PHP Environment
on Windows

WSL2 + Docker Desktop installation guide
and step-by-step usage — Windows 10 / 11

🌐 Apache 2.4   ⚙️ PHP 7.4   💾 MariaDB 10.6   🔧 phpMyAdmin   🐳 Portainer

🌐 PHP Site	`http://localhost:8080`
📱 phpMyAdmin	`http://localhost:8081`
🐳 Portainer (GUI)	`http://localhost:9000`
🏠 Dashboard	`http://localhost:8082`

Table of Contents — Windows

> **ℹ About this tutorial**
>
> This guide is designed for Windows 10/11 with Docker Desktop + WSL2. All operations are performed through graphical interfaces after the initial setup.

1 Install WSL2 and Docker Desktop

Two programs to install once

① Enable WSL2 (Windows Subsystem for Linux)

WSL2 allows Docker to run efficiently on Windows. It is free and built into Windows 10/11.

1 Open PowerShell as administrator

Start menu → search for "PowerShell" → **right-click → "Run as administrator"**

2 Type the installation command

Copy-paste exactly:

```
wsl --install
```

3 Restart your PC

Required after installing WSL2

4 After reboot: Ubuntu installs automatically

Create a Linux username and password when prompted (remember them!)

> **i WSL2 already installed?**
>
> Check the version: in PowerShell type `wsl --list --verbose`. If you see "VERSION 2", you're good. Otherwise: `wsl --set-default-version 2`

② Install Docker Desktop for Windows

1 **Go to docker.com → "Download for Windows"**

Download the `Docker Desktop Installer.exe` file

2 **Run the installer**

Check **"Use WSL 2 instead of Hyper-V"** if offered — then install

3 **Restart if prompted**

4 **Open Docker Desktop → Settings → Resources → WSL Integration**

Enable integration with **Ubuntu** → click "Apply & Restart"

✅ Docker Desktop is ready when...

The whale icon 🐳 appears in the taskbar

Bottom-right corner of the Windows screen (notification area). It must be stable (not animated = ready).

> ⚠️ **Docker Desktop must stay open**
> Docker Desktop must remain running (even in the background) during your entire work session.

2 First launch

Launch the environment with start.bat (double-click)

🚀 Launch the environment

1 **Open Docker Desktop from the Start menu**

Wait for the whale icon to appear in the taskbar (bottom-right corner)

2 **Double-click on** `start.bat`

This file is in the project folder. A black command window opens.

3 **Wait for the confirmation message**

`[OK] Environment started successfully!`

4 **Your browser opens automatically to the dashboard**

http://localhost:8082 — all your services are accessible

! **Go to Portainer IMMEDIATELY!**

Open http://localhost:9000 within the first 5 minutes to create your account. See Ch.3 for details.

⏱ **First time: wait 5 to 10 minutes**

Docker is downloading all the images (PHP, MariaDB...). This is slow **only the first time**. Subsequent launches: less than 10 seconds.

⚠️ If start.bat closes immediately

Solution A

Right-click on `start.bat`

→ **"Run as administrator"**

Solution B

Open **CMD** in the folder (Shift + right-click) → type `start.bat` to see the error message

📁 Where is the project folder?

