AlmaLinux 9 for Web Hosting Beginners

Install, Configure, and Secure Apache, Nginx, PHP, and MySQL on a VPS

Preface

Every website you visit, every application you rely on, every online store where you make a purchase—behind it all stands a server, quietly doing its job. And more often than not, that server runs Linux.

If you've ever wondered what it takes to set up your own web server from scratch, you're holding the right book.

Why This Book Exists

AlmaLinux 9 for Web Hosting Beginners was written for a specific reader: someone who wants to host websites on a real server, using a real enterprise-grade operating system, but doesn't know where to start. AlmaLinux 9—a free, open-source, community-driven distribution built for stability and long-term support—has rapidly become one of the most trusted platforms for web hosting and production server environments. It is the natural successor for those seeking a RHEL-compatible foundation without licensing costs, and it deserves a guide that meets beginners exactly where they are.

This book is that guide.

What You'll Learn

Across sixteen chapters and five practical appendices, you'll walk through the complete journey of turning a blank Virtual Private Server (VPS) into a fully functional,

secured web hosting environment running on AlmaLinux 9. The path is deliberate and cumulative. You'll begin by understanding what AlmaLinux is and why it stands out among Linux distributions for hosting workloads. From there, you'll provision and harden your VPS, master package management with `dnf`, and then dive into the heart of web hosting: installing and configuring **Apache** and **Nginx**, integrating **PHP**, and setting up **MySQL (MariaDB)** for database-driven applications.

But this book doesn't stop at installation. You'll deploy actual websites on both Apache and Nginx, secure your AlmaLinux server with firewalls and SSL certificates from **Let's Encrypt**, and learn the monitoring and maintenance practices that separate a hobbyist setup from a production-ready system. The final chapter bridges the gap between beginner hosting and professional server administration, giving you a clear view of where to go next.

How This Book Is Structured

The chapters are designed to be followed **in order**, each building on the knowledge established before it. However, if you already have an AlmaLinux 9 VPS running and hardened, you can jump directly to the web server chapters that interest you most. The appendices—including an AlmaLinux commands cheat sheet, configuration templates, a troubleshooting guide for common hosting errors, a hardening checklist, and a learning roadmap—are references you'll return to long after you've finished reading.

Who This Book Is For

You don't need prior Linux administration experience. You *do* need curiosity, a willingness to work in the terminal, and access to a VPS or virtual machine where you can practice. Whether you're a developer who wants control over your hosting environment, a student exploring system administration, or simply someone tired of shared hosting limitations, this book will give you the confidence and competence to manage your own AlmaLinux-based web server.

A Note of Gratitude

This book would not exist without the **AlmaLinux OS Foundation** and the vibrant community of contributors who have built and sustained a distribution that thousands of organizations now depend on. I'm also deeply grateful to the developers and maintainers of Apache, Nginx, PHP, MariaDB, and Let's Encrypt—the open-source projects that form the backbone of modern web hosting. Finally, thank you to every reader who chose this book as their starting point. Your willingness to learn is what keeps the open-source ecosystem thriving.

Let's Begin

There is something deeply satisfying about typing a domain name into a browser and watching a page load from a server *you* built, *you* configured, and *you* secured —running on AlmaLinux 9. That moment is closer than you think.

Let's get started.

Bas van den Berg

Table of Contents

Chapter 1: Understanding AlmaLinux 9 and VPS Hosting

When you first step into the world of web hosting, the sheer volume of choices can feel overwhelming. Which operating system should you run on your server? What kind of hosting environment best suits your needs? How do you balance cost, performance, and reliability? These are the questions that every aspiring web administrator must answer, and this chapter is designed to walk you through those decisions with clarity and confidence. By the end of this chapter, you will have a thorough understanding of what AlmaLinux 9 is, why it has become one of the most trusted server operating systems in the industry, and how Virtual Private Server hosting provides the ideal foundation for your web hosting journey.

