Ubuntu 24.04 LTS Server Administration

Deploying, Securing, and Managing Modern Ubuntu Servers in Production

Preface

Ubuntu has earned its place as the world's most widely deployed Linux distribution for servers, powering everything from small business web applications to massive cloud infrastructures at the world's largest enterprises. With the release of **Ubuntu 24.04 LTS (Noble Numbat)**, Canonical has once again delivered a platform that combines long-term stability with modern capabilities — a foundation that system administrators and engineers can trust in production for years to come.

This book, *Ubuntu 24.04 LTS Server Administration: Deploying, Securing, and Managing Modern Ubuntu Servers in Production*, was written with a single, clear purpose: **to give you the practical knowledge and confidence to deploy, secure, and manage Ubuntu servers in real-world environments.** Whether you are setting up your first Ubuntu server or managing a fleet of them in a data center, this book is designed to meet you where you are and take you further.

What This Book Covers

The journey begins with understanding what makes Ubuntu 24.04 LTS unique — its release cycle, support commitments, and the ecosystem that surrounds it. From there, we move through the full lifecycle of Ubuntu server administration: installation and initial configuration, user and group management, secure remote access via SSH, and mastering APT, Ubuntu's powerful package management system.

The heart of the book dives deep into the daily responsibilities of an Ubuntu administrator — managing services with **systemd**, configuring disks and filesystems, building resilient storage with **LVM and RAID**, and setting up networking with Ubuntu's modern tooling. You'll learn to host production web and database services, lock down your servers with firewalls and security hardening, and implement robust logging, monitoring, backup, and recovery strategies — all within the Ubuntu ecosystem.

The final chapters look forward. You'll explore automation and scripting techniques that reduce manual toil, and in the closing chapter, we bridge the gap between traditional Ubuntu server administration and the world of **DevOps engineering** — a natural evolution for today's Linux professionals.

The appendices serve as lasting reference material: a command cheat sheet tailored to Ubuntu Server, a security hardening checklist, web server configuration templates, a troubleshooting guide, and a learning roadmap to guide your continued growth.

Who This Book Is For

This book is for **system administrators, IT professionals, developers, and students** who want to build genuine competency with Ubuntu Server. Prior experience with Linux is helpful but not required — the early chapters establish a solid foundation, while later chapters challenge even experienced administrators with advanced topics and production-grade practices.

How to Use This Book

The chapters are arranged in a logical progression, and I recommend reading them in order if you are newer to Ubuntu server administration. More experienced readers may prefer to jump directly to specific topics. Every chapter emphasizes *hands-on practice* — you will get the most from this book by working alongside it on an actual Ubuntu 24.04 LTS system, whether physical, virtual, or cloud-based.

Acknowledgments

No technical book is the work of one person alone. I am grateful to the **Ubuntu community and Canonical** for building and maintaining a platform that makes Linux accessible and enterprise-ready. Thanks to the countless open-source contributors whose tools and documentation make work like this possible. I also owe a debt of gratitude to the technical reviewers, editors, and early readers whose feedback sharpened every chapter.

Most importantly, thank you — the reader — for choosing to invest your time in mastering Ubuntu server administration. The skills you develop here are not abstract; they are the same skills that keep the internet running, that protect critical data, and that power the infrastructure behind the applications the world depends on.

Ubuntu's motto, *"I am because we are,"* reminds us that we succeed together. I hope this book becomes a trusted companion on your journey.

Let's get started.

Bas van den Berg

Table of Contents

Chapter 1: Understanding Ubuntu 24.04 LTS

Ubuntu 24.04 LTS, codenamed "Noble Numbat," represents a significant milestone in the evolution of one of the most widely deployed Linux distributions in the world. Whether you are a seasoned system administrator managing hundreds of production servers or a newcomer stepping into the world of server management for the first time, understanding the foundation upon which Ubuntu 24.04 LTS is built will serve as the bedrock of every decision you make throughout the lifecycle of your infrastructure. This chapter walks you through the history, philosophy, architecture, and distinguishing features of Ubuntu 24.04 LTS, giving you the context and confidence needed to deploy, secure, and manage modern Ubuntu servers in production environments.

The Origins and Evolution of Ubuntu

To truly understand Ubuntu 24.04 LTS, we must first appreciate where it came from. Ubuntu was founded in 2004 by Mark Shuttleworth and his company, Canonical Ltd., with a bold and simple mission: to make Linux accessible to everyone. The name "Ubuntu" itself is drawn from the Southern African philosophy meaning "humanity to others," and this ethos has guided the distribution's development for two decades. Ubuntu is built upon Debian, one of the oldest and most respected Linux distributions, inheriting its robust package management system and its commitment to free software while adding a more predictable release schedule, commercial support options, and a polished user experience.