Access from Windows Explorer

If the project is on your Desktop:

```
C:\Users\YourName\Desktop\PHPenv\
```

Your PHP files go in the `htdocs\` subfolder

3 Access the web interfaces
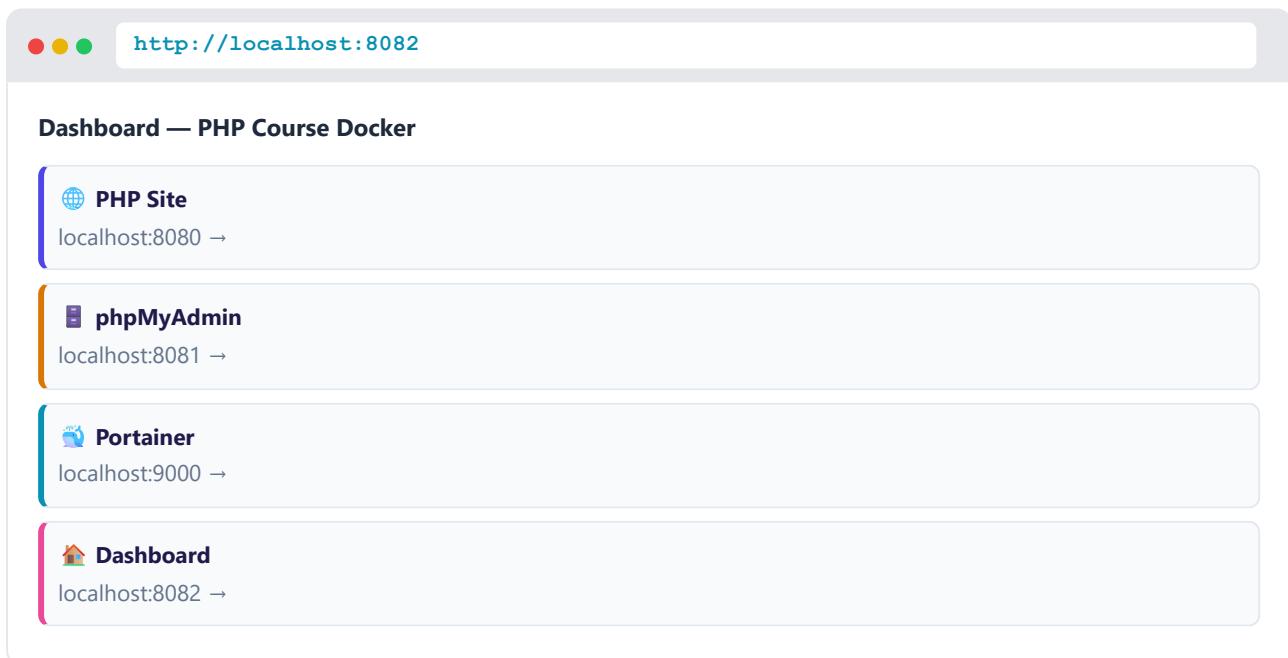
Everything happens in your Windows browser

🗺️ Overview

SERVICE	ADDRESS	DESCRIPTION
🏠 Dashboard	`localhost:8082`	Home page with all links
🌐 PHP Site	`localhost:8080`	Your PHP files (htdocs/)
🗄️ phpMyAdmin	`localhost:8081`	Visual management of the DB
🐳 Portainer	`localhost:9000`	Docker container management

🏠 Dashboard — localhost:8082

Your daily **starting point**. It centralizes all links and your credentials.
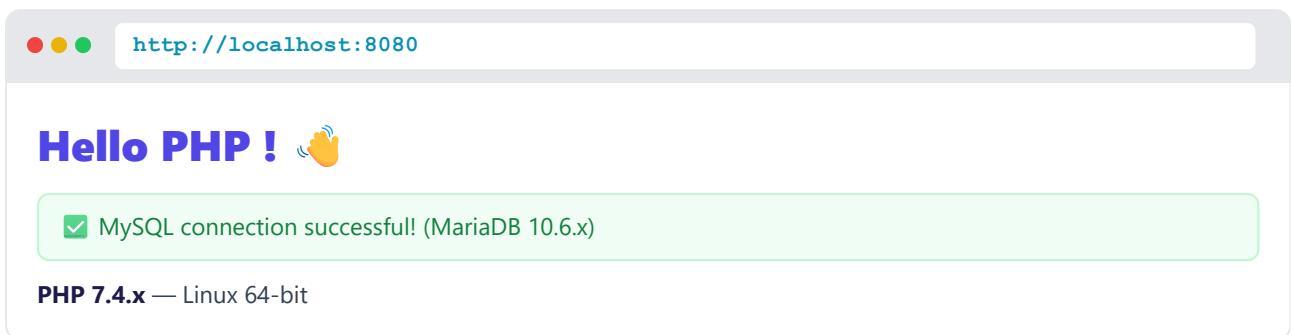
```
● ● ●    http://localhost:8082
```