The Story Behind AlmaLinux

To truly appreciate AlmaLinux, you need to understand the landscape from which it emerged. For nearly two decades, CentOS served as the backbone of countless web servers, enterprise environments, and data centers around the world. CentOS was a community-driven rebuild of Red Hat Enterprise Linux, which meant that administrators could enjoy the stability, security, and long-term support of an enterprise-grade operating system without the associated licensing costs. It was, in many ways, the perfect solution for budget-conscious organizations and individual developers who needed reliability without compromise.

Then, in December 2020, Red Hat announced a dramatic shift. CentOS Linux, as the community had known it, would be discontinued. The traditional CentOS release model, which provided a stable, downstream rebuild of RHEL, would be replaced by CentOS Stream, a rolling-release distribution that sits upstream of RHEL. This meant that CentOS Stream would receive updates before they were tested and included in RHEL, effectively transforming it from a stable production platform into a development preview. The announcement sent shockwaves through the Linux community, leaving thousands of organizations scrambling for an alternative.

CloudLinux, a company with deep expertise in Linux-based hosting solutions, responded swiftly. Within weeks of the CentOS announcement, CloudLinux founder Igor Seletskiy announced the creation of AlmaLinux OS, a free, open-source, community-driven distribution that would serve as a direct replacement for CentOS. The name "Alma" comes from the Latin word for "soul," reflecting the project's mission to carry forward the soul and spirit of what CentOS had always represented. The first stable release of AlmaLinux arrived in March 2021, and the distribution quickly gained traction among system administrators, hosting providers, and enterprises worldwide.

AlmaLinux 9, released in May 2022 and based on RHEL 9, represents the latest major version of this distribution. It brings with it a modernized kernel, improved security features, updated software packages, and enhanced performance characteristics that make it exceptionally well-suited for web hosting workloads. With a commitment to providing free, production-grade updates until 2032, AlmaLinux 9 offers the kind of long-term stability that web hosting environments demand.

What Makes AlmaLinux 9 Ideal for Web Hosting

Choosing an operating system for your web server is not a decision to be taken lightly. The operating system forms the foundation upon which every other component of your hosting stack will run. Apache, Nginx, PHP, MySQL, your firewall, your SSL certificates, your monitoring tools — all of these depend on a stable, secure, and well-maintained operating system. AlmaLinux 9 excels in every one of these areas, and understanding why requires a closer look at its key characteristics.

Binary Compatibility with Red Hat Enterprise Linux is perhaps the single most important feature of AlmaLinux. Because AlmaLinux is built from the same source code as RHEL, software packages compiled for RHEL will run on AlmaLinux without modification. This means that you have access to the vast ecosystem of enterprise software, documentation, and community knowledge that has been built around the RHEL family over the past two decades. When you search for a solution to a server configuration problem, guides written for RHEL or CentOS will almost always apply directly to your AlmaLinux system.

Security and Stability are paramount in any hosting environment. AlmaLinux 9 ships with SELinux (Security-Enhanced Linux) enabled by default, providing mandatory access control that goes far beyond traditional Unix file permissions. The distribution includes robust support for firewalld, the dynamic firewall management tool that makes it straightforward to control network access to your server. AlmaLinux also benefits from the security patches and updates that flow from the RHEL ecosystem, ensuring that vulnerabilities are addressed promptly and thoroughly.

The DNF Package Manager in AlmaLinux 9 provides a powerful and reliable mechanism for installing, updating, and managing software on your server. DNF, which stands for Dandified YUM, is the successor to the older YUM package man-

ager and offers improved performance, better dependency resolution, and a more intuitive command structure. Throughout this book, you will use DNF extensively to install and manage your web hosting stack.

Application Streams and Modularity represent another significant advantage. AlmaLinux 9 uses a modular repository system that allows you to choose from multiple versions of key software components. For example, you might need PHP 8.1 for one project and PHP 8.2 for another. The modular approach makes it possible to install and manage these different versions without conflicts, giving you flexibility that was much harder to achieve in older CentOS releases.