From its earliest releases, Ubuntu distinguished itself by shipping on a regular six-month cadence. Every April and October, a new version arrives, identified by its year and month of release. Ubuntu 24.04, therefore, was released in April 2024. What makes certain releases special is the "LTS" designation, which stands for Long Term Support. Every two years, the April release receives this designation, and it carries with it a commitment from Canonical to provide five years of standard security updates and maintenance, with the option to extend that support to a full twelve years through Ubuntu Pro. This predictable support lifecycle is one of the primary reasons enterprises, cloud providers, and mission-critical workloads gravitate toward Ubuntu LTS releases.

The journey from Ubuntu 4.10 "Warty Warthog" in 2004 to Ubuntu 24.04 "Noble Numbat" in 2024 spans exactly twenty years. During that time, Ubuntu has grown from a desktop-focused distribution into the dominant platform for cloud computing, container orchestration, Internet of Things deployments, and enterprise server infrastructure. According to multiple industry surveys, Ubuntu consistently ranks as the most popular Linux distribution on public cloud platforms including Amazon Web Services, Microsoft Azure, and Google Cloud Platform. Understanding this trajectory helps explain why Ubuntu 24.04 LTS ships with the features and defaults that it does.

The LTS Support Model Explained

The concept of Long Term Support is central to everything discussed in this book, so it deserves careful explanation. When Canonical designates a release as LTS, it enters a structured support lifecycle that governs how updates, patches, and fixes are delivered over time.

The following table breaks down the support phases for Ubuntu 24.04 LTS:

Support Phase	Duration	Coverage Period	What Is Included
Standard Security Maintenance	5 years	April 2024 through April 2029	Security patches for packages in the Main repository, critical bug fixes, kernel updates, and hardware enablement stacks
Extended Security Maintenance (ESM)	Additional 7 years	April 2029 through April 2036	Continued security patches for critical and high-severity vulnerabilities, available through Ubuntu Pro subscription
Total Possible Support	12 years	April 2024 through April 2036	Combined standard and extended maintenance covering the full lifecycle

This support structure has profound implications for server administrators. When you deploy Ubuntu 24.04 LTS on a production server today, you can be confident that security updates will be available for at least five years without any additional cost. For organizations that require longer lifecycles, such as those in healthcare, finance, telecommunications, or government, the Extended Security Maintenance available through Ubuntu Pro stretches that window to twelve years. This eliminates the disruptive and risky cycle of frequent operating system upgrades that plagues organizations using non-LTS releases.

It is worth noting the difference between the Main and Universe repositories in this context. The Main repository contains packages that Canonical directly maintains and supports. The Universe repository contains community-maintained packages. Under standard support, only Main repository packages receive guaranteed security updates. With Ubuntu Pro, ESM coverage extends to Universe packages as well, which is a significant benefit given that many popular server applications reside in the Universe repository.

To check whether your system is enrolled in Ubuntu Pro and to see the status of ESM, you can use the following command:

```
pro status
```

This command will display output indicating whether your machine is attached to an Ubuntu Pro subscription and which services are enabled, including ESM for Infrastructure and ESM for Applications.

To attach your system to Ubuntu Pro using a token obtained from the Ubuntu Pro dashboard, you would run:

```
sudo pro attach YOUR_TOKEN_HERE
```

Note: Ubuntu Pro is free for personal use on up to five machines. For commercial deployments, Canonical offers tiered pricing based on the number of machines and the level of support required.

What Is New in Ubuntu 24.04 LTS

Ubuntu 24.04 LTS arrives with a carefully curated set of updates that reflect the current state of the art in server computing. Every component, from the kernel to the default packages, has been selected and tested to provide a stable, secure, and performant foundation for production workloads.

The Linux kernel shipped with Ubuntu 24.04 LTS is version 6.8. This kernel brings substantial improvements in several areas that directly impact server administrators. Hardware support has been expanded to cover the latest generation of processors from Intel, AMD, and ARM-based architectures. The kernel includes improved support for the io_uring asynchronous I/O interface, which benefits high-performance database and storage workloads. Networking performance has been enhanced through updates to the TCP stack and improvements to eBPF, the extended Berkeley Packet Filter framework that is increasingly used for observability, security, and networking in modern server environments.

The following table summarizes the key software versions included in Ubuntu 24.04 LTS:

Component	**Version in Ubuntu 24.04 LTS**	**Purpose and Significance**
Linux Kernel	6.8	Core operating system kernel providing hardware abstraction, process management, and security
GCC	14	GNU Compiler Collection used for building software from source
Python	3.12	Default Python interpreter for system scripts and application development
OpenSSL	3.0	Cryptographic library providing TLS/SSL functionality for secure communications
systemd	255	Init system and service manager responsible for booting the system and managing services

GNU C Library (glibc)	2.39	Fundamental C library providing core system call interfaces
OpenSSH	9.6	Secure shell server and client for remote administration
Netplan	1.0	Network configuration abstraction layer using YAML-based configuration files
AppArmor	4.0	Mandatory access control framework for confining applications
cloud-init	24.1	Cloud instance initialization tool for configuring servers on first boot