AlmaLinux Hosting Mastery

A Practical Step-by-Step Guide to Deploying, Securing, and Managing Web Servers on AlmaLinux 9

Preface

When Red Hat announced the end of CentOS Linux as we knew it, the hosting community faced an uncomfortable question: *What do we build on now?* AlmaLinux emerged as a clear, community-driven answer—a 1:1 binary-compatible fork of RHEL that restored the stability, predictability, and enterprise-grade foundation that millions of servers depended on. But choosing AlmaLinux is only the first step. Knowing how to deploy, secure, and manage a production hosting environment on it—that is the journey this book was written to guide you through.

AlmaLinux Hosting Mastery is a practical, step-by-step handbook for anyone who wants to run web hosting infrastructure on AlmaLinux 9 with confidence. Whether you are a system administrator migrating from CentOS, a developer standing up your first production server, or an experienced engineer looking for a structured reference, this book meets you where you are and walks you through every layer of a modern hosting stack—from initial server preparation to long-term scaling strategy.

What This Book Covers

The book is organized into **sixteen chapters** and **five appendices**, each designed to be both a sequential learning path and a standalone reference.

We begin by examining **why AlmaLinux 9 is an ideal platform for hosting**, exploring its heritage, release cycle, and ecosystem. From there, we move into the practical work of **preparing and hardening a production server**—covering system

updates, user management, SSH security, and firewall configuration, all tailored to AlmaLinux's tooling and defaults.

The heart of the book focuses on the **core hosting stack**: installing and configuring both Apache and NGINX, deploying PHP, managing MariaDB databases, and hosting multiple websites on a single AlmaLinux server. You will learn not just *how* to install these components, but how to configure them for security, performance, and maintainability in a real-world AlmaLinux environment.

We then turn to the critical concerns that separate a functional server from a *production-ready* one: **domain and DNS configuration**, **SSL certificates with Let's Encrypt**, **file permissions and storage architecture**, and **comprehensive server hardening**. The final chapters address the operational disciplines that keep hosting infrastructure healthy over time—**monitoring, performance optimization, backup and disaster recovery**, and **scaling strategies** for when your AlmaLinux hosting environment needs to grow.

The appendices provide ready-to-use resources: a **command cheat sheet** specific to AlmaLinux 9, **configuration templates** for Apache and NGINX, a **security checklist**, **backup automation scripts**, and a **troubleshooting guide** for the most common hosting issues you will encounter.

How to Use This Book

Every chapter includes concrete commands, configuration examples, and explanations grounded in AlmaLinux 9's specific package management, SELinux policies, and system architecture. You can read the book cover to cover to build a hosting server from scratch, or jump to any chapter when you need targeted guidance on a specific topic.

Who This Book Is For

This book is for **system administrators, DevOps practitioners, web developers, and IT students** who want hands-on, no-nonsense guidance for running hosting infrastructure on AlmaLinux. A basic familiarity with Linux command-line operations is helpful, but no prior experience with AlmaLinux specifically is required.

Acknowledgments

This book would not exist without the **AlmaLinux OS Foundation** and the vibrant community of contributors who have made AlmaLinux a trusted platform for production workloads worldwide. I am also grateful to the developers and maintainers of Apache, NGINX, MariaDB, Let's Encrypt, and the countless open-source tools that make modern web hosting possible. Finally, my sincere thanks to the readers, reviewers, and colleagues whose feedback shaped this book into something genuinely useful.

Building a reliable hosting server is not about following trends—it is about making deliberate, informed choices at every layer of the stack. AlmaLinux 9 gives you a rock-solid foundation. This book gives you the blueprint. Let's build something that lasts.

Miles Everhart

Table of Contents

Chapter 1: Why AlmaLinux 9 for Hosting

When the world of enterprise Linux experienced one of its most significant disruptions in December 2020, the community found itself at a crossroads. Red Hat announced that CentOS Linux, the beloved free rebuild of Red Hat Enterprise Linux that had served millions of servers worldwide for nearly two decades, would shift its focus to CentOS Stream, a rolling-release distribution that sits upstream of RHEL rather than downstream. For countless system administrators, hosting providers, and organizations that had built their entire infrastructure on the stability and predictability of CentOS, this announcement sent shockwaves through the industry. Servers needed a new home, and the community needed a new champion. That champion emerged in the form of AlmaLinux.