Dashboard — PHP Course Docker

🌐 **PHP Site**
localhost:8080 →

🗄️ **phpMyAdmin**
localhost:8081 →

🐳 **Portainer**
localhost:9000 →

🏠 **Dashboard**
localhost:8082 →

🌐 PHP Site — localhost:8080

Displays the contents of your `htdocs\` folder. The test page confirms that PHP and the database are working.

Hello PHP ! 👋

✅ MySQL connection successful! (MariaDB 10.6.x)

PHP 7.4.x — Linux 64-bit

🗄 phpMyAdmin — localhost:8081

Graphical interface to manage your database. Pre-configured connection (automatic).

🌐 Address	**http://localhost:8081**
👤 Username	**root**
🔑 Password	**root**
💾 Database	**app**

⏱ **phpMyAdmin not responding?**
The DB takes 10 to 20 seconds to start. Wait and refresh with **Ctrl+R** or **F5**.

🐳 Portainer — localhost:9000

Graphical interface to manage your Docker containers (start, stop, restart, view logs).

⚠️ **Act within the first 5 minutes!**
Portainer expires its initialization for security reasons. After startup, go **immediately** to http://localhost:9000 and create your account. If you see the "*timeout*" page → see the Troubleshooting chapter (p. 10) or double-click on `reset-portainer.bat`.

Create your account (first access)

1 **Open http://localhost:9000**

Portainer asks you to create an administrator account

2 **Enter a username and a password**

Example: `admin` — Password: 12 characters minimum

3 **Click "Create user"**

Then choose **"Get Started" → "local"**

What you can do in Portainer

`http://localhost:9000 → Containers`

Container	Status	Available actions
cours_web	running	▶ ■ ↻ 📋
cours_db	running	▶ ■ ↻ 📋
cours_phpmyadmin	running	▶ ■ ↻ 📋

▶ Start   ■ Stop   ↻ Restart   📋 View logs

4 Working with PHP

Create, edit and test your PHP scripts on Windows

📁 The htdocs folder — your workspace

All your PHP files must be in the `htdocs\` folder of the project. It is directly accessible from **Windows Explorer**.

> 📁 **Where is htdocs?**
>
> ```
> C:\Users\YourName\Desktop\PHPenv\
>
> ├── 📁 htdocs\          ← YOUR PHP FILES HERE
> │   ├── index.php        ← test page (already created)
> │   └── hello.php        ← your next file
> └── docker-compose.yml ← do not modify
> ```

✍️ Create your first PHP file

1 **Open the** `htdocs\` **folder in Windows Explorer**

Double-click from your Desktop → `PHPenv` folder → `htdocs`

2 **Create a new file** `hello.php`

In VS Code, Notepad++, or any editor. The `.php` extension is required.

3 **Write your PHP code and save**

4 **Open in the browser**

`http://localhost:8080/hello.php` — refresh with **Ctrl+F5**

```php
<?php

echo "Hello world!";

?>
```

> 🎉 **No need to restart!**
> Every change to your PHP files is visible immediately after **Ctrl+F5** in the browser.

🔌 Database connection (PDO)

```php
// Connect to the database
$pdo = new PDO(
    "mysql:host=db;dbname=app;charset=utf8mb4",
    "app",    // username
    "app"     // password
);


// Retrieve data
$stmt = $pdo->query("SELECT * FROM my_table");
foreach ($stmt as $row) {
    echo $row['name'] . "<br>";
}
```

> 💡 **The host is "db", not "localhost"**
> In Docker, services communicate via their name. The database is called `db` — even on Windows.

