Raspberry Pi: Install Apache, MySQL, and PHP

In this guide, you will learn how to install a LAMP (Linux, Apache, MySQL, PHP) server on a Raspberry Pi computer.

LAMP is a software bundle (often called a 'stack') that is used for web development. Your Raspberry Pi will have **Raspberry Pi OS** installed, and you will also install **phpMyAdmin** to allow you to easily manage your database through a web interface.

Prerequisites

Before continuing with this tutorial:

- You will need the following hardware:
  - A Raspberry Pi 4 or Raspberry Pi 5 computer and power supply
  - A class 10 microSD card with a capacity of 16GB or more, with the Raspberry Pi OS operating system installed
- You should be familiar with the Raspberry Pi computer

You can run the below commands on your Raspberry Pi set up as a desktop computer or using an SSH connection (see page 9 to learn how to connect to your Raspberry Pi via SSH).

Update and upgrade

Before starting the installation process, open a terminal window and run the following command to update your Raspberry Pi:

```
Shell
pi@raspberrypi:~ $ sudo apt update && sudo apt upgrade -y
```

This command will ensure:

- Your computer has access to the latest package versions
- All software is up-to-date with security patches and bug fixes

Install Apache2 on Raspberry Pi

Apache2 is the most widely used web server software. In short, a web server is the software that handles requests to access a webpage. Depending on the page you have requested, the server generates the document to serve you (.html, .php, etc).

To install Apache2 on your Raspberry Pi, run the following command:

```
Shell
pi@raspberrypi:~ $ sudo apt install apache2 -y
```

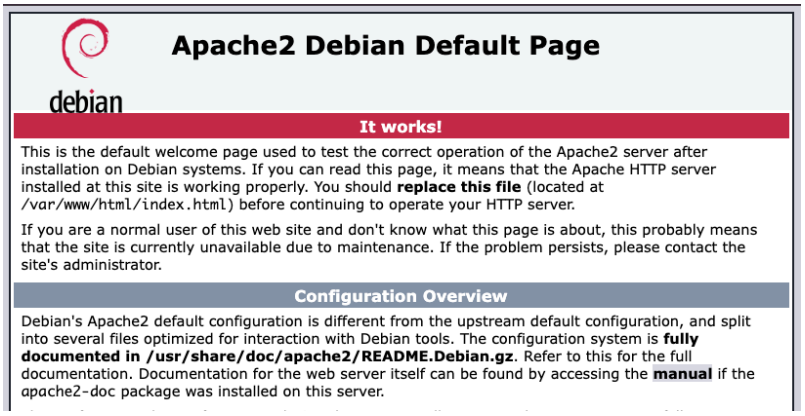
That's it! Apache is now installed. To test your installation, change to the `/var/www/html` directory and list the files:

```
Shell
pi@raspberrypi:~ $ cd /var/www/html
pi@raspberrypi:/var/www/html $ ls
index.html
```

An `index.html` file will have been created in that folder during the installation process. To open that page in your browser, you need to know your Raspberry Pi's IP address. Use the following command to display it:

```
Shell
pi@raspberrypi:/var/www/html $ hostname -I
```

In this example, the IP address is `192.0.2.0`. If you open your Raspberry Pi's IP address in any browser in your local network (for example, `http://192.0.2.0`), a similar webpage should load:



Install PHP on Raspberry Pi

PHP (PHP: Hypertext Preprocessor) is a server-side scripting language that is used to develop dynamic web applications. A PHP file contains `<?php … ?>` tags and ends with the extension '.php'.

