

📦 Docker 📦📦📦📦📦

WordPress 📦

📦📦📦📦 Docker → WordPress 📦📦📦

1. 📦 Docker 📦

- 📦 **docker-compose.yml** 📦📦📦
 - **nginx** 📦 Web server 📦📦📦📦📦📦📦📦 Proxy 📦
 - **php (php-fpm)** 📦 WordPress/PHP
 - **mariadb** 📦📦📦 WordPress 📦
 - **redis** 📦 Object Cache 📦📦📦📦
 - **wp-cli** 📦📦📦
 - **cron** 📦 WordPress 📦 wp-cron 📦
- 📦
 - Nginx 📦📦📦📦📦📦📦
 - PHP FPM 📦 memory_limit 📦 pm.max_children 📦📦📦 4C/4G RAM 📦
 - MariaDB 📦 volume 📦📦📦 utf8mb4 📦
 - Redis 📦 volume 📦📦📦

2. 📦📦 Proxy (Ubuntu Nginx)

- 📦 **Nginx** 📦📦 → 127.0.0.1:8080 📦📦 nginx 📦
- 📦 **SSL (Let's Encrypt Certbot)** 📦📦📦📦 server_name mysite.com; 📦
- 📦 **502** 📦 📦📦📦 nginx 📦📦 port 📦 127.0.0.1:8080 📦
- 📦 proxy_header 📦 WordPress 📦📦 HTTPS 📦📦 CSS/JS 📦📦📦

```
proxy_set_header X-Forwarded-Proto https;
proxy_set_header X-Forwarded-Port 443;
```

3. WordPress

- `wp-cli` WordPress

```
docker compose exec wp-cli wp core download --allow-root
```

- `wp-config.php`
 - DB `wp`
 - `wpuser`
 - `change_me`
 - Host `db:3306` ← Docker localhost
-

4.

- **Nginx**
 - `fastcgi_cache_path` http server
 - →
- **cron**
 - Restarting → `cron/wp-cron.sh`
- **wp-config.php**
 - Redis `define('WP_REDIS_HOST', redis);` → `define('WP_REDIS_HOST', 'redis');`
 - HTTPS

```
if (isset($_SERVER['HTTP_X_FORWARDED_PROTO']) && $_SERVER['HTTP_X_FORWARDED_PROTO'] === 'https') {
    $_SERVER['HTTPS'] = 'on';
}
```

5. Redis

- Redis `docker compose exec redis redis-cli ping → PONG`
- `wp-config.php`

```
define( 'WP_REDIS_HOST', 'redis' );
define( 'WP_REDIS_PORT', 6379 );
define( 'WP_REDIS_DATABASE', 1 );
```

- PHP opcache docker compose restart php
- wp-cli Redis drop-in

```
docker compose exec wp-cli wp redis enable --allow-root
docker compose exec wp-cli wp redis status --allow-root
```

6. WordPress

- CSS/JS 502
- Redis Object Cache redis:6379
- Proxy + Nginx + PHP-FPM + DB



```
wordpress-perf/
├─ docker-compose.yml
├─ .env
├─ nginx/
│  └─ conf.d/
│     └─ wordpress.conf
│     └─ 00-cache.conf
├─ cron/
│  └─ wp-cron.sh
└─ README.md
```



1. .env

```
HTTP_PORT=8080
```

```
DB_NAME=wp
```

```
DB_USER=wpuser
```

```
DB_PASSWORD=change_me
```

```
DB_ROOT_PASSWORD=change_root
```

```
# PHP 7.4 / 4GB RAM 4C/4G RAM
```

```
PHP_MEMORY_LIMIT=256M
```

```
PHP_MAX_CHILDREN=10
```

```
PHP_MAX_REQUESTS=400
```

2. docker-compose.yml

```
services:
```

```
  nginx:
```

```
    image: nginx:1.25-alpine
```

```
    restart: always
```

```
    ports:
```

```
      - "127.0.0.1:${HTTP_PORT:-8080}:80"
```

```
    volumes:
```

```
      - ./nginx/conf.d:/etc/nginx/conf.d:ro
```

```
      - wp_data:/var/www/html:ro
```

```
      - cache_data:/var/cache/nginx
```

```
    depends_on:
```

```
      - php
```

```
  php:
```

```
    build:
```

```
      context: .
```

```
      dockerfile_inline: |
```

```
        FROM wordpress:php8.3-fpm-alpine
```

```
        RUN docker-php-ext-install mysqli pdo pdo_mysql
```

```
        COPY --from=mlocati/php-extension-installer /usr/bin/install-php-extensions /usr/local/bin/
```

```
    restart: always
```

```
    environment:
```

WORDPRESS_DB_NAME: \${DB_NAME}
WORDPRESS_DB_USER: \${DB_USER}
WORDPRESS_DB_PASSWORD: \${DB_PASSWORD}
WORDPRESS_DB_HOST: db:3306
PHP_MEMORY_LIMIT: \${PHP_MEMORY_LIMIT}

volumes:

