:
  image: redis:alpine
  container_name: redis
  restart: unless-stopped
  ports:
    - '6379:6379'
  command: redis-server --requirepass ${REDIS_HOST_PASSWORD} --maxmemory 128mb --maxmemory-policy allkeys-lru
  volumes:
    - ${VOLUME_PATH}/redis:/var/lib/redis
    - ${VOLUME_PATH}/redis-data:/data
  networks:
    - traefik
```