**Drupal Docker Setup: For Existing Drupal Projects**

This guide explains how to run your **existing Drupal 10/11 project** using Docker Compose with:

* **MariaDB (database),**
* **PHP (custom build),**
* **Nginx (web server),**
* Ubuntu helper container for debugging and tools (optional)

**1️. System Requirements**

* **Docker Engine** installed  
  *(check: docker --version)*
* **Docker Compose** installed  
  *(check: docker-compose --version)*

**2️. Expected Project Structure**

The existing Drupal project should have the following structure (or be adjusted to fit):

my\_drupal\_ubuntu/

├── docker-compose.yml # Docker Compose file

├── nginx.conf # Nginx configuration

├── php.ini # (Optional) PHP settings override

├── web/ # Your existing Drupal site's document root

│ └── index.php

│ └── core/

│ └── sites/

│ └── ...

├── vendor/ # Composer dependencies

├── composer.json # Existing Drupal project’s composer file

├── php-ubuntu/ # PHP Docker build context

│ └── Dockerfile

If your Drupal uses /docroot, need to update docker-compose.yml and nginx.conf to point to the docroot.

**3️. Files & Configs**

**docker-compose.yml**

**nginx.conf**

**php-ubuntu/Dockerfile**

**php.ini** (optional, to override PHP settings, e.g., memory limits or file upload size.)

**4️. Setup Instructions (for Existing Drupal Project)**

**1️. Prep project**

* Copy existing Drupal site into the Sites (or other folder*.*

my\_drupal\_ubuntu/

├── web/

│ └── index.php

│ └── core/

│ └── modules/

│ └── ...

├── composer.json

**2️. Check vendor/ exists**

If not, run composer install

**3️. Set database credentials**

Check web/sites/default/settings.php or local settings file to  
update the **database settings** to match Docker Compose:

'database' => [

'default' => [

'default' => [

'database' => 'drupal10',

'username' => 'drupal10',

'password' => 'drupal10',

'host' => 'my\_drupal\_ubuntu\_db',

'port' => '3306',

'driver' => 'mysql',

'prefix' => '',

],

],

],

**4️. Build and run containers**

In the project root, run:

docker-compose up --build -d

**5️. Check container status** docker-compose ps

**6️. Test the site in the browser**

* On local machines:  
  <http://localhost:8088>
* On Azure Dev Box:  
  Replace localhost with **Dev Box IP address.**

**5️. Database:**

docker exec -i my\_drupal\_ubuntu\_db mysql -u drupal10 -pdrupal10 drupal10 < your\_database.sql

**6️. Useful Docker Commands**

**Check logs:**

docker-compose logs nginx

docker-compose logs php

docker-compose logs db

**Enter the PHP container:**

docker exec -it my\_drupal\_ubuntu\_php bash

**Stop all containers:**

docker-compose down

**Rebuild & restart:**

docker-compose down

docker-compose up --build -d

**7️. Using the Ubuntu helper container**

To debug or run Linux tools inside the container:

docker exec -it my\_drupal\_ubuntu\_shell bash

apt update && apt install vim

cd /var/www/web

**What to lookout for existing project**

* Make sure your existing site's folder (e.g., web/, docroot/) matches what’s configured in nginx.conf and docker-compose.yml.
* Verify vendor/ is there
* Make sure the DB user/password/database name in settings.php or in local settings file matches what’s defined in Docker Compose.

Drush commands:

./vendor/bin/drush status # check status

./vendor/bin/drush uli # one time login

./vendor/bin/drush cr # clear cache