The following table summarizes the key features of AlmaLinux 9 that are most relevant to web hosting:

Feature	Description	Relevance to Web Hosting
RHEL Binary Compatibility	Built from RHEL 9 source code, ensuring identical behavior	Access to enterprise-grade software and extensive documentation
SELinux	Mandatory access control security framework enabled by default	Provides an additional layer of protection for web applications and data
DNF Package Manager	Modern package management with improved dependency resolution	Simplifies installation and maintenance of Apache, Nginx, PHP, MySQL
Firewalld	Dynamic firewall management with zone-based configuration	Easy-to-manage network security for HTTP, HTTPS, SSH, and database ports
Systemd	Modern init system for service management	Reliable starting, stopping, and monitoring of web server processes
Application Streams	Modular software delivery allowing multiple versions	Flexibility to run different PHP or database versions as needed

Cockpit Web Console	Browser-based server management interface	Visual monitoring and management for administrators who prefer a GUI
Long-Term Support	Updates and security patches guaranteed until 2032	Production stability without the need for frequent major upgrades
Kernel 5.14	Modern Linux kernel with performance improvements	Better hardware support, improved networking, and enhanced security
OpenSSL 3.0	Updated cryptographic library	Modern TLS support essential for HTTPS and secure communications

Understanding VPS Hosting

Now that you understand what AlmaLinux 9 brings to the table, it is equally important to understand the hosting environment where you will deploy it. Virtual Private Server hosting, commonly abbreviated as VPS, represents a middle ground between shared hosting and dedicated server hosting. To appreciate why VPS hosting is the recommended environment for this book, let us examine each hosting type and understand where VPS fits in the spectrum.

Shared Hosting is the most basic and affordable type of web hosting. In a shared hosting environment, your website shares a single physical server with dozens or even hundreds of other websites. The server's resources — its CPU, memory, disk space, and bandwidth — are divided among all the tenants. While shared hosting is inexpensive and requires minimal technical knowledge, it comes with significant limitations. You typically have no root access to the server, meaning you cannot install custom software, modify system configurations, or implement security measures beyond what the hosting provider allows. Performance can be un-

predictable because a spike in traffic on another tenant's website can affect your site's responsiveness. For learning purposes and for any serious web hosting endeavor, shared hosting is simply too restrictive.

Dedicated Server Hosting sits at the opposite end of the spectrum. With a dedicated server, you lease an entire physical machine for your exclusive use. You have complete control over every aspect of the hardware and software, from the operating system to the BIOS settings. Dedicated servers offer maximum performance and flexibility, but they come with a correspondingly high price tag. Monthly costs for a dedicated server can range from one hundred to several hundred dollars or more, making them impractical for beginners and small projects. Additionally, you are responsible for all hardware-related concerns, including dealing with hardware failures and capacity planning.

VPS Hosting provides the best of both worlds. A VPS is created by using virtualization technology to partition a single physical server into multiple isolated virtual machines. Each VPS operates as an independent server with its own dedicated allocation of CPU, memory, and storage. You receive full root access, which means you can install AlmaLinux 9, configure it exactly as you need, and run any software you choose. Because the virtualization layer ensures isolation between virtual machines, the activities of other VPS tenants on the same physical host will not affect your server's performance or security.

The following table provides a clear comparison of these hosting types:

Characteristic	Shared Hosting	VPS Hosting	Dedicated Server
Cost	Very low, typically under ten dollars per month	Moderate, typically five to fifty dollars per month	High, typically one hundred dollars or more per month
Root Access	No	Yes	Yes
Resource Allocation	Shared among all tenants	Dedicated allocation per VPS	Entire physical server

Performance	Variable and unpredictable	Consistent and reliable	Maximum available
Scalability	Very limited	Easy to scale up resources	Requires hardware changes
Technical Knowledge Required	Minimal	Moderate	Advanced
Operating System Choice	Determined by provider	Full choice, including AlmaLinux 9	Full choice
Security Isolation	Minimal	Strong virtualization-based isolation	Complete physical isolation
Customization	Very limited	Full customization	Full customization
Suitability for Learning	Poor	Excellent	Overkill for beginners

For the purposes of this book, VPS hosting is the ideal environment. It gives you the root access and flexibility needed to install and configure AlmaLinux 9 with a complete web hosting stack, while keeping costs manageable and providing a realistic production-like environment for learning.

