Ubuntu System Administration Guide

Ubuntu Desktop, Server, security, and DevOps automation

Mattias Hemmingsson

www.bpbonline.com

First Edition 2025

ISBN: 978-93-65895-124

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true and correct to the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but the publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

This book is dedicated to you, the reader who has opened it and is taking the first steps on your Linux journey.

I am honored to be a part of that adventure with you.

My family and friends, thank you for your unwavering support and encouragement throughout this incredible challenge.

About the Author

Mattias Hemmingsson is a seasoned IT professional with over two decades of experience in designing, building, and securing modern IT infrastructure. With deep roots in cloud computing, DevOps, and system architecture, he has successfully led the development and operations of high-availability systems for mission-critical environments, especially within the financial sector, where stability and performance are non-negotiable.

Mattias specializes in a wide range of technologies, including Kubernetes, Docker, Jenkins, GitLab CI, and cloud platforms such as AWS and Google Cloud. His skill set spans server orchestration, infrastructure as code, and containerized application delivery. A true advocate for developer productivity, he has implemented CI/CD pipelines, automated monitoring systems, and scalable platform services that have helped teams deploy faster with greater reliability.

Security is a cornerstone of Mattias's expertise. He has worked extensively in IT security and compliance, securing high-performance payment gateways and writing security policies to meet industry and regulatory standards. He actively manages vulnerability assessments, runs enterprise-grade security scanners, and participates in incident response operations. His security knowledge bridges the technical and procedural, ranging from configuring firewall rules to training development teams in secure coding practices.

As a full-stack developer, Mattias works with both Python and JavaScript, building frontend and backend systems using popular frameworks like Next.js and Vue. He brings a unique blend of hands-on coding ability and infrastructure mastery, making him equally at home writing applications or deploying them in scalable, production-ready environments.

Mattias also plays an active role in the tech community. He co-hosts the DevSecOps Talks podcast and YouTube channel, where he shares insights on the intersection of development, operations, and security. He teaches classes on DevSecOps practices and serves as the local organizer for one of Stockholm's tech meetups, helping others stay informed and connected in the ever-evolving world of IT.

Through his blog, Life and Shell, Mattias documents his professional journey, sharing practical tips and deep dives into real-world solutions for fellow developers, sysadmins, and security professionals. His work reflects a passion not just for running services, but for securing them, ensuring they are both compliant and resilient.

Whether building infrastructure, writing code, or leading training sessions, Mattias brings a comprehensive understanding of modern systems from their initial setup to their long-term security and scalability.

About the Reviewers

- **Nathan Molete** is a skilled Linux DevOps engineer with a strong background in cloud computing, automation, and infrastructure management. With extensive experience in Linux systems administration, networking, and application deployment, he specializes in ensuring the reliability, scalability, and security of enterprise IT environments.

  Nathan is proficient in automation tools like Ansible and Terraform, enabling efficient configuration management and **infrastructure as code** (**IaC**) practices. He has expertise in containerization technologies, including Docker and Podman, as well as orchestration platforms like Kubernetes and Minikube. His hands-on experience with CI/CD pipelines, using Jenkins and Git, allows him to streamline software delivery and improve operational efficiency.

  Working in both production and test environments, he is responsible for maintaining optimal system performance, troubleshooting complex issues, and implementing best practices for high availability and security. His knowledge spans across web servers such as Apache and NGINX, database management with MySQL and PostgreSQL, and cloud platforms like Azure.

  Passionate about open-source technologies, Nathan continuously seeks to enhance systems and workflows through automation and innovation. With a detail-oriented and problem-solving mindset, he plays a crucial role in optimizing IT infrastructure, ensuring seamless operations, and driving digital transformation within organizations.

- **Cyrus** is a software developer, educator, and content creator with expertise in React, Next.js, and modern web technologies. He has developed educational platforms, built scalable web applications, and created a LinkedIn Learning course, Testing in React with Vitest, completed by over 1,000 learners. Passionate about empowering others, Cyrus shares insights on testing performance optimization and cutting-edge tools, helping developers build smarter, more efficient applications.