- wp_data:/var/www/html

db:

image: mariadb:10.11

restart: always

environment:

MYSQL_DATABASE: \${DB_NAME}

MYSQL_USER: \${DB_USER}

MYSQL_PASSWORD: \${DB_PASSWORD}

MYSQL_ROOT_PASSWORD: \${DB_ROOT_PASSWORD}

volumes:

- db_data:/var/lib/mysql

redis:

image: redis:7-alpine

restart: always

volumes:

- redis_data:/data

wp-cli:

image: wordpress:cli-php8.3

depends_on:

- php

- db

volumes:

- wp_data:/var/www/html

entrypoint: wp

cron:

image: wordpress:cli-php8.3

depends_on:

- php

volumes:

```
- wp_data:/var/www/html
- ./cron:/cron
entrypoint: /cron/wp-cron.sh
```

volumes:

```
wp_data:
db_data:
redis_data:
cache_data:
```

4. nginx/conf.d/wordpress.conf

```
server {
    listen 80 default_server;
    server_name _;
    root /var/www/html;
    index index.php index.html;

    # [] [] [] []
    set $skip_cache 0;
    if ($request_method = POST) { set $skip_cache 1; }
    if ($query_string != "") { set $skip_cache 1; }
    if ($http_cookie ~*
"wordpress_logged_in_|woocommerce_items_in_cart|woocommerce_cart_hash|wp_woocommerce_session_") {
    set $skip_cache 1; }
    if ($request_uri ~* "^/(wp-admin/|wp-login\.php|cart|checkout|my-account|wc-api/|wp-json/)" ) { set
$skip_cache 1; }
    if ($request_uri ~* "add-to-cart=") { set $skip_cache 1; }
    if ($request_uri ~* "admin-ajax\.php") { set $skip_cache 1; }

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_pass php:9000;
```

```
fastcgi_buffers 16 32k;
fastcgi_buffer_size 32k;
fastcgi_read_timeout 120s;

fastcgi_cache_bypass $skip_cache;
fastcgi_no_cache $skip_cache;
fastcgi_cache PHP_MICRO;
fastcgi_cache_valid 200 301 302 5s;

add_header X-Cache $upstream_cache_status;
fastcgi_param HTTPS on;
fastcgi_param SERVER_PORT 443;
fastcgi_param HTTP_X_FORWARDED_PROTO https;
}

location ~* \.(?:css|js|jpg|jpeg|gif|png|svg|webp|ico|ttf|otf|woff|woff2)$ {
    expires 30d;
    access_log off;
}

client_max_body_size 64m;
}
```

5. cron/wp-cron.sh

```
#!/bin/sh
set -e
# 1 1 1 1 1 wp-cron
while true; do
    wp --path=/var/www/html --allow-root cron event run --due-now >/dev/null 2>&1 || true
    sleep 60
done
```

6. README.md

```
# WordPress on Docker (WooCommerce Ready)
```

```
## Services
```

- Nginx 1.25 (reverse proxy inside container)
- PHP-FPM 8.3 (with mysqli, pdo_mysql)
- MariaDB 10.11
- Redis 7 (object cache)
- WP-CLI
- Cron container for wp-cron

```
## Usage
```

```
```bash
```

```
docker compose up -d
```

## WordPress setup

```
docker compose exec wp-cli wp core download --allow-root
docker compose exec wp-cli wp config create \
 --dbname=$DB_NAME --dbuser=$DB_USER --dbpass=$DB_PASSWORD \
 --dbhost=db:3306 --skip-check --allow-root
docker compose exec wp-cli wp core install \
 --url="https://mysite.com" \
 --title="MySite Shop" \
 --admin_user="admin" \
 --admin_password="ChangeMe123!" \
 --admin_email="you@example.com" \
 --allow-root
```

## Redis setup

```
docker compose exec wp-cli wp config set WP_REDIS_HOST 'redis' --allow-root
docker compose exec wp-cli wp config set WP_REDIS_PORT 6379 --raw --allow-root
docker compose exec wp-cli wp config set WP_REDIS_DATABASE 1 --raw --allow-root
docker compose restart php
docker compose exec wp-cli wp redis enable --allow-root
```

## Proxy setup (Host machine)

## Nginx reverse proxy:

```
server {
 listen 443 ssl http2;
 server_name mysite.com;

 ssl_certificate /etc/letsencrypt/live/mysite.com/fullchain.pem;
 ssl_certificate_key /etc/letsencrypt/live/mysite.com/privkey.pem;

 location / {
 proxy_pass http://127.0.0.1:8080;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto https;
 }
}
```

# Result

- WordPress     (<https://mysite.com>)
- Redis Object Cache
- Nginx + PHP-FPM + MariaDB + Redis

---

#2

tainan   2025-08-22 00:30:38 UTC

tainan   2025-08-22 01:08:26 UTC