Choosing a VPS Provider for AlmaLinux 9

Selecting a VPS provider is an important practical decision that will affect your experience throughout this book. While there are many reputable VPS providers available, there are specific criteria you should evaluate when making your choice.

AlmaLinux 9 Availability should be your first consideration. Not all VPS providers offer AlmaLinux 9 as a pre-installed operating system option. Before committing to a provider, verify that AlmaLinux 9 is available in their list of supported operating systems. Most major providers, including DigitalOcean, Linode (now

Akamai), Vultr, Hetzner, and Amazon Lightsail, offer AlmaLinux 9 as a deployment option. Some providers may list it under "AlmaLinux" without specifying the version, so check the release version to ensure you are getting AlmaLinux 9 specifically.

Minimum Resource Requirements for running a basic web hosting stack on AlmaLinux 9 are surprisingly modest. For the exercises in this book, a VPS with the following specifications will be sufficient:

Resource	Minimum Recommended	Comfortable Recommendation
CPU Cores	1 vCPU	2 vCPUs
RAM	1 GB	2 GB
Storage	20 GB SSD	40 GB SSD
Bandwidth	1 TB per month	2 TB per month
Network	1 Gbps port	1 Gbps port

A VPS meeting the minimum specifications typically costs between five and ten dollars per month from most providers, making it an affordable investment in your education.

Data Center Location matters for both performance and compliance reasons. Choose a data center location that is geographically close to your target audience. If you are primarily serving visitors in North America, select a data center in the United States or Canada. If your audience is in Europe, choose a European data center. The physical distance between your server and your visitors directly affects latency, which in turn affects how quickly your web pages load.

Network Quality and Uptime Guarantees vary between providers. Look for providers that offer a Service Level Agreement with at least 99.9 percent uptime. Check reviews and community forums for reports about network reliability and the quality of the provider's technical support.

Preparing for Your AlmaLinux 9 Journey

Before you proceed to the next chapter, where you will deploy your first AlmaLinux 9 VPS and connect to it for the first time, take a moment to prepare your local workstation. You will need a few tools installed on your personal computer to interact with your VPS.

An SSH Client is essential. SSH, which stands for Secure Shell, is the primary method you will use to connect to and manage your AlmaLinux 9 server. If you are using macOS or Linux on your personal computer, an SSH client is already built into the terminal application. If you are using Windows 10 or later, the built-in Windows Terminal or PowerShell includes an SSH client. For older versions of Windows, you may want to install PuTTY, a free and widely-used SSH client.

To verify that SSH is available on your system, open a terminal or command prompt and type:

```
ssh -V
```

This command will display the version of the SSH client installed on your system. If you see output similar to "OpenSSH_8.9p1" or any version number, you are ready to proceed.

A Text Editor on your local machine will be useful for editing configuration files before uploading them to your server, or for taking notes as you work through the exercises. Any text editor will do, but editors designed for code, such as Visual Studio Code, Sublime Text, or Notepad++, offer syntax highlighting that makes it easier to read and edit configuration files.

A Web Browser is something you almost certainly already have, but it is worth mentioning because you will use it frequently to test your web server configurations. Having a modern browser like Firefox or Chrome is recommended, as their

developer tools provide valuable information about HTTP headers, SSL certificates, and page loading performance.

Note: Throughout this book, all commands are designed to be executed on an AlmaLinux 9 system. When a command should be run on your local machine rather than on the server, this will be explicitly stated. Pay close attention to this distinction, as running server commands on your local machine or vice versa can lead to confusion and errors.