AlmaLinux was born out of necessity, but it quickly grew into something far more significant than a simple replacement. Created by CloudLinux Inc., a company with over a decade of experience building and maintaining Linux distributions specifically designed for the hosting industry, AlmaLinux arrived with a pedigree that few other alternatives could match. CloudLinux had already been producing a commercially supported operating system used by thousands of hosting providers around the globe, which meant the team behind AlmaLinux understood the precise demands of production hosting environments. They understood uptime. They understood security. They understood what it means when a server goes down and thousands of websites disappear from the internet.

The first stable release of AlmaLinux arrived in March 2021, and it was a one-to-one binary compatible rebuild of Red Hat Enterprise Linux. This compatibility was

not merely a marketing claim; it meant that software compiled for RHEL would run on AlmaLinux without modification, that system configurations could be migrated with minimal effort, and that the vast ecosystem of tools, documentation, and expertise built around the RHEL family of distributions remained entirely relevant. For hosting professionals who had spent years mastering CentOS, the transition to AlmaLinux felt less like learning a new system and more like moving into a new house with the same familiar floor plan.

AlmaLinux 9, the version that serves as the foundation for everything discussed in this book, represents the maturation of this distribution. Based on RHEL 9, it brings with it a modernized kernel, updated system libraries, improved security features, and a forward-looking architecture that will receive updates and security patches through 2032. This extended support lifecycle is not a trivial detail for hosting environments. When you deploy a web server, you are not building something disposable. You are creating infrastructure that needs to remain stable, secure, and performant for years. The ten-year support window of AlmaLinux 9 gives hosting professionals the confidence to deploy today knowing that their foundation will remain solid well into the next decade.

Understanding why AlmaLinux 9 stands as an exceptional choice for hosting requires examining the specific characteristics that matter most in server environments. Stability sits at the very top of this list. In the hosting world, stability is not simply a desirable feature; it is the fundamental requirement upon which everything else is built. A hosting server might run hundreds of websites, handle thousands of email accounts, process millions of database queries, and serve content to visitors from every corner of the globe. Any instability in the underlying operating system cascades upward through every layer of the stack, potentially affecting every single customer and every single service. AlmaLinux 9 inherits the rigorous testing and conservative package management philosophy of RHEL, which means that packages are thoroughly vetted before inclusion, updates are carefully tested

for regressions, and the system behaves predictably day after day, month after month, year after year.

Security represents another critical pillar of the AlmaLinux hosting proposition. AlmaLinux 9 ships with SELinux enabled by default in enforcing mode, providing mandatory access controls that go far beyond traditional Unix permissions. For a hosting server exposed to the public internet, SELinux acts as an additional layer of defense that can contain the damage even when an application vulnerability is exploited. The distribution also includes modern cryptographic defaults, with system-wide cryptographic policies that make it straightforward to enforce minimum security standards across all services. OpenSSL, GnuTLS, and other cryptographic libraries are configured to reject weak algorithms and protocols by default, which means that a freshly installed AlmaLinux 9 server already meets many compliance requirements without additional hardening.

The following table provides a comprehensive comparison of AlmaLinux 9 against other distributions commonly considered for hosting environments:

Feature	**AlmaLinux 9**	**Ubuntu Server 22.04**	**Debian 12**	**Rocky Linux 9**	**CentOS Stream 9**
Base	RHEL 9	Independent	Independent	RHEL 9	Upstream RHEL
Support Lifecycle	2032	2027 (standard)	2028	2032	Rolling
Default Init System	systemd	systemd	systemd	systemd	systemd
Package Manager	DNF	APT	APT	DNF	DNF
SELinux Default	Enforcing	AppArmor	None (optional)	Enforcing	Enforcing

Kernel Version	5.14	5.15	6.1	5.14	5.14+
Binary RHEL Compatible	Yes	No	No	Yes	Partial
cPanel/WHM Support	Full	No	No	Full	No
Plesk Support	Full	Full	Full	Full	Limited
CloudLinux Compatible	Full	No	No	Full	No
Community Governance	AlmaLinux OS Foundation	Canonical	Debian Project	Rocky Enterprise Software Foundation	Red Hat
Free to Use	Yes	Yes	Yes	Yes	Yes
Commercial Support Available	Yes (via partners)	Yes (Canonical)	Limited	Yes (via partners)	Yes (Red Hat)

This comparison reveals several important details. Notice that AlmaLinux 9 and Rocky Linux 9 share many characteristics because they are both RHEL 9 rebuilds. However, AlmaLinux distinguishes itself through its governance model and the backing of CloudLinux Inc., which brings specific hosting industry expertise. The AlmaLinux OS Foundation operates as a 501(c)(6) nonprofit organization, ensuring that the distribution's direction is guided by community interests rather than a single corporate entity's commercial strategy. At the same time, the practical involvement of CloudLinux means that hosting-specific concerns receive attention and priority that a purely community-driven project might not provide.

The compatibility with major hosting control panels deserves particular emphasis. In the hosting industry, control panels such as cPanel/WHM, Plesk, and DirectAdmin are not optional luxuries; they are essential tools that enable hosting

providers to manage hundreds or thousands of accounts efficiently. These control panels have deep integration requirements with the underlying operating system, and their developers must certify each distribution version before it can be officially supported. AlmaLinux 9 has achieved full certification from all major control panel vendors, which means that hosting providers can deploy it with confidence knowing that their management tools will function correctly and receive ongoing support.

Let us examine the practical aspects of verifying an AlmaLinux 9 installation, which serves as your first hands-on exercise in this book. After installing AlmaLinux 9 on your server, whether on bare metal hardware, a virtual machine, or a cloud instance, you should verify the system details to confirm that everything is in order.

To check the operating system version and release information, execute the following command:

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

This command reads the release file specific to AlmaLinux and will output something similar to:

```
AlmaLinux release 9.3 (Shamrock Pampas Cat)
```

Each AlmaLinux release carries a codename inspired by cat species, a nod to the distribution's name which derives from the Latin word "alma" meaning "soul" or "nourishing." For more detailed information about the system, you can use the following command:

```
cat /etc/os-release
```

The output provides structured information that scripts and tools can parse:

```
NAME="AlmaLinux"
VERSION="9.3 (Shamrock Pampas Cat)"
ID="almalinux"
```

```
ID_LIKE="rhel centos fedora"
VERSION_ID="9.3"
PLATFORM_ID="platform:el9"
PRETTY_NAME="AlmaLinux 9.3 (Shamrock Pampas Cat)"
ANSI_COLOR="0;34"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:almalinux:almalinux:9::baseos"
HOME_URL="https://almalinux.org/"
DOCUMENTATION_URL="https://wiki.almalinux.org/"
BUG_REPORT_URL="https://bugs.almalinux.org/"

ALMALINUX_MANTISBT_PROJECT="AlmaLinux-9"
ALMALINUX_MANTISBT_PROJECT_VERSION="9.3"
REDHAT_SUPPORT_PRODUCT="AlmaLinux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.3"
```

Note the `ID_LIKE` field, which lists "rhel centos fedora." This field tells software and scripts that AlmaLinux behaves like these distributions, ensuring broad compatibility with tools that check for distribution family membership.

To examine the kernel version running on your AlmaLinux 9 system, use:

```
uname -r
```

A typical output would be:

```
5.14.0-362.8.1.el9_3.x86_64
```

The kernel version string contains important information. The "5.14.0" portion indicates the upstream kernel version. The numbers following the dash represent the Red Hat patch level and build number. The "el9_3" portion confirms this is an Enterprise Linux 9.3 kernel. The "x86_64" suffix indicates the processor architecture.

For hosting environments, understanding the system's hardware resources is equally important. The following commands provide essential information:

```
# Display CPU information
lscpu
```

```
# Display memory information
free -h

# Display disk information
lsblk

# Display network interfaces
ip addr show
```

Each of these commands serves a specific purpose in evaluating your server's readiness for hosting duties. The `lscpu` command reveals the processor model, core count, and architecture details that determine how many concurrent requests your server can handle. The `free -h` command shows total, used, and available memory in human-readable format, which is critical for planning how many websites or applications your server can accommodate. The `lsblk` command lists all block devices and their partitions, helping you understand your storage layout. The `ip addr show` command displays all network interfaces and their assigned IP addresses, which you will need when configuring web servers and DNS records.

To verify that SELinux is running in enforcing mode, which is the recommended configuration for hosting servers, execute:

```
getenforce
```

The expected output is:

```
Enforcing
```

If the output shows "Permissive" or "Disabled," you should enable enforcing mode. The following table explains the three SELinux modes and their implications for hosting:

SELinux Mode	Behavior	Hosting Recommendation
Enforcing	SELinux policy is enforced. Access violations are blocked and logged.	Recommended for all production hosting servers. Provides maximum security.
Permissive	SELinux policy is not enforced but violations are logged.	Useful only for debugging SELinux issues. Should not be used in production.
Disabled	SELinux is completely turned off. No policy is loaded or enforced.	Never recommended for hosting servers. Removes an important security layer.

To permanently set SELinux to enforcing mode, edit the configuration file:

```
sudo vi /etc/selinux/config
```

Ensure the following line is present:

```
SELINUX=enforcing
```

A system reboot is required for changes to the SELinux mode to take full effect when transitioning from disabled to enforcing, because SELinux needs to relabel the entire filesystem.

The package management system in AlmaLinux 9 uses DNF (Dandified YUM), which is the evolution of the YUM package manager that CentOS administrators will remember fondly. DNF provides faster dependency resolution, better memory management, and a more consistent command interface. To verify that your system is up to date, which is the first action you should take on any new hosting server, run:

```
sudo dnf update -y
```

This command contacts the AlmaLinux repositories, downloads the latest package versions, and installs them. The `-y` flag automatically answers "yes" to confirmation

prompts, which is convenient but should be used with awareness that it will apply all available updates without manual review.

To list the configured repositories on your system:

```
dnf repolist
```

A default AlmaLinux 9 installation will show repositories similar to:

```
repo id                    repo name
appstream                  AlmaLinux 9 - AppStream
baseos                     AlmaLinux 9 - BaseOS
extras                     AlmaLinux 9 - Extras
```

The BaseOS repository contains the core operating system packages that provide the foundation of the system. The AppStream repository contains user-space applications, runtime languages, databases, and web servers in multiple versions through the use of modularity. The Extras repository contains additional packages that supplement the core distribution. This repository structure mirrors RHEL exactly, which means that documentation and guides written for RHEL 9 package management apply directly to AlmaLinux 9.

The performance characteristics of AlmaLinux 9 also merit discussion. Because AlmaLinux is a binary-compatible rebuild of RHEL, it inherits the same performance optimizations that Red Hat engineers have implemented. The kernel includes tuning for modern hardware, support for large memory configurations, and optimizations for network throughput that are essential in hosting environments. The default filesystem is XFS, which excels at handling large files and high-throughput workloads common in web hosting. XFS supports online resizing, efficient space allocation, and scales well on systems with many CPU cores, all characteristics that align perfectly with the demands of a busy hosting server.

AlmaLinux 9 also introduces improvements in container support through Podman, which replaces Docker as the default container runtime. While container or-

chestration may seem tangential to traditional web hosting, the modern hosting landscape increasingly involves containerized applications, microservices, and hybrid deployments. Having robust container support built into the operating system means that AlmaLinux 9 is prepared for both traditional hosting workloads and modern application deployment patterns.

The firewall management in AlmaLinux 9 uses firewalld, a dynamic firewall manager that provides a higher-level abstraction over iptables and nftables. For hosting servers that need to manage access to web services, email servers, database servers, and administrative interfaces, firewalld offers zone-based configuration that simplifies complex firewall rules. You can verify that firewalld is running with:

```
sudo systemctl status firewalld
```

And list the currently allowed services with:

```
sudo firewall-cmd --list-all
```

This chapter has established the foundation for understanding why AlmaLinux 9 represents an exceptional choice for hosting deployments. Its lineage from RHEL provides enterprise-grade stability and security. Its community governance through the AlmaLinux OS Foundation ensures independence and transparency. Its backing by CloudLinux Inc. brings hosting industry expertise to the development process. Its binary compatibility with RHEL guarantees access to a vast ecosystem of software, tools, and documentation. And its ten-year support lifecycle provides the long-term stability that hosting infrastructure demands.

As you progress through the remaining chapters of this book, every concept, configuration, and deployment strategy will build upon the AlmaLinux 9 foundation established here. You will configure web servers, harden security, optimize performance, deploy applications, and manage the full lifecycle of a production

hosting environment. The operating system beneath all of these activities will be AlmaLinux 9, and by the end of this journey, you will understand not only how to use it but why it has become the preferred choice for hosting professionals around the world.

Chapter 2: Preparing a Production Server

The journey from a freshly installed AlmaLinux system to a production-ready web server is one that demands careful attention, methodical planning, and a deep understanding of the foundational elements that make a server reliable, secure, and performant. This chapter walks you through every critical step of that transformation. We will take your AlmaLinux 9 installation and shape it into a hardened, optimized machine that is ready to host websites, applications, and services with confidence. Every command you execute, every configuration file you edit, and every decision you make in this chapter lays the groundwork for everything that follows in this book.

