



Architecture & Deployment

2025-2026 v0.1.0 on branch main Rev: 03b1cdace14bb0b0720e24be862097b3792214ea

Deploy a PHP application with Git

The goal of this exercise is to deploy a PHP application much like the [previous exercise](#), but using Git to put the code on the server instead of SFTP.

Table of contents

- 📜 [Legend](#)
- 💠 [Requirements](#)
- ! [Use your own repository.](#)
- ! [Clone the repository.](#)
- ! [Update the configuration](#)
- ! [Run the PHP development server](#)
- 🏁 [What have I done?](#)
 - 🏠 [Architecture](#)



Legend

Parts of this exercise are annotated with the following icons:

- ! A task you **MUST** perform to complete the exercise
- ? An optional step that you may perform to make sure that everything is working correctly, or to set up additional tools that are not required but can help you
- 🏁 The end of the exercise
- 🏠 The architecture of the software you ran or deployed during this exercise.
- ☀️ Troubleshooting tips: how to fix common problems you might encounter

Requirements

Make sure you have completed the [previous exercise](#) and the [Git collaboration](#) exercise first.

Stop your `php -S` command if it is still running.

Tip

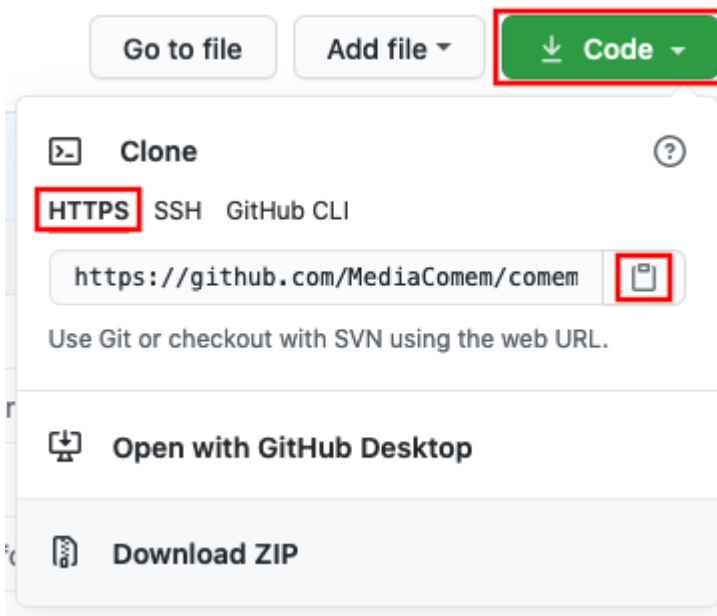
You can use Ctrl-C to stop any command currently running in your terminal.

Use your own repository

If you were not Bob during the collaboration exercise (i.e. the person who owns the repository), [create your own fork of the repository](#) so that you can modify it independently.

Clone the repository

Instead of manually uploading files through SFTP, you will connect to the server through SSH and clone the repository from GitHub. Copy your repository's public HTTPS URL:



Why the HTTP and not the SSH URL? As long as your repository is public, it is simpler to use the HTTP URL to clone it, since it requires no credentials.

To clone the repository with the SSH URL from your server, you would need to have SSH public key authentication set up on your server the same way you did on your local machine. You would need to generate an SSH key pair on the server, and add its public key to your GitHub account (or to the repository's Deploy Keys). Or you would need to put your own personal SSH key pair on the server, which would make it vulnerable in the event the server is compromised.

While connected to your server, you need to clone the repository somewhere. For example, you could clone it to the `todolist-repo` directory in your home directory.

Tip

The command to clone a Git repository is `git clone <url> [<directory-name>]`. The directory name is optional, and defaults to the last component of the URL's path without the ".git" extension. For example:

- `git clone https://github.com/bob/awesome-repo.git` will create a directory named "awesome-repo".

- `git clone https://github.com/bob/awesome-repo.git foo` will create a directory named “foo”.

! Update the configuration

Since your configuration is still hardcoded, you need to update the first few lines of `index.php` with the same configuration as for the previous exercise (`BASE_URL` , `DB_USER` , `DB_PASS` , etc).

There are several ways you can do this:

- Clone the repository locally (if you haven't already), make the change on your local machine and commit and push it to GitHub. Then connect to your server move into the cloned repository and pull the latest changes from GitHub.
- Go into the cloned repository on the server and edit `index.php` with nano or Vim, or edit it on your machine and overwrite it with FileZilla, as you prefer.

In both cases, make sure the configuration fits your server's environment.

! Run the PHP development server

Run a PHP development server on port 3000 like you did during the previous exercise, but do it in the cloned repository this time:

```
$> php -S 0.0.0.0:3000
```

You (and everybody else) should be able to access the application in a browser at the correct IP address and port (e.g. `W.X.Y.Z:3000`).

🚩 What have I done?

You are now transferring code to your deployment environment (your server) using a version control tool (Git) instead of manually, as recommended in the [Codebase section of The Twelve-Factor App](#). Always deploying from the same codebase makes it less likely that you will make a mistake like:

- Copying an outdated version of the codebase from the wrong directory.
- Forgetting to upload some of the modified files when you upload them by hand.

Using Git now also allows you to use Git commands like `git pull` to easily pull the latest changes from the repository.

🏠 Architecture

This is a simplified architecture of the main running processes and communication flow at the end of this exercise. Note that it has not changed compared to [the previous exercise](#) since we have neither created any new processes nor changed how they communicate:

