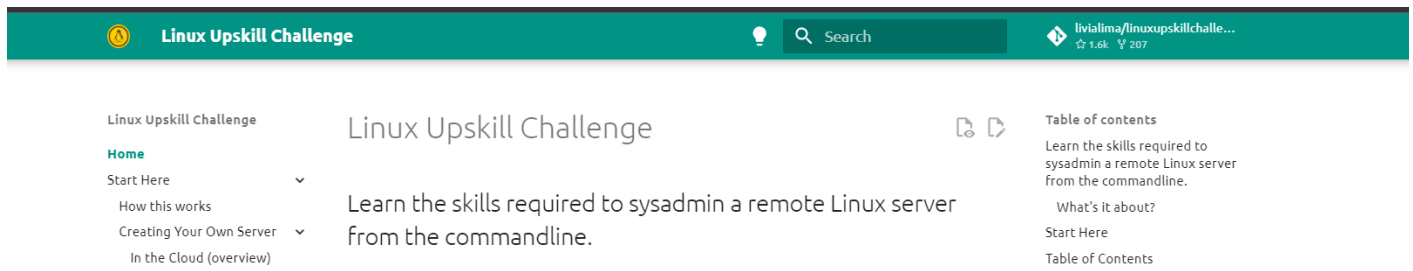# The Linux Upskills Challenge

Course Overview

We begin our journey into Systems and Programming with the **Linux Upskill Challenge** (**GitHub**)

This challenge is a great way to take our first steps into Linux. It is a month long commitment that you can follow at your own pace, but if you dedicate 2-3 each day it will take you roughly 21 days to complete the challenge.
For our purposes we will be using the Challenge to as a way to learn Linux for the first time and to also begin building on our Linux Command Line Skills.

Goals of this Course

- Set up your own Linux Server either **on the cloud** or **locally** on your own machine
- Learn to Navigate the **Linux File System** & learn about the **Linux File Hierarchy Structure** (**LFHS**)
- Cover the basics of the powerful **vim** text editor
- Set up your own web server with **Apache** and set up some traffic rules with ufw
- Understand how to filter information with commands like grep, cat, more, less, cut, tail, etc.
- Learn about various file transfer protocols from SMB to SFTP and more
- Grasp the basics of permissions involving users and groups
- Ubuntu Universe vs Multiverse
- Learn about Archiving & Compressing and how to build from source
- Inodes, symlinks and other shortcuts
- Get a brief introduction to shell scripting

Course Notes

The Linux Upskill Challenge has a lot of great concise and detailed explanations for all of the topics they cover. Which makes it pretty hard to make any notes that wouldn't just be a copy & paste.
As a middle ground the following notes are commands and information I found particularly useful.

> A quick note for people who are windows users!

> Some of you may be using WSL to access your cloud instance. You might experience clock-drift on your device making you unable to access to the cloud CLI. If this happens try running the following command in the block below.
> This should serve as a temporary fix to your clock-drift but I would recommend looking online for a more permanent solution.

```
sudo hwclock -s
```

The ls command and its switches
- (-l) : lists directory contents
- (-a) : lists hidden files
- (-t) : lists by date created
- (-r) : lists in reverse order

Some notes on Sudo & Root Best practices
- Don't leave root open
- Be careful of using sudo to open GUI applications, if you have to, use sudo -h
- /etc/shadow : A very important space that stores information about the user passwords on a Linux system
- Hackers can't auto-crack if root is locked down

Commands used to find files
- locate | find | grep | which

Ownership and Permissions in Linux
- ls -ltr is a good way of showing who owns what files and what permissions they have
- chmod
  - u = user | g = group | o = others | a = all
  - (+) = add permission | (-) = remove permission | (=) = set permission
  - r = read | w = write | x = execute
  - u = user | g = group | o = others | a = all
  - 1 = execute | 2 = write | 4 = read

examples:
  - chmod u+w hello.txt | Gives user write permissions to the hello.txt file
  - chmod 777 hello.txt | Gives read write and execute permissions to everyone

Important File Transfer protocols
Linux can share files through a variety of protocols. These 3 are key protocols to keep in mind
  - SMB: Microsoft file sharing, useful on a local network of Windows Machines
  - AFP: Apples file sharing, useful on a local network of Apple Machines
  - SFTP: Access & transfer files over SSH

You can get practice with vim by running vimtutor
Turn a file into a bash script with #!/bin/bash

Required Reading

The required reading section in the Linux Upskill Challenge covers all the material you need to get started. I added setting up SSH login as required since the challenge lists it as an extension.