Before a single web page is served or a single database query is processed, the underlying server must be prepared. Think of this process as building the foundation of a house. No matter how beautiful the architecture above ground may be, if the foundation is weak, the entire structure is compromised. In the world of AlmaLinux hosting, preparing your production server is that foundation.

Understanding the AlmaLinux 9 Minimal Installation

When you install AlmaLinux 9 for production hosting purposes, the recommended approach is to begin with a minimal installation. A minimal installation includes only the essential packages required for the operating system to function. There is

no graphical desktop environment, no unnecessary services running in the background, and no bloated software consuming resources. This philosophy aligns perfectly with production server best practices because every additional package installed on a server represents a potential security vulnerability and a consumer of system resources.

After completing a minimal installation of AlmaLinux 9, you will find yourself at a command-line interface. This is exactly where you want to be. Production servers are managed through the terminal, and becoming comfortable with this environment is essential for any system administrator.

To verify your AlmaLinux version and confirm that your installation is correct, execute the following command:

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

This command will output something similar to:

```
AlmaLinux release 9.3 (Shamrock Pampas Cat)
```

You can also gather more detailed information about your operating system using:

```
hostnamectl
```

This command displays the hostname, operating system, kernel version, architecture, and other relevant details about your AlmaLinux system. Take note of this information as it will be useful for documentation and troubleshooting purposes throughout the life of your server.

Setting the Hostname and System Identity

Every production server needs a proper hostname. The hostname serves as the identity of your machine on the network and is referenced in log files, email headers, SSL certificates, and numerous other contexts. Choosing a meaningful and consistent hostname is not merely cosmetic; it is an operational necessity.

To set the hostname on your AlmaLinux 9 server, use the following command:

```
hostnamectl set-hostname server1.yourdomain.com
```

After setting the hostname, you should also update the `/etc/hosts` file to ensure that the system can resolve its own hostname locally. Open the file with a text editor:

```
vi /etc/hosts
```

Add a line that maps your server's IP address to its hostname:

```
192.168.1.100   server1.yourdomain.com   server1
```

The following table explains the components of the `/etc/hosts` file entry:

Component	Description	Example
IP Address	The static IP address assigned to your server	192.168.1.100
FQDN	The fully qualified domain name of the server	server1.yourdomain.com
Short Name	An abbreviated alias for the hostname	server1

To verify that the hostname has been set correctly, you can run:

```
hostname
```

```
hostname -f
```

The first command displays the short hostname, while the second displays the fully qualified domain name. Both should reflect the values you configured.

Configuring Static Network Settings

Production servers must have static IP addresses. Dynamic IP addresses assigned through DHCP are unsuitable for servers because the address could change at any time, breaking all services that depend on a consistent address. On AlmaLinux 9, network configuration is managed through NetworkManager and its command-line tool, `nmcli`.

First, identify your active network connection:

```
nmcli connection show
```

This command lists all network connections configured on your system. Note the name of your active connection, which is typically something like `ens160` or `eth0`.

To configure a static IP address, execute the following series of commands, replacing the values with those appropriate for your network environment:

```
nmcli connection modify ens160 ipv4.addresses 192.168.1.100/24
nmcli connection modify ens160 ipv4.gateway 192.168.1.1
nmcli connection modify ens160 ipv4.dns "8.8.8.8 8.8.4.4"
nmcli connection modify ens160 ipv4.method manual
nmcli connection up ens160
```

The following table explains each of these commands in detail:

Command	Purpose	Explanation
`ipv4.addresses 192.168.1.100/24`	Sets the static IP and subnet mask	The /24 notation indicates a 255.255.255.0 subnet mask
`ipv4.gateway 192.168.1.1`	Sets the default gateway	This is the router address through which traffic exits the local network
`ipv4.dns "8.8.8.8 8.8.4.4"`	Sets DNS resolvers	These are Google's public DNS servers; you may use your own
`ipv4.method manual`	Disables DHCP	Tells NetworkManager to use the static configuration instead of requesting an address
`connection up ens160`	Activates the changes	Restarts the connection with the new settings applied

To verify your network configuration, use:

```
ip addr show ens160
ip route show
cat /etc/resolv.conf
```

These commands display your IP address, routing table, and DNS configuration respectively. Confirm that all values match what you configured.