To install PHP on your Raspberry Pi, run the following command:

```shell
pi@raspberrypi:/var/www/html $ sudo apt install php -y
```

To test the installation of PHP, remove the `index.html` file and create a PHP script called `index.php`:

```shell
pi@raspberrypi:/var/www/html $ sudo rm index.html
pi@raspberrypi:/var/www/html $ sudo nano index.php
```

In your `index.php` file, add the following code to echo the "hello world" message:

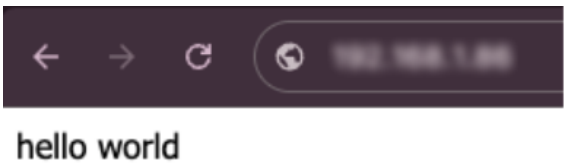
```php
<?php echo "hello world"; ?>
```

To save your file, press Ctrl + X, followed by Y, and press Enter to exit.

Finally, restart Apache2:

```shell
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
```

To test if Apache2 is serving .php files, open your Raspberry Pi's IP address in a browser. It should display the "hello world" message from the `index.php` script created earlier.



If everything is working, you can remove the `index.php` file from the `/var/www/html` directory:

```
Shell
pi@raspberrypi:/var/www/html $ sudo rm index.php
```

Install MariaDB Server

MariaDB Server is a popular database management system. Many PHP websites use a backend database to store information, so these steps guide you through installing and configuring MariaDB Server.

Type this command to install the MariaDB Server and PHP-MySQL packages:

```
Shell
sudo apt install mariadb-server php-mysql -y
```

Now restart Apache:

```
Shell
sudo service apache2 restart
```

MariaDB Server setup

Run the MySQL secure installation command in the terminal window.

```
Shell
sudo mysql_secure_installation
```

Go through the setup wizard using the following answers:

- [ ] “Enter current password for root (enter for none):”: Press Enter
- [ ] “Switch to unix_socket authentication [Y/n]”: Type N and press Enter (this may not appear on modern installations)
- [ ] “Change the root password?”: Type Y and press Enter
- [ ] “New password:”: Type a password and press Enter, then repeat a second time — **Important: Remember this root password, as you will need it later**
- [ ] “Remove anonymous users”: Type Y
- [ ] “Disallow root login remotely”: Type Y
- [ ] “Remove test database and access to it”: Type Y
- [ ] “Reload privilege tables now”: Type Y

When setup is complete, you will see the message “All done! and Thanks for using MariaDB!”

Optional: If you experience any errors logging in to phpMyAdmin, you may need to create a new user to log in. The following commands will create a new user with a name (admin) and password (your_password).

```
Shell
pi@raspberrypi:/var/www/html $ sudo mysql --user=root
--password
> create user admin@localhost identified by 'your_password';
> grant all privileges on *.* to admin@localhost;
> FLUSH PRIVILEGES;
> exit;
```

Install phpMyAdmin on Raspberry Pi

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL using a web interface.

To install phpMyAdmin on your Raspberry Pi, type the following command into the terminal:

```
Shell
pi@raspberrypi:/var/www/html $ sudo apt install phpmyadmin -y
```

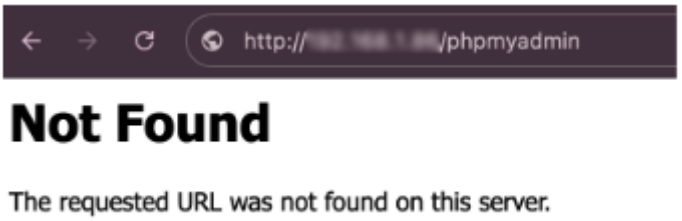
The phpMyAdmin installation program will ask you a few questions. You will use `dbconfig-common`, a Debian package tool that simplifies setting up and managing databases for applications (like phpMyAdmin) during installation. It automatically creates the phpMyAdmin database and user, sets the right permissions, imports the necessary tables, and updates the config files so phpMyAdmin can connect and run properly. Follow the steps below to complete the installation:

1. When prompted, select 'Apache2' and press the Enter key
2. When "Configuring phpmyadmin?" is displayed, select 'OK'
3. When "Configure database for phpmyadmin with dbconfig-common?" is displayed, select 'Yes'
4. Type your password and select 'OK'
5. Confirm your password and select 'OK'