Acknowledgement

I want to express my sincere gratitude to all those who contributed to the completion of this book.

First and foremost, I extend my heartfelt appreciation to my family and friends for their unwavering support, patience, and encouragement throughout this journey. Your belief in me has been a constant source of strength.

I am especially thankful to my colleagues and friends from Fareoffice, Enterprise Car Rental, and Booli. Your valuable input, insights, and constructive feedback have been instrumental in shaping the direction and content of this book.

A special note of thanks goes to my close friends and neighbors for your patience in listening, your thoughtful feedback, and your enduring support. Your honest perspectives helped refine and elevate the quality of the work.

I am also deeply grateful to BPB Publications for their expert guidance and dedication in bringing this book to life. Your professionalism and support made the publishing process a smooth and rewarding experience.

I would like to sincerely appreciate the reviewers, technical experts, and editors who contributed their time and expertise to reviewing this manuscript. Your insightful comments and suggestions have significantly enhanced its clarity, depth, and accuracy.

Lastly, to the readers, thank you for your interest and support. Knowing that this book may inform, inspire, or assist you is the most rewarding outcome of all.

To everyone who played a part in making this book a reality, thank you.

Preface

Linux is swiftly establishing itself as the foundation of today's digital infrastructure, playing a central role in everything from personal desktops and enterprise-grade web servers to cutting-edge cloud platforms, smart home devices, and embedded IoT systems. Its presence extends into nearly every corner of computing. Whether you are browsing a website, deploying an app in the cloud, or working on a development project, chances are that Linux is working behind the scenes.

For aspiring developers, IT professionals, and system administrators, learning Linux is not just an option, it is a strategic advantage. Seasoned professionals often recommend starting with Linux before venturing into other tools such as Docker, Kubernetes, or even advanced cloud-native frameworks. Why is that? Because Linux provides deep insight into how operating systems function. It teaches you how processes run and interact, how file systems are organized, how networks are structured and secured, and how permissions and users are managed. These are the core concepts that underpin all modern computing tools.

By mastering Linux, you develop a strong mental model of how systems operate—one that allows you to troubleshoot more effectively, script more efficiently, and deploy with greater precision. It builds the kind of confidence that lets you approach complex technologies like container orchestration, virtualization, and continuous integration pipelines with a solid understanding of what is really happening behind the scenes.

This book is centered on Ubuntu, one of the most accessible and widely adopted Linux distributionsributions in the world. Ubuntu is known for its balance of power, simplicity, and versatility. It delivers the full capabilities of Linux in a package that is approachable for newcomers yet robust enough for professionals managing enterprise environments. With its regular updates, strong community support, and broad compatibility with both open-source and commercial software, Ubuntu serves as an ideal entry point into the Linux ecosystem.

Whether you are looking to replace your current operating system and use Ubuntu as your daily desktop driver, or you are interested in deploying servers, managing virtual machines, hosting applications, or building your own cloud infrastructure, this book will walk you through everything you need to get started. Ubuntu provides a secure, stable, and customizable platform that can grow with your needs, supporting everything from basic productivity tasks to advanced server-side operations, software development, and DevOps workflows.

In short, learning Ubuntu and Linux opens doors. It gives you the freedom to explore, experiment, and take control of your computing environment. It lays the groundwork for further exploration into the vast world of open-source technologies.

This book is your roadmap to getting there, with practical guides, hands-on exercises, and real-world use cases that will help you build both confidence and capability as a Linux user.

Chapter 1: Getting Familiar with Ubuntu Ecosystem - This chapter introduces the Linux and Ubuntu ecosystem, starting with the history of Linux and its evolution into one of the most widely used operating systems. It explores the Linux stack, popular distributions, and Ubuntu's development, release cycle, and various editions. The chapter highlights the growing importance of Linux in servers, IoT, and cloud computing. Readers are also introduced to key Ubuntu-based systems like Mint and Pop!_OS, and other Linux distributionsros like Red Hat and Alpine. To prepare for upcoming hands-on exercises, the chapter concludes by guiding readers to create GitHub and Blogger accounts for saving code and documenting progress.

