**YAML FILE UNDERSTANDING**

**1. Version Declaration**

version: '3.8'

* This specifies the **Docker Compose file format version**.
* '3.8' is compatible with Docker Engine 19.03.0+ and supports various networking and volume features.
* It ensures that the correct features are available in the Docker Compose environment.

**2. Services Section**

This section defines two **Docker services**:

1. **Database (db)**
2. **WordPress (wordpress)**

Each service has specific configurations.

**2.1. MySQL Database Service (db)**

db:

image: mysql:5.7

container\_name: wordpress\_db

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: rootpass

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wpuser

MYSQL\_PASSWORD: wppass

volumes:

- db\_data:/var/lib/mysql

* **image: mysql:5.7** → Uses **MySQL version 5.7** as the database.
* **container\_name: wordpress\_db** → Assigns a **custom name** to the database container (wordpress\_db).
* **restart: always** → Ensures that the database container restarts automatically if it stops or crashes.
* **environment** → Defines necessary **environment variables** for MySQL:
  + MYSQL\_ROOT\_PASSWORD: rootpass → Sets the **root password**.
  + MYSQL\_DATABASE: wordpress → Creates a **database** named wordpress.
  + MYSQL\_USER: wpuser → Creates a **user** named wpuser.
  + MYSQL\_PASSWORD: wppass → Sets the **password** for wpuser.
* **volumes:** → Defines a **persistent storage volume** (db\_data) for MySQL.
  + Maps the container’s MySQL data directory (/var/lib/mysql) to the volume db\_data, ensuring that database data persists even if the container is restarted or removed.

**2.2. WordPress Application Service (wordpress)**

wordpress:

depends\_on:

- db

image: wordpress:latest

container\_name: wordpress\_app

restart: always

ports:

- "80:80"

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wpuser

WORDPRESS\_DB\_PASSWORD: wppass

WORDPRESS\_DB\_NAME: wordpress

volumes:

- wp\_data:/var/www/html

* **depends\_on:** → Ensures that the **database service (db) starts before** the WordPress service.
* **image: wordpress:latest** → Uses the **latest** WordPress **official image**.
* **container\_name: wordpress\_app** → Assigns a **custom name** (wordpress\_app) to the WordPress container.
* **restart: always** → Ensures that the WordPress container **automatically restarts** if it crashes.
* **ports:**
  + **"80:80"** → Maps port **80 on the host** to port **80 inside the container**, making WordPress accessible via http://localhost (or the public IP of the machine).
* **environment:** → Defines **database connection details**:
  + WORDPRESS\_DB\_HOST: db:3306 → Points to the **MySQL service (db)** at port **3306**.
  + WORDPRESS\_DB\_USER: wpuser → Uses the **database user** created in the MySQL service.
  + WORDPRESS\_DB\_PASSWORD: wppass → Uses the **password** for wpuser.
  + WORDPRESS\_DB\_NAME: wordpress → Uses the **database** created in the MySQL service.
* **volumes:** → Defines a **persistent storage volume (wp\_data)** for WordPress.
  + Maps /var/www/html (WordPress's web root) to wp\_data, ensuring that files persist across container restarts.

**3. Volumes Section**

volumes:

db\_data:

wp\_data:

* Defines two **named volumes**:
  1. **db\_data** → Stores MySQL database files.
  2. **wp\_data** → Stores WordPress site files (themes, plugins, uploads).
* These volumes ensure **data persistence**, so deleting or restarting the containers won't remove the database or website files.

**4. What Happens When You Run docker-compose up?**

When you execute:

docker-compose up -d

* Docker Compose reads the docker-compose.yml file and **creates both containers** (wordpress\_db and wordpress\_app).
* The **MySQL service (db) starts first**, as required by WordPress.
* The **WordPress service (wordpress) starts next**, connecting to the MySQL database.
* Port **80** on your local machine is now linked to WordPress.
* You can **access the WordPress installation page** in a browser via:
* http://localhost (or your public EC2 IP if using AWS)
* Since **persistent storage (db\_data and wp\_data) is used**, data remains even if containers are restarted.

**5. Summary of What This File Does**

* **Defines a complete WordPress + MySQL stack**.
* **Ensures proper database connectivity** using environment variables.
* **Uses named volumes** to persist data across restarts.
* **Ensures MySQL starts first** using depends\_on.
* **Maps WordPress to port 80** so it's accessible via a browser.
* **Automatically restarts services** using restart: always.

This setup is useful for **local development, testing, and production environments** where a self-contained WordPress instance is required.

**1. MySQL Service (db) Environment Variables**

environment:

MYSQL\_ROOT\_PASSWORD: rootpass

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wpuser

MYSQL\_PASSWORD: wppass

* **MYSQL\_ROOT\_PASSWORD: rootpass**
  + This sets the **root user password** for MySQL.
  + By default, MySQL does **not** set a root password, so we explicitly define it here.
* **MYSQL\_DATABASE: wordpress**
  + This creates a **new database** called wordpress.
  + If not specified, MySQL would start with **no pre-created databases**.
* **MYSQL\_USER: wpuser**
  + Creates a **custom database user** named wpuser.
  + By default, MySQL **only provides a root user**. This allows WordPress to use a **non-root user** for better security.
* **MYSQL\_PASSWORD: wppass**
  + Sets the **password** for wpuser.
  + Without this, the custom user (wpuser) would not be created.

📌 **Conclusion**: These variables ensure MySQL is **pre-configured** with a database and a user that WordPress can connect to, instead of relying on manual setup.

**2. WordPress Service (wordpress) Environment Variables**

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wpuser

WORDPRESS\_DB\_PASSWORD: wppass

WORDPRESS\_DB\_NAME: wordpress

* **WORDPRESS\_DB\_HOST: db:3306**
  + Tells WordPress **where to find the database**.
  + db refers to the **MySQL service name** defined in docker-compose.yml.
  + 3306 is the **default port for MySQL**.
* **WORDPRESS\_DB\_USER: wpuser**
  + Specifies the **database username** that WordPress should use.
  + This must match the MYSQL\_USER in the MySQL service.
* **WORDPRESS\_DB\_PASSWORD: wppass**
  + Specifies the **password** for the wpuser database account.
  + Must match the MYSQL\_PASSWORD in the MySQL service.
* **WORDPRESS\_DB\_NAME: wordpress**
  + Tells WordPress **which database to connect to**.
  + Must match MYSQL\_DATABASE in the MySQL service.

📌 **Conclusion**: These variables allow WordPress to automatically connect to MySQL without needing **manual configuration** during installation.

**3. What Would Happen Without These Environment Variables?**

If we **did not specify** these environment variables, Docker would **fall back to defaults**, which could lead to issues like:

* **MySQL would not have a database** for WordPress.
* **WordPress would fail to connect** due to missing credentials.
* **Manual database setup** would be required.

By **explicitly defining** these values, we ensure that everything is correctly configured **automatically** when we run:

docker-compose up -d