Enable the PHP MySQLi extension and restart Apache2 for the changes to take effect:

```
Shell
pi@raspberrypi:/var/www/html $ sudo phpenmod mysqli
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
```

When you visit your Raspberry Pi's IP address followed by `/phpmyadmin` in a browser (for example, `http://192.0.2.0/phpmyadmin`), you may see the "Not Found" error page:



If that is the case, move the `phpmyadmin` folder to `/var/www/html` by running the following command:

```
Shell
pi@raspberrypi:/var/www/html $ sudo ln -s
/usr/share/phpmyadmin /var/www/html/phpmyadmin
```

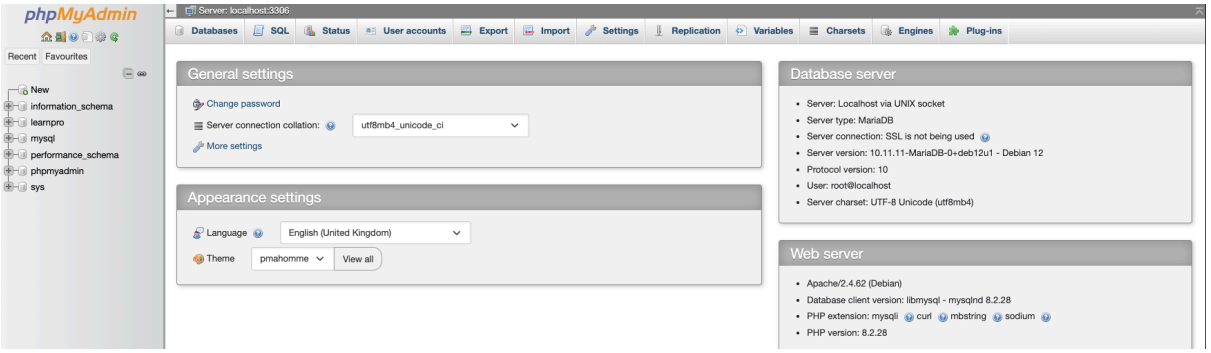
Now, if you list the files using `ls`, it should show the `phpmyadmin` folder:

```shell
pi@raspberrypi:/var/www/html $ ls
phpmyadmin
```

Reload your webpage (for example, `http://192.0.2.0/phpmyadmin`). You should see the login page for the phpMyAdmin web interface:

Enter your defined username (it should be `root`) and the password you defined during the installation.

Press the 'Log in' button. A new page will load:



That's it! Your Raspberry Pi is prepared with a LAMP server, including Apache2, MySQL, and PHP. Your server also includes phpMyAdmin for easier database management through a web interface.

Optional: Change folder permissions

To manage your webpages, you should change the permissions for your `/var/www/html/` folder. To do this, run the following commands:

```
Shell
pi@raspberrypi:~ $ ls -lh /var/www/
pi@raspberrypi:~ $ sudo chown -R pi:www-data /var/www/html/
pi@raspberrypi:~ $ sudo chmod -R 770 /var/www/html/
pi@raspberrypi:~ $ ls -lh /var/www/
```