5 Manage the database

Create tables and manipulate data with phpMyAdmin

🔑 Log in to phpMyAdmin

Go to **http://localhost:8081**. Automatic connection (pre-configured).

🌐 Address	**http://localhost:8081**
👤 Username	**root**
🔑 Password	**root**
💾 Database	**app**

🛠️ Create a table

1 **In the left column, click on "app"**

This is your database

2 **"Structure" tab → "Create table"**

Give it a name (e.g.: `users`) and choose the number of columns

3 **Define your columns, then "Save"**

Always add an `id` column as INT AUTO_INCREMENT as the primary key

📊 Example "users" table

COLUMN	TYPE	LENGTH	OPTIONS
`id`	INT	11	PRIMARY KEY, AUTO_INCREMENT
`name`	VARCHAR	100	—
`email`	VARCHAR	150	—
`date_created`	DATETIME	—	DEFAULT CURRENT_TIMESTAMP

🔍 Execute an SQL query

"SQL" tab in phpMyAdmin:

```sql
-- Insert data

INSERT INTO users (name, email)

VALUES ('Alice Martin', 'alice@example.com');


-- Read all data

SELECT * FROM users;
```

6 Stop and restart

Manage the environment without the command line

☑ Stop the environment

Option A — The simplest

Double-click on `stop.bat` in the project folder. A black window opens and everything stops cleanly.

Option B — Via Portainer

1. Open `localhost:9000`
2. **"Containers"** tab
3. Check all containers
4. Click **"Stop"**

> 💾 **Your data is preserved!**
>
> Stopping the services does not delete your PHP files or your database data. Everything is preserved for the next launch.

↺ Restart a specific service

1 **Portainer → http://localhost:9000 → "Containers"**

2 **Find the container to restart**
`cours_web` = Apache+PHP | `cours_db` = DB | `cours_phpmyadmin`

3 **Click the ↺ (Restart) button**
The service restarts in a few seconds

💡 **Daily Windows routine**

Morning: Open Docker Desktop → wait for whale → double-click `start.bat`
Evening: Double-click `stop.bat`

7 Windows Troubleshooting

Solutions to the most common issues

🔴 Common issues

❌ Portainer — "timeout" page (/timeout.html)

Solution A: Double-click on `reset-portainer.bat` in the project folder.

Solution B: In CMD from the project folder:

```
docker compose restart portainer
```

Then go **immediately** to http://localhost:9000.

Solution C — Full reset: Docker Desktop → Volumes → delete `cours_portainer_data` → relaunch Portainer.

❌ Docker Desktop not detected / start.bat fails

Open **Docker Desktop** from the Start menu → wait for the whale in the taskbar → relaunch `start.bat`.

❌ start.bat closes immediately

Right-click → **"Run as administrator"**. Or open CMD in the folder and type `start.bat`.

❌ "port already in use" / "bind failed" error

Another program is using port 8080 or 8081. Close Skype, IIS, WAMP, or any other server. Restart and relaunch `start.bat`.

❌ PHP files are not displayed / not updating

Verify that your files are in `htdocs\` of the project. Refresh with **Ctrl+F5** (clears cache).

❌ phpMyAdmin — connection error

The DB takes 10 to 20 seconds to start → wait and refresh (F5 or Ctrl+R).

🔑 Credentials summary

🌐 PHP Site	`http://localhost:8080`
🗄 phpMyAdmin	`http://localhost:8081`
🐳 Portainer	`http://localhost:9000`
🏠 Dashboard	`http://localhost:8082`
📁 DB Host (in PHP)	`db`
💾 Database	`app`
👤 DB Username	`app`
🔑 DB Password	`app`
👑 Root password	`root`

Windows Tutorial — Apache 2.4 + PHP 7.4 + MariaDB 10.6 + phpMyAdmin 5.2 + Portainer CE
Windows 10 / 11 — Docker Desktop + WSL2