Chapter 2: Install, Upgrade, and Configure Ubuntu Desktop - This chapter guides readers through installing and configuring Ubuntu Desktop. It begins with downloading the Ubuntu ISO, creating a bootable USB using tools like Etcher, and performing installation via USB or dual-boot with Windows. It covers pre-installed Linux options, basic troubleshooting, and verifying hardware compatibility. The chapter also introduces software installation using the Ubuntu Software Center, Snap Store, and terminal commands. Readers learn how to update and upgrade Ubuntu, manage apps, and back up configurations using the .config folder and Git. By the end, users will have a fully functional Ubuntu system ready for further customization.

Chapter 3: Environments and Window Managers - This chapter explores customizing Ubuntu with the i3 tiling window manager, offering efficient window navigation through keyboard shortcuts. It covers installing and configuring i3, setting screen resolutions, adding custom startup scripts, and improving usability with tools like sound controllers, screenshot utilities, and lock screens. The chapter also introduces essential productivity tools, including email clients, password managers, file encryption with PGP, and communication apps like Slack and Teams. For developers, it explains setting up Git, creating SSH keys, and using code editors like VS Code and PyCharm. By the end, readers have a fully personalized and productive Ubuntu environment.

Chapter 4: Setting up Firewall, VPN, and Wi-Fi Networks - This chapter covers essential networking tools in Ubuntu, focusing on connectivity and security. It explains how to configure DHCP and static IP addresses, connect to VLANs and wireless networks, and mask your device's MAC address for privacy. It introduces VPN setup using OpenVPN and WireGuard, and demonstrates encrypting DNS traffic with DNSS for added protection. Readers also learn to configure custom firewall rules using iptables and safeguard their systems with ClamAV antivirus. By the end, users can confidently connect to, secure, and manage networks on Ubuntu Desktop in both personal and professional environments.

Chapter 5: Preparing Virtualization Environment - This chapter introduces virtualization in Ubuntu using KVM, enabling users to run multiple isolated operating systems on a single host. It covers installing KVM, setting up network bridges, and creating **virtual machines** (**VMs**). Readers learn to manage VM settings, take snapshots, and enable device passthrough like GPUs. The chapter also explores alternative tools such as VirtualBox, VMware, and Vagrant for VM creation and sharing. Guidance is provided on converting VM image formats for compatibility across platforms. By the end, users can effectively create, configure, and manage VMs and virtual environments for both testing and development purposes.

Chapter 6: Up and Running with Kubernetes and Docker - This chapter introduces Docker and Kubernetes as essential tools for containerized development. It covers Docker installation, running a Minecraft server container, and managing multi-service applications like WordPress and Metabase using Docker Compose. Readers learn to connect multiple containers, persist data, and streamline development with local volumes. The chapter then transitions to Kubernetes using Minikube, demonstrating how to deploy WordPress and MySQL with Kubernetes manifests. Key concepts include namespaces, services, deployments, and load balancers. By the end, readers can run, scale, and manage containers locally using Docker and Kubernetes for efficient, isolated application development and testing.

Chapter 7: Install Ubuntu Server on Metal, Cloud, and Network - This chapter explores various methods to install and manage Ubuntu Server across physical machines, virtual environments, and cloud platforms. It covers using SSH keys for secure access, installing via USB or VM, and deploying on cloud providers like Google Cloud and Hetzner. Readers also learn about scalable server provisioning using **machine as a service** (**MAAS**), including PXE booting and automated installations. Concepts like cattle vs. pets emphasize treating servers as disposable for easier management. By chapter's end, users can confidently set up single or multiple Ubuntu Servers suited to local or enterprise-grade infrastructure..