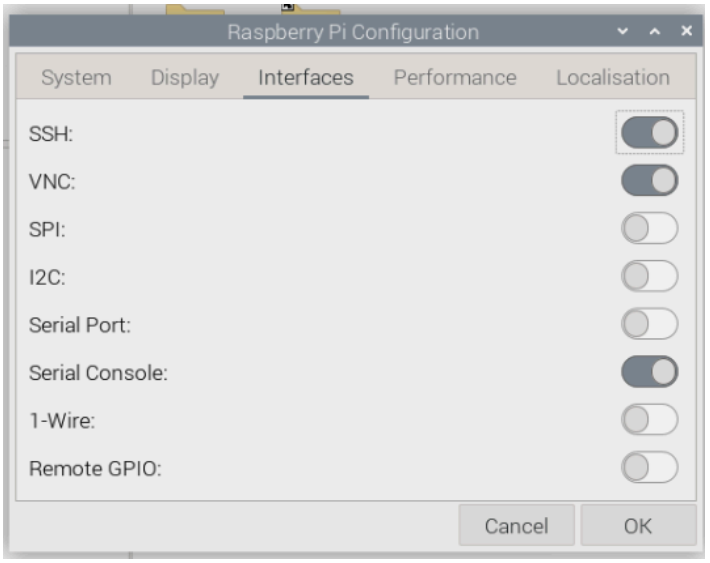
Why this is needed:
By default, the `/var/www/html/` folder (where your website files are stored) is owned by the `root` user, which means you might not have permission to edit or manage the files easily. These commands change the folder's ownership so that **you (the `pi` user)** and the **web server (`www-data`)** can both read, write, and run files in that folder. This makes it easier to update your webpages while still keeping things secure.

Optional: Connect Visual Studio Code to a remote Raspberry Pi computer using SSH

If you are using Visual Studio Code as your IDE, you may want to set up SSH to connect to your remote Raspberry Pi server.

SSH (Secure Shell) is a network protocol used to securely connect to another computer over the internet or a local network. It allows you to access the command line of a remote machine (like your Raspberry Pi) and control it as if you were sitting in front of it. SSH is commonly used in computing and IT because it encrypts the connection, keeping your data and passwords safe from hackers. This makes it a reliable way to manage servers, update files, or run programs remotely.

In your Raspberry Pi Configuration menu, ensure you have enabled SSH:



(You may be asked to restart your Raspberry Pi after enabling this.)

Check you can connect via SSH to your Raspberry Pi using the terminal first. Type the following into your command prompt (using the IP address of your Raspberry Pi):

Shell

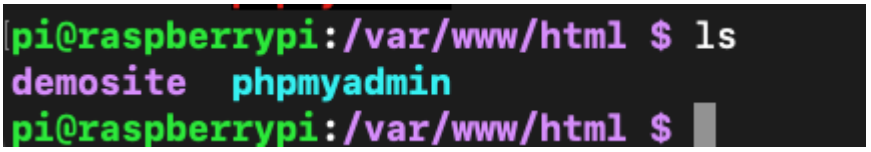
```
ssh pi@192.0.2.0
pi@192.0.2.0's password: [type the password for your Raspberry Pi]
```

```
laura.james@LauraJames-MBA-M3 ~ % ssh pi@192.0.2.0
pi@192.0.2.0's password:
Linux raspberrypi 6.12.25+rpt-rpi-2712 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1
025-04-30) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 25 11:42:12 2025 from 192.0.2.0
pi@raspberrypi:/var/www $
```

Navigate to the folder that will store the project and list the files and folders:

Shell

```
cd /var/www/html
ls -a
```



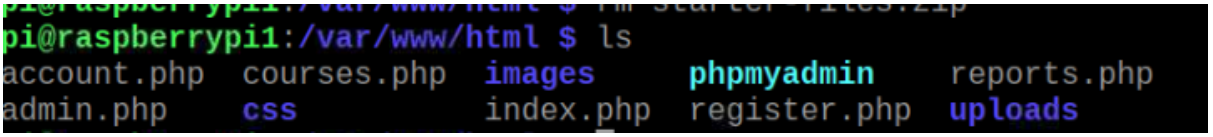
If you can see a list of the remote files on your Raspberry Pi, you know you can successfully connect to it over SSH.