- **Passwordless SSH Login**

Recommended Reading

Similar to before, the recommended readings posted on the challenge are very good and definitely worth going through. The 2 articles below are extensions that I found particularly interesting or practical.

- **(Article) What does free mean**
- **(Article) Moving SSH Port**

Section Projects

Project 1 - Take your first steps into the world of Self-Hosting

What is Self-hosting?
Self-hosting is the practice of running and maintaining a website or service using a private web server, instead of using a service outside of someone's own control.

We should now be knowledgeable enough about Linux to comfortably follow tutorials on self hosting.
The world of self hosting is a rabbit hole and there are more services out there than you will ever have time to understand and use, which also means that there's bound to be something for everyone.

> For this project your task will be to **self host 3 services**

The first 4 services I listed are the ones I first set up when I finished the challenge. WordPress and Pi-hole were relatively straightforward but getting Grafana & Prometheus to work was definitely a challenge.
I also linked to the Awesome Selfhosted GitHub page that shows 100's of services you can self host.

- **WordPress** - WordPress was originally a blogging platform but then evolved into a **CMS** which stands for **Content Management System**. A CMS is a more general tool that allows you to build and publish websites that can be anything from forums, online stores, media galleries, blogs and so on.

  In my case I use WordPress to host my blog **01blu3.com**.

  In the attached link you'll be following **Digital Ocean's guide** on **how to set up a blog**. Setting up WordPress is a perfect start to self hosting because you'll be exposed to the

LAMP stack. LAMP standing for **Linux, Apache, MySQL & PHP**. These are a set of 4 software technologies that are used together to build Websites and Web Applications. These following two articles by AWS and IBM will give a good overview of why we use these technologies together.

- **What is a Lamp Stack? | AWS**

- **What is a Lamp Stack? | IBM**

We will be covering Linux, Apache and nginx servers, SQL Databases and scripting more in depth in the future. The main thing to keep in mind, is that as sysadmins we will end up deploying, monitoring, optimizing and hardening these systems at both small and large scales.

- **BookStack** | **MediaWiki** | **Outline** - The following 3 services are open-source wiki software. I would definitely recommend hosting a wiki software for yourself, what better way to organize your notes and links than with a personal wiki?

  Wiki software allows you to collaborate with others to create and edit pages or entries via a web browser. These wikis can also be used privately as well.
  Every wiki has their own style and way of doing things. If none of these 3 suit your style I encourage to look for other wiki software that fits your needs and preferences.

  Bookstack - The software I use to host this lab-book. Setup is pretty straightforward if you're doing a fresh install but if you're not, things can get difficult. I recommend following the **video walkthrough** made by the BookStack creator himself. He uses Debian 10 in this video but much of what he does is applicable to Ubuntu 22.04.

  MediaWiki - The very wiki software that runs Wikipedia! A lot more old school but very feature rich and powerful. There is a learning curve to managing it and some may find it easier to work with than others.

  Outline - A very beautiful open-source wiki. However the biggest downside is that hosting it will prove a difficult challenge as there are many steps involved in setting it up.

- **Grafana & Prometheus** - Setting up Grafana and Prometheus will serve as an introduction to the world of data visualization and server monitoring. You can take a whole course on these 2 alone so only take this on if you're looking for a challenge.
  Grafana is an open source analytics and interactive visualization web application. Essentially **Grafana** is a powerful data visualization tool that can work with a large variety of data sources.
  Prometheus is a free software application used for event monitoring and alerting. This allows you to monitor the state of your systems based on data generated by the system. This data can be anything from system temperature to CPU usage, network statistics and so on.

  The **Prometheus Fundamentals** YouTube playlist by Julian will give you what you need to set up Prometheus together with Grafana.

- **Pi-Hole** - If you happen to have a Raspberry Pi lying around I would definitely recommend running Pi-Hole on it.
  To summarize, Pi-hole is a network wide ad blocker. It's main use-case is for blocking web ads, trackers and telemetry collection. While it won't block browser ads it should cover 3rd party ads, general website ads, mobile game ads and malicious traffic. It's definitely a worthwhile service.

  The **Pi-hole Tutorial** YouTube video by Crosstalk Solutions should have you covered.

- **Awesome-Selfhosted** - Lastly, you may not actually be interested in any of the services I mentioned. Although I still recommend you host a **CMS** and a **Wiki software**. However besides, that there are a world of things you can try setting up on your own and the Awesome Selfhosted GitHub page has you covered.
  This is a list of over a 100 services covering just about any use case you can imagine. I definitely recommend you check it out!

Revision #38
Created 31 October 2023 10:13:25 by 01Blu3
Updated 16 January 2024 22:13:06 by 01Blu3