Chapter 8: Keeping Check on Your Ubuntu Server - This chapter explores tools and techniques for monitoring and securing Ubuntu Servers. It begins with basic Linux commands like top, netstat, and du for local performance insights. For GUI monitoring, Cockpit is introduced, while Grafana, Prometheus, and Node Exporter enable the visualization of scalable server metrics. Filebeat, Elasticsearch, and Kibana are used for centralized log collection and analysis. Security tools such as Fail2Ban and OSSEC help detect intrusion attempts and automate responses. Readers learn to create dashboards, configure alerts, and integrate logs and metrics into unified views—building a robust foundation for proactive server management and security.

Chapter 9: Setup Advanced Network, Firewall and VPN Servers - This chapter guides configuring Ubuntu Server as a network firewall and router. It covers setting up network interfaces, VLANs, and routing using Netplan, and securing traffic with iptables. You learn to install and configure dnsmasq as both a DHCP and DNS server, enabling client IP management and domain resolution. The chapter also details creating secure VPN tunnels using OpenVPN and WireGuard, including certificate generation and client-server configurations. By the end, you'll have a fully functional Ubuntu-based router capable of secure communication and traffic control for both local and remote networks.

Chapter 10: Running Virtualization Server Environment - This chapter explores setting up an Ubuntu Server as a virtualization host using KVM. It guides installing and managing VMs via CLI, desktop GUI (Virtual Machine Manager), and a web interface (Cockpit). The chapter also introduces containerization using Podman, a Docker alternative, to deploy a monitoring stack with Grafana, Prometheus, Loki, and Promtail. Detailed setup for container-based metrics and log collection from multiple servers is provided, using podman-compose. The chapter concludes by contrasting virtualization and containers, emphasizing containers for lightweight, scalable service deployment and preparing readers for Kubernetes in the next chapter.

Chapter 11: Setup Webserver, Deploy and Run Webapps - This chapter explores setting up web servers and deploying web applications on Ubuntu. It begins with installing Apache and NGINX, configuring domains, and serving web content. Two databases are introduced: MariaDB (SQL) and MongoDB (NoSQL), including setup, usage, and backup processes. Practical deployments include WordPress and Observium, demonstrating PHP app hosting and virtual host configurations. Rocket.Chat is deployed via Docker, connected to MongoDB. The chapter also covers performance tuning for web servers and emphasizes using proper database users for security. By the end, readers can host, optimize, and manage web apps and databases on Ubuntu Servers.

Chapter 12: Kubernetes Run and Setup - This chapter guides readers through setting up a Kubernetes cluster on Ubuntu Servers using kubeadm. It covers preparing master and worker nodes, disabling swap, and installing core components. The chapter introduces Helm for managing Kubernetes packages and demonstrates deploying essential services like OpenEBS for storage, Prometheus and Grafana for monitoring, and Traefik with MetalLB for ingress and load balancing. A full WordPress deployment, including MySQL, is configured and accessed using both NodePort and ingress routes. The chapter concludes with basic kubectl commands for troubleshooting and managing the cluster, forming a strong foundation for further automation.

Chapter 13: Task Automations, CI/CD Pipeline, and Service Deployment - This chapter focuses on automating infrastructure tasks using Bash, Ansible, and Terraform to ensure repeatable, reliable server and service setups. It starts with creating reusable Bash scripts, then advances to Ansible for managing tasks across multiple servers via Docker containers. The chapter introduces building and pushing Docker images, running host-level tasks from containers, and deploying applications to Kubernetes using Terraform. These automation techniques form a complete CI/CD pipeline foundation. By the end, readers gain the tools to configure, deploy, and manage services efficiently, marking a transition to professional DevOps practices within Linux and Ubuntu environments.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/ou09t8w

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Ubuntu-System-Administration-Guide.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Getting Familiar with Ubuntu Ecosystem

Introduction

In this chapter, we will be introduced to Linux and Ubuntu. We will start by discussing the creation of Linux. Then, we will go into Ubuntu and look at how it began, understand Ubuntu releases, and cover some of the different versions of Ubuntu. Finally, we will cover other Linux distributions and how they are different from Ubuntu. Additionally, we will discuss what parts make up the Linux stack.