Download the starter files over SSH

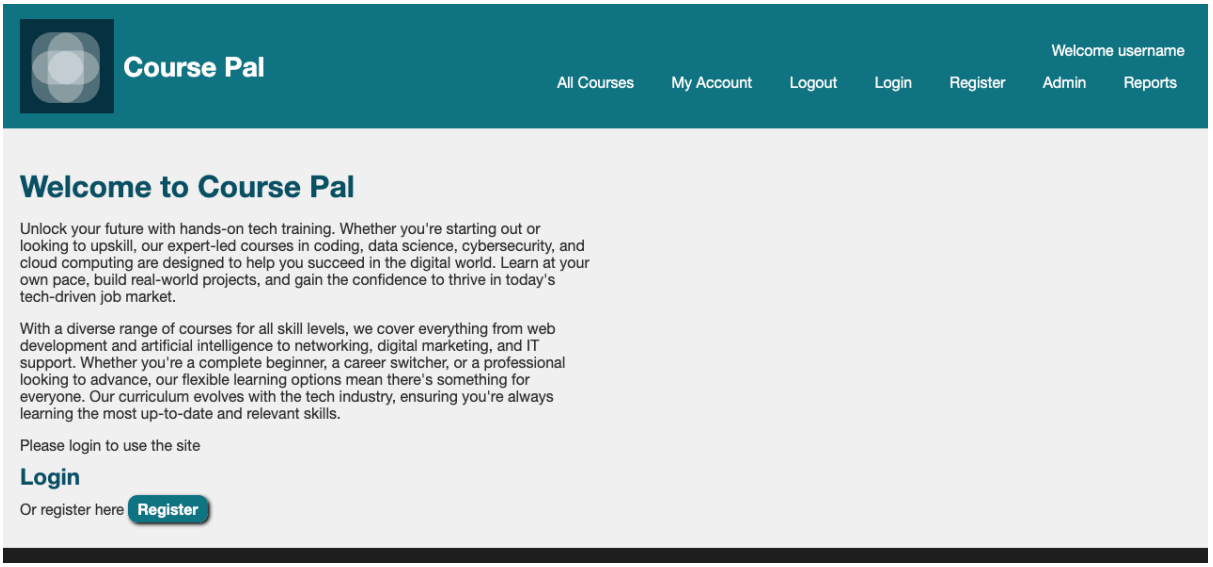
Run the following commands to download the zipped starter files from the given URL, unzip them into the current folder, and remove the downloaded zip file:

```shell
cd /var/www/html
wget
https://cdn.adacomputerscience.org/ada/example_websites/web_project_course_p
al_starter_files.zip?new -O starter-files.zip
unzip starter-files.zip
rm starter-files.zip
```

When you list the files in the `/var/www/html` directory, you should see the following:



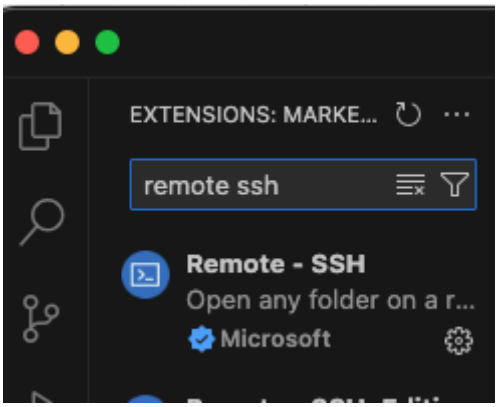
When you visit the home page of your Raspberry Pi, you should see the Course Pal starting home page:



Set up Remote - SSH in Visual Studio Code

If you are using VS Code as your code editor, you can install a free extension called **Remote - SSH** to allow you to connect to your remote Raspberry Pi server and edit the files as if they were on the computer in front of you.

In the VS Code extensions library, search for **Remote - SSH** and install it.



Use Remote - SSH with a startup command (manual setup)

You can create a **custom SSH config** for VS Code that sets the start folder, which will be the starter folder for your website project. This makes it easier to use as it will navigate to the folder where you are storing your website files the first time you connect to the server.

1. Open VS Code.
2. Press F1 or Ctrl + Shift + P on Microsoft Windows or Cmd + Shift + P on macOS, then search for “Remote-SSH: Open SSH Configuration File”.