The Architecture of a Web Hosting Stack on AlmaLinux 9

Before diving into hands-on configuration, it is valuable to understand the overall architecture of the web hosting stack you will be building throughout this book. Each component plays a specific role, and understanding how they interact will make troubleshooting much easier when something does not work as expected.

At the base layer sits **AlmaLinux 9 itself**, providing the operating system kernel, file system, networking stack, and security framework. Everything else runs on top of this foundation.

The next layer consists of your **web server software**. In this book, you will learn to install and configure both Apache and Nginx. Apache, formally known as the Apache HTTP Server, is the most widely deployed web server in history and remains an excellent choice for hosting PHP-based applications. Nginx, pronounced "engine-x," is a high-performance web server that excels at serving static content and acting as a reverse proxy. You will learn when to use each one and how to configure them for optimal performance on AlmaLinux 9.

Above the web server sits your **application runtime**, which in this book is PHP. PHP is the scripting language that powers the majority of content management

systems, including WordPress, Drupal, and Joomla. AlmaLinux 9 provides PHP through its application stream modules, allowing you to install the specific version your applications require.

Your **database server**, MySQL, provides persistent data storage for your web applications. MySQL stores everything from user accounts and blog posts to product catalogs and session data. On AlmaLinux 9, you will install and secure MySQL, create databases and users, and configure appropriate permissions.

Wrapping around all of these components is your **security layer**, which includes firewalld for network-level access control, SELinux for mandatory access control, SSL/TLS certificates for encrypted communications, and various hardening measures that protect your server from common attacks.

The following table illustrates this layered architecture:

Layer	**Component**	**Role on AlmaLinux 9**
Security	Firewalld, SELinux, SSL/TLS, Fail2Ban	Protects the server and its services from unauthorized access and attacks
Database	MySQL	Stores and manages application data
Application Runtime	PHP	Executes dynamic web application code
Web Server	Apache or Nginx	Receives HTTP requests and serves web content to visitors
Operating System	AlmaLinux 9	Provides the foundation: kernel, networking, file system, and system services
Infrastructure	VPS (Virtual Private Server)	Provides the virtualized hardware resources

Each subsequent chapter in this book will focus on one or more of these layers, building your web hosting stack from the ground up in a logical, methodical progression.

What You Will Accomplish

By working through this book from beginning to end, you will transform a bare AlmaLinux 9 VPS into a fully functional, secure web hosting server. You will understand not just how to run commands, but why each configuration choice matters. You will develop the confidence to manage a production web server, troubleshoot common issues, and implement security best practices that protect both your server and the websites it hosts.

The journey begins in the next chapter, where you will provision your AlmaLinux 9 VPS, connect to it via SSH for the first time, and perform the initial system configuration that prepares your server for the web hosting stack you will build in the chapters that follow. Every great server starts with a solid foundation, and that foundation is AlmaLinux 9.

Chapter 2: Preparing Your VPS

When you first receive access to a Virtual Private Server, it can feel like being handed the keys to an empty building. The structure is there, the walls are standing, and the foundation is solid, but everything inside needs to be arranged, secured, and optimized before it becomes a functional workspace. This chapter walks you through every essential step of preparing your AlmaLinux 9 VPS for web hosting. By the time you finish reading and practicing these exercises, your server will be updated, secured with a firewall, configured with a non-root user, and ready for the software installations that follow in later chapters.

Choosing a VPS Provider and Installing AlmaLinux 9

Before you can configure anything, you need a server running AlmaLinux 9. Several well-known VPS providers offer AlmaLinux 9 as a pre-built operating system image that you can deploy with just a few clicks. The most popular providers include DigitalOcean, Linode (now Akamai), Vultr, Hetzner, and Contabo. Each of these providers maintains AlmaLinux 9 in their image library, which means you do not need to perform a manual installation from an ISO file.

