Lab 2: Deploying a LAMP stack on the Raspberry Pi

# Objectives

At the end of this lab, students should have a fully functional web server deployed on their Raspberry Pis and accessible from anywhere on the local network.

# Preparation

You will need to install Raspberry Pi OS (formerly known as Rasbian) for this course, a Linux distro based on Debian Bullseye. Note that this is not the only OS your Raspberry Pi supports; a partial list of them appears [here](https://www.raspberrypi.org/software/operating-systems/). The Raspberry Pi runs off an SD card (although it may be configured to boot from SSD).

You have two main options for installation:

* Install with NOOBS (New Out Of the Box Software). If you have purchased a Raspberry Pi kit that comes with an SD card, it is likely pre-configured with NOOBS.
* Install using the [Raspberry Pi imager](https://www.raspberrypi.com/software/)

This procedure will assume you are using the Raspberry Pi Imager as it is the easiest way to image your SD card and preconfigure the OS. Further instructions can be obtained [here](https://www.raspberrypi.com/documentation/computers/getting-started.html).

**Throughout this lab, be sure to remember all credentials (i.e. usernames/passwords) you configure. Note them and store them securely (ideally, in a password manager).**

* 1. Obtain the [Raspberry Pi Imager from the Files tab in Teams](https://collegedawson.sharepoint.com/:u:/r/sites/fundamentalsofwebservers_9687/Class%20Materials/imager_1.7.5.exe?csf=1&web=1&e=ZOQpxN).
  2. Insert your SD card into your PC and from the Imager, select Raspberry Pi 64-bit as your OS and the location of your SD card.
  3. Click the gear icon:
     1. Set a unique hostname to differentiate your Pi on the network
     2. Enable SSH
     3. Configure a username and password
     4. Set your wireless LAN country to CA
     5. Set your locale

Graphical user interface, text, application

Description automatically generated

* 1. You can now remove the SD card, place it in your pi, and power on.
  2. ssh into your Pi using the hostname and credentials you configured OR simply connect it to peripherals (monitor, keyboard, mouse).
  3. Enable VNC:
     1. Run raspi-config as root.
     2. Enable VNC through the Interface sub-menu.
     3. Change the VNC display resolution through the Display sub-menu to something suitable, like 1920x1080.
     4. Create a free [VNC account](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi999LDha_6AhXOMVkFHUVvB8YQFnoECAgQAQ&url=https%3A%2F%2Fwww.realvnc.com%2Fen%2F&usg=AOvVaw1ckuoR--7UEUuP9_5PeEVu) (not a trial!) so that you can log in to your Pi remotely. You will need to log into your account on both the Pi and the client to use this feature. Test it now on the lab computer.
  4. Perform a package update:

sudo apt update && sudo apt upgrade -y && sudo apt autoremove -y

You can add this as an alias to your ~/.bashrc file.

* 1. Optionally, install zsh:

sudo apt install zsh -y

sh -c "$(curl -fsSL https://raw.github.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"

* 1. When you are at home, you may wish to do the following:
     1. While it is possible to [give your Pi a static IP](https://www.raspberrypi.org/documentation/configuration/tcpip/), you will be switching the Pi between home and Dawson networks, so a router DHCP reservation is best if you know the admin credentials for your router.
     2. To login to your Pi using only the hostname (e.g. ssh piinthesky), you can add the following to your ~/.ssh/config file on your local machine, modifying as appropriate for your configuration:

Host piinthesky

HostName piinthesky.local

User pi

* + 1. If applicable, encrypt your WPA passphrase from the command line and paste the result (making sure to delete the commented line with your actual passphrase) into /etc/wpa\_supplicant/wpa\_supplicant.conf, so your WiFi password is not hanging around in plaintext:

wpa\_passphrase [ssid] [passphrase]  
sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf

* + 1. Configure [passwordless access](https://www.raspberrypi.org/documentation/remote-access/ssh/passwordless.md).

# Procedure

* 1. [Install the LAMP stack](https://projects.raspberrypi.org/en/projects/lamp-web-server-with-wordpress) *without* WordPress. STOP when you get to the WordPress instructions. You are not required to install WordPress (although you can certainly experiment with it on your own).
  2. Ensure that you can access your web server remotely from your computer by entering the hostname of your Pi in the browser’s address bar (e.g. <http://piinthesky.local>). Take the time to read through the displayed page. It will help you answer some of the questions in this lab and perform the next step.

A screenshot of a computer

Description automatically generated

* 1. Change the default Apache port from 80 to 8080 by following the instructions on the page from the previous step. Restart the Apache service (you should know the command to do this now) and test by navigating to (for example) <http://piinthesky.local:8080> in your local browser.

A screenshot of a computer

Description automatically generated

* 1. Install the Apache 2 manual using: sudo apt install apache2-doc -y. You can now access it at <http://localhost/manual> from your Pi (or remotely using the hostname and port you configured).
  2. Create the file index.php in your web server root. Add the line:

<?php phpinfo(); ?>

Ensure that you can access it by navigating to (for example) <http://piinthesky.local:8080/index.php>

* 1. Read through this file and note in your report what type of information is displayed. Be sure to note Apache environment variables and http header information. Note the PHP version and user-agent string. A screenshot of a computer

     Description automatically generated

A screenshot of a computer

Description automatically generated





* 1. Browse to the same location, but from a different machine. Is the information displayed identical as in the last step? Note the user-agent string and any other differences.





**Besides the user agent being different depending on the machine you accessed the site, the remote address will also correlate to the devices ip address.**

* 1. Rename index.html in your web server root to apache.html. Navigate to your web server again without specifying a file path, e.g. <http://piinthesky.local>. What file displays now by default? Show that you can access the renamed file from your browser.

A screenshot of a computer

Description automatically generated

**It shows a blank page.**

* 1. Install git on your Pi from the command line and clone your repository:
     1. On your Pi, ensure command-line git is installed: apt install git -y
     2. Navigate to /var/www/html.
     3. Clone your repository: git clone https://github.com/user/repo (sub your user and repository name). This will create a new directory with the name of your repository. You can change the name of the folder.

A computer screen with blue text

Description automatically generated

* + 1. On your computer, ensure that you can access your repository in a browser over http at the port you configured (8080).
  1. Demonstrate to your instructor or include screenshots in your report submission.

A screenshot of a computer

Description automatically generated

# Questions

* 1. Discuss the differences between the files index.html and index.php.
     1. Open both in a text file or (or cat to the terminal). Note the differences between them (length, format, language, etc.).

**In the index.html file and index.php are both treated differently, where the index.html is treated as an html file where as index.php is treated as a php file. This also changes the formatting and language within the file where we see the standard html format on the index.html, we see php format on index.php**

* + 1. Note the differences between the files in terms of the html each file sends to the browser by viewing the page source relative to the content of the file on the server.

**On the php file , we added a dynamic php content where the server is pulling all the information from the device where as html were mostly static, meaning manually inputed.**

* + 1. Explain how this illustrates the difference between static and dynamic web pages.

**Dynamic web pages takes info from elsewhere or the server to create a webpage for you where static is all manually inputted by the admin/user.**

* + 1. Explain how Apache chooses which file to show by default when none is specified. Locate the Apache configuration file that governs this behavior and paste the relevant setting below.

**It seems like the Apache chooses the last file seen in its file**

* 1. Describe a technical issue that you encountered during this lab and explain how you diagnosed and solved it. List the steps that you took (in list and not paragraph format).

**I did not encounter any issues as a continued through this lab besides some minor inconvenience such as not having space to connect to the network or the hassle of creating VNC account.**