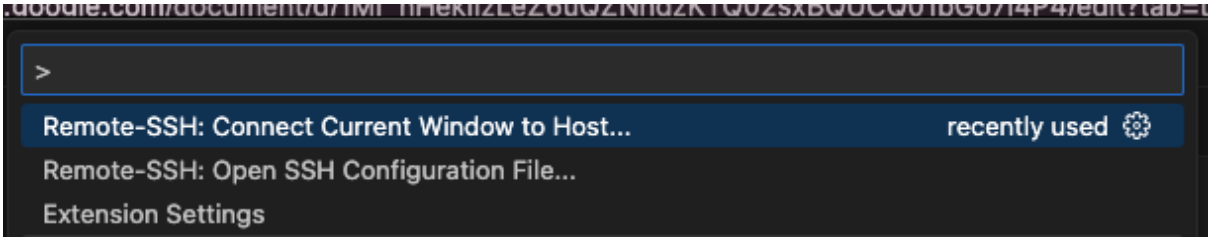
3. Select the SSH config file (likely `~/.ssh/config`), and add an entry like this:

```
Host raspberrypi

    HostName raspberrypi.local
    User pi
    RemoteCommand cd /var/www && exec bash
```

Or, for IP-based SSH:

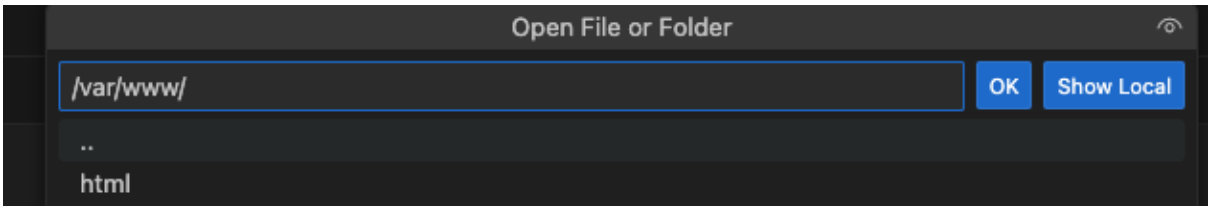
```
Host rpi
    HostName 192.0.2.0
    User pi
    RemoteCommand cd /var/www && exec bash
```

4. Save the file.
5. Open VS Code and press F1 or Ctrl + Shift + P on Microsoft Windows or Cmd + Shift + P on macOS, then search for “Remote-SSH: Connect Current Window to Host…” and select it in the menu.

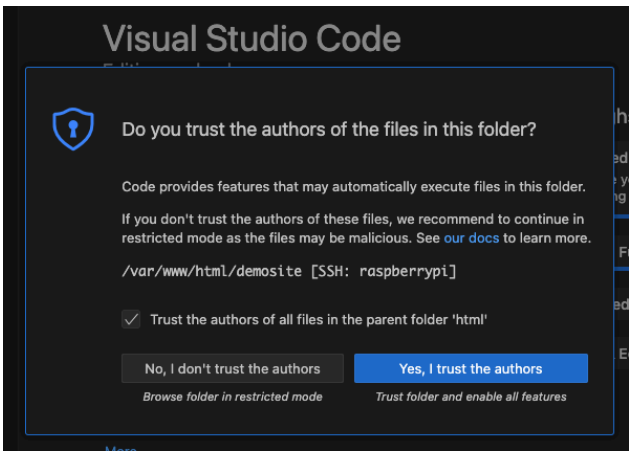


6. Then, select “raspberrypi” (you will be prompted for your password).

VS Code will now open with the terminal starting in `/var/www`.



The first time you connect, you will need to click the button that says you trust the authors of the files in this folder:


Visual Studio Code
Do you trust the authors of the files in this folder?
Code provides features that may automatically execute files in this folder.
If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.
/var/www/html/demosite [SSH: raspberrypi]
Trust the authors of all files in the parent folder 'html'
No, I don't trust the authors
Browse folder in restricted mode
Yes, I trust the authors
Trust folder and enable all features