We encourage you to create a blog page and an account by the end of the chapter so that you are ready for the subsequent chapters in the book.

Structure

In this chapter, we will cover the following topics:

- Linux history
- Linux stack
- Usage and stats of Linux
- Ubuntu history
- Ubuntu releases
- Ubuntu versions

- Other Linux distributions
- Create GitHub and blogger account

Objectives

By the end of the chapter, we will understand how Linux started and how the different parts of the Linux stack are put together. We will also understand the difference between different Linux versions and how Ubuntu is versioned and released.

Linux history

Linux was created by *Linus Torvalds*, a computer science student at the *University of Helsinki*, in 1991 at the age of 21. What started as a small hobby is today one of the most used operating systems.

At first, *Linus* named the invention Freax, but one of the administrators at FUNET, where the project was uploaded, did not like that name and renamed it to Linux.

Tux, the penguin mascot as depicted in *Figure 1.1*, was chosen by *Linus* after a small penguin bit him during a visit to the *National Zoo and Aquarium* in *Canberra*.

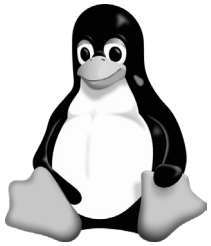

***Figure 1.1**: Tux*

Today, you can find the Linux kernel development on GitHub. **https://github.com/torvalds/linux** and follow the development there, and today, there are over 13k contributors to the Linux kernel. Refer to the following link: **https://en.wikipedia.org/wiki/Ubuntu**

Linux stack

The Linux kernel is the base of all Linux operations systems. It is the one that boots and adds all the drivers. If the car brand is named Ubuntu, the engine is the Linux kernel. On top of the Linux kernel, we can choose/build different operating systems, such as Ubuntu and Red Hat.

Now, to use your Linux system as a desktop computer, you would also need some type of windows manager. In this book, we will look more at the windows manager i3 and the default windows manager in Ubuntu GNOME.

Usage and stats of Linux

The usage of Linux and Ubuntu is growing fast. Today, Linux is the primary OS used by our supercomputers. It powers NASA servers and is the most used OS for IoT devices.

In the cloud, it is the dominant OS, with 90% of all OS running Linux. Linux is also the OS powering the Kubernetes cluster, which is becoming the default platform for hosting applications. Some other stats from Linux are as follows:

- Web servers also completely rely on Linux. According to Linux server statistics, of the top 1 million web servers, 96.3% employ Linux environments.
- The Linux kernel development report reveals that 90% of the workload deployed on the cloud is based on the Linux system.
- 65 SpaceX missions were completed using Linux-powered technology, according to the latest Linux statistics.
- According to the latest Linux distributions statistics, Ubuntu is the most popular Linux distributions (32.8%), followed by Debian (14.4%) and CentOS (10.8%).

Ubuntu history

One of the most used Linux operating systems is Debian, and Ubuntu is based on Debian. Ubuntu was built by the British company *Canonical* and is set to be a friendly and easy-to-use Linux system. The first release was Ubuntu 4.10, which was released in October 2004. Today, a lot has happened, and Ubuntu has now released version 22.04.

There is a base version of Ubuntu. Ubuntu Desktop are the desktop for laptops, office, and home computers with a GUI. Ubuntu Server runs on servers on both cloud and metal, and Ubuntu Core is used to run on IoT devices.

https://en.wikipedia.org/wiki/Ubuntu

Ubuntu releases

Ubuntu is released twice yearly, one version on 04 April and one on 10 October. The release makes up the name with the year. Ubuntu 22.10 will be released in October 2022. So, how long is then a release supported? Well, that depends. Ubuntu makes **long-term support** (**LTS**) releases, and the LTS releases are supported for ten years. If you plan to set up a server that will run longer or a desktop to work on, consider using the LTS releases. However, if you want the latest kernels and tools, you can look at the newest release. In this book, we will recommend installing the LTS releases. Ubuntu also provides tools to upgrade from the LTS release to the next one.