Updating the System and Managing Packages

One of the most critical steps in preparing a production server is ensuring that all installed packages are up to date. Software updates contain security patches, bug

fixes, and performance improvements that are essential for a stable production environment. On AlmaLinux 9, package management is handled by the `dnf` package manager.

To perform a complete system update, execute:

```
dnf update -y
```

The `-y` flag automatically confirms all prompts, which is useful for scripting but should be used with awareness of what is being updated. On a fresh installation, this command may download and install a significant number of updates. After the update completes, it is advisable to reboot the server to ensure that the new kernel and all updated libraries are loaded:

```
reboot
```

After the reboot, verify that your system is running the latest kernel:

```
uname -r
```

Beyond updating existing packages, you should also install a set of essential utilities that will be needed throughout the server preparation process:

```
dnf install -y vim wget curl tar unzip net-tools bind-utils
policycoreutils-python-utils bash-completion
```

The following table describes each of these packages and why they are important:

Package	Description	Use Case
vim	Advanced text editor	Editing configuration files with syntax highlighting
wget	Command-line download utility	Downloading files from the internet
curl	Data transfer tool	Testing HTTP connections and APIs

tar	Archive utility	Extracting compressed archives
unzip	ZIP extraction tool	Handling ZIP format archives
net-tools	Classic networking utilities	Commands like netstat and ifconfig
bind-utils	DNS utilities	Commands like dig and nslookup for DNS troubleshooting
policycoreutils-python-utils	SELinux management tools	Managing SELinux policies and contexts
bash-completion	Tab completion for bash	Improves command-line productivity

Note: On AlmaLinux 9, the `dnf` package manager is the successor to `yum`. While `yum` commands still work as they are aliased to `dnf`, it is best practice to use `dnf` directly in all your commands and scripts.

Configuring the Firewall with firewalld

AlmaLinux 9 ships with `firewalld` as its default firewall management tool. A properly configured firewall is the first line of defense for any production server. The principle to follow is simple: deny everything by default and explicitly allow only the traffic that is necessary.

First, ensure that firewalld is running and enabled to start at boot:

```
systemctl start firewalld
systemctl enable firewalld
systemctl status firewalld
```

By default, firewalld uses the concept of zones. The default zone is typically `public`, which is appropriate for most server deployments. To check the current default zone and its configuration:

```
firewall-cmd --get-default-zone
firewall-cmd --list-all
```

For a web hosting server, you will need to allow SSH, HTTP, and HTTPS traffic. Here are the commands to open these services:

```
firewall-cmd --permanent --add-service=ssh
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --reload
```

The `--permanent` flag ensures that the rules persist across reboots. The `--reload` command applies the permanent rules to the running configuration. Without reloading, permanent rules do not take effect until the next reboot.

To verify that your firewall rules are correctly configured:

```
firewall-cmd --list-all
```

The output should show ssh, http, and https listed under the services section. If you need to open a specific port number rather than a named service, you can use:

```
firewall-cmd --permanent --add-port=8080/tcp
firewall-cmd --reload
```

The following table summarizes common firewall operations you will use on your AlmaLinux production server:

Operation	Command	Description
List all rules	`firewall-cmd --list-all`	Shows all active rules in the current zone

Add a service	`firewall-cmd --permanent --add-service=http`	Opens the port associated with a named service
Remove a service	`firewall-cmd --permanent --remove-service=http`	Closes the port associated with a named service
Add a port	`firewall-cmd --permanent --add-port=3306/tcp`	Opens a specific TCP port
Reload rules	`firewall-cmd --reload`	Applies permanent rules to the running configuration
List zones	`firewall-cmd --get-zones`	Lists all available firewall zones
Check zone of interface	`firewall-cmd --get-zone-of-interface=ens160`	Shows which zone an interface belongs to

Note: Never remove the SSH service from your firewall rules while connected remotely. Doing so will immediately lock you out of the server, and you will need physical or console access to regain control.

Configuring SELinux for Production

Security-Enhanced Linux, known as SELinux, is a mandatory access control system built into the AlmaLinux kernel. Many administrators, frustrated by SELinux denials that seem to block legitimate operations, make the mistake of disabling SELinux entirely. This is a serious security error. On a production server, SELinux should always remain in enforcing mode. Instead of disabling it, you should learn to work with it.

To check the current SELinux status:

```
getenforce
```