When selecting a VPS plan for web hosting, your choice depends on the expected traffic and the complexity of the websites you plan to host. For learning purposes and small websites, a plan with 1 CPU core, 1 GB of RAM, and 25 GB of

SSD storage is sufficient. For production environments hosting multiple websites with moderate traffic, consider at least 2 CPU cores, 2 GB of RAM, and 50 GB of SSD storage.

The following table summarizes recommended minimum specifications for different use cases:

Use Case	CPU Cores	RAM	Storage	Monthly Cost (Approximate)
Learning and Testing	1	1 GB	25 GB SSD	$5 to $6
Single Small Website	1	2 GB	50 GB SSD	$10 to $12
Multiple Small Websites	2	4 GB	80 GB SSD	$20 to $24
Medium Traffic Production	4	8 GB	160 GB SSD	$40 to $48

Once you have selected your provider and plan, choose AlmaLinux 9 from the available operating system images. Most providers label it as "AlmaLinux 9" or "AlmaLinux 9.x" in their image selection dropdown. After deployment, the provider will give you an IP address and root password, or you may have the option to add an SSH key during the creation process. Write down your server's IP address because you will use it extensively throughout this book.

Note: If your provider only offers AlmaLinux 8, do not select it. The configurations and commands in this book are specifically written for AlmaLinux 9, which uses different package versions and slightly different default configurations. If AlmaLinux 9 is not available, consider uploading a custom ISO or choosing a different provider.

Connecting to Your VPS via SSH

Secure Shell, commonly known as SSH, is the standard method for remotely accessing and managing Linux servers. Every interaction you have with your AlmaLinux 9 VPS throughout this book will happen through an SSH connection. Understanding how to establish and manage this connection is a fundamental skill.

If you are using macOS or Linux on your local computer, the SSH client is already built into your terminal application. If you are using Windows, you can use the built-in Windows Terminal, PowerShell, or a dedicated application like PuTTY. Windows 10 and later versions include an OpenSSH client by default.

To connect to your server, open your terminal and type the following command, replacing the IP address with your actual server IP:

```
ssh root@203.0.113.50
```

The first time you connect, your SSH client will display a fingerprint verification message asking whether you trust this host. Type "yes" and press Enter. Then enter the root password provided by your VPS provider. Once authenticated, you will see the AlmaLinux 9 command prompt, which typically looks like this:

```
[root@localhost ~]#
```

This prompt tells you several things. The word "root" indicates you are logged in as the root user, which has unrestricted access to every file and command on the system. The "localhost" portion shows the hostname of the server, which you will change shortly. The tilde symbol represents your current directory, which is the root user's home directory at `/root`. The hash symbol at the end confirms you have root-level privileges.

Note: Working as the root user is dangerous for daily operations because any mistake, such as accidentally deleting critical system files, is executed without any

safety net. Later in this chapter, you will create a regular user account with sudo privileges, which provides a safer way to perform administrative tasks.

Updating AlmaLinux 9 Packages

The very first action you should take on any freshly deployed server is updating all installed packages to their latest versions. VPS providers create their operating system images at a specific point in time, and by the time you deploy one, there may be weeks or even months of security patches and bug fixes waiting to be applied.

AlmaLinux 9 uses the DNF package manager, which is the successor to YUM. DNF handles downloading, installing, updating, and removing software packages from AlmaLinux repositories. To update all packages on your system, run the following command:

```
dnf update -y
```

The `dnf` command invokes the package manager. The `update` argument tells it to check all installed packages against the repository and download newer versions where available. The `-y` flag automatically answers "yes" to any confirmation prompts, which is convenient when you know you want to proceed with all updates.