More on the Ubuntu releases is found here, **https://ubuntu.com/about/release-cycle**.

Ubuntu version

Ubuntu and the Linux kernel are open-source, which has led many developers to make their own version of Ubuntu. People who wanted to use another graphical desktop environment on a particular version of Ubuntu for schools or music studios.

Some of the different versions of Ubuntu are as follows:

Mint

Linux Mint is based on Ubuntu and the XFC Windows desktop. Mint Linux is pre-filled with many of the standard tools to get you started and working quickly after installation.

Linux Mint is an operating system for desktop and laptop computers. It is designed to work out of the box and comes fully equipped with the apps most people need.

You can find more information on Mint Linux and download links at **https://linuxmint.com/**.

Pop

System76 is a company building Linux computers, and its operating system is based on Ubuntu.

Pop!_OS is designed for fast navigation, easy workspace organization, and fluid, convenient workflow. Your operating system should encourage discovery, not obstruct it.

You can find more info on Pop Linux and download links at **https://pop.system76.com/**.

LXLE

Lxle is a light version of Ubuntu. We used computers that were low in resources but still needed to be able to be used in a secure and updated way.

The developers of LXE describe it as light on resources and heavy on functions. LXLE is based on Ubuntu, and it is super-fast to boot up.

You can find more info on LXLE and download links at **https://lxle.net/**.

Other Linux distributions

Ubuntu is based on the Linux distributionsro called **Debian**, but there are more Linux OSs than Ubuntu. Raspberry Pi OS is another famous OS based on Debian. However, there are many more. Kali Linux is a Linux OS specially built for hackers. It includes all the tools you would need as a hacker and runs easily from a USB stick.

But our Linux distributions does not stop with the Debian family. There are more families of Linux.

Red Hat has Red Hat Enterprise Linux, a stable Linux distributionsro that many companies use. It includes open-source software and Red Hat's own software. Fedora Linux will be the next Red Hat Linux. In Fedora, we see the latest software and kernel versions. If you start using Docker, you will get to know **Alpine Linux**. Alpine is a minimal Linux perfect for building small Docker images, **https://www.alpinelinux.org/.**

Regarding Kubernetes, new Linux distributionsributions release unique builds for running containers. The most famous are Flatcar and CoreOS, where Flatcar is a fork of CoreOS. AWS has also released its own Linux build for containers called **Bottlerocket**.

One new project is Talos Linux, a special Linux distributionsro only made to run Kubernetes. One special part of Talos Linux is that it does not have any external access, and all configs are used using API calls.

To read more and get installation instructions on the special Linux distributionsro, visit the following links in the chapter's reference section.

Let us look at the difference between Linux versions:

It is all about how you use your computer and server. On your laptop, run Ubuntu LTS, and run Ubuntu LTS on my home server. Then, you can reuse the tools and script, for example, to set up backup and access.

However, flatcar Linux is run in the Kubernetes cluster, which is good for running containers and Kubernetes.

We may have some servers that run Red Hat, and the difference is, for example, the package manager. In Ubuntu, as you will learn in this book, the package tool is called **`apt`**.

The command installs **`apache`** on a Ubuntu/Debian Linux:

```
1. apt-get install apache
```

However, on a Red Hat server, your package manager is named **`yum`**. So, to install the Apache web server, you would run.

```
1. yum install httpd
```

They both will install the Apache web server, but the way you type the command for installing Apache is different. The same command in Alphine Linux would be as follows:

```
1. apk add apache
```

There are some differences in this way, but as you will learn to use **`apt`** in this book, it is simple to move to a Red Hat-based Linux, and the base of how you install the package is the same. You only need to find the right command.

Create GitHub and blogger account

In this book, we will use code and configuration files when we set up our desktop and install services like a web service on our Ubuntu Server. We will build Docker apps to run