This process may take several minutes depending on how many packages need updating. You will see output showing each package being downloaded and installed. When it finishes, you may see a message suggesting a reboot, especially if the Linux kernel was updated. To reboot your server, run:

```
reboot
```

After rebooting, wait approximately 30 to 60 seconds, then reconnect via SSH using the same command you used before. Verify that the update was successful by checking your AlmaLinux version:

```
cat /etc/almalinux-release
```

This command displays the current AlmaLinux release information. You should see output similar to:

```
AlmaLinux release 9.4 (Seafoam Ocelot)
```

You can also verify the kernel version to confirm the latest kernel is running:

```
uname -r
```

This outputs the currently running kernel version, such as `5.14.0-362.24.1.el9_3.x86_64`. The version numbers confirm you are running an AlmaLinux 9 kernel.

The following table explains the key DNF commands you will use frequently throughout this book:

Command	Description
`dnf update -y`	Updates all installed packages to the latest available versions
`dnf install package_name -y`	Installs a new package from the repository
`dnf remove package_name -y`	Removes an installed package from the system
`dnf search keyword`	Searches the repository for packages matching the keyword
`dnf info package_name`	Displays detailed information about a specific package
`dnf list installed`	Lists all currently installed packages on the system

`dnf clean all`	Clears all cached package data to free disk space
`dnf history`	Shows the history of all DNF transactions performed

Setting the Hostname

Every server should have a meaningful hostname that identifies it on the network. The default hostname on a fresh VPS is often "localhost" or a random string assigned by the provider. Setting a descriptive hostname helps you identify which server you are connected to, especially if you manage multiple servers.

To set your hostname, use the `hostnamectl` command:

```
hostnamectl set-hostname webserver01.example.com
```

Replace `webserver01.example.com` with your actual desired hostname. If you own a domain, using a fully qualified domain name is ideal. If you are just learning, a simple name like `webserver01` works fine.

Verify the change by running:

```
hostnamectl
```

This displays comprehensive information about your system, including the static hostname, operating system, kernel version, and architecture. You should see your new hostname reflected in the output.

To make the new hostname appear in your current terminal session without logging out and back in, run:

```
exec bash
```

Your prompt should now display the new hostname instead of "localhost."

Creating a Non-Root User with Sudo Privileges

Operating your server as the root user on a daily basis is one of the most common mistakes beginners make. The root account has absolute power over the system, which means a single mistyped command can destroy your entire server configuration. The professional approach is to create a regular user account that can temporarily elevate its privileges using the `sudo` command when administrative tasks are needed.

To create a new user, run the following command, replacing "webadmin" with your preferred username:

```
adduser webadmin
```

Set a strong password for this new user:

```
passwd webadmin
```

You will be prompted to enter and confirm the password. Choose a password that is at least 12 characters long and includes a mix of uppercase letters, lowercase letters, numbers, and special characters.

Now grant this user sudo privileges by adding them to the "wheel" group. In AlmaLinux 9, the wheel group is preconfigured to have sudo access:

```
usermod -aG wheel webadmin
```

The `usermod` command modifies a user account. The `-aG` flags mean "append to group," and `wheel` is the group that grants sudo privileges on AlmaLinux and other Red Hat-based distributions.

Verify that the user was added to the wheel group:

```
groups webadmin
```

The output should show:

```
webadmin : webadmin wheel
```

Now test the new account by switching to it:

```
su - webadmin
```

The prompt changes to show the new username and a dollar sign instead of a hash, indicating regular user privileges:

```
[webadmin@webserver01 ~]$
```

Test that sudo works by running:

```
sudo dnf check-update
```

You will be asked to enter the webadmin user's password. After entering it, the command should execute successfully. If you see package update information or a message saying no updates are available, sudo is working correctly.

Type `exit` to return to the root session for now, as there are more configuration tasks to complete.

Note: From this point forward in the book, when you see commands prefixed with `sudo`, it means you should be logged in as your non-root user. Commands without `sudo` that require root access assume you have either switched to root with `su -` or are performing the initial setup as root.

Configuring SSH Key Authentication

Password authentication, while functional, is vulnerable to brute-force attacks where automated scripts try thousands of password combinations per minute. SSH key authentication is significantly more secure because it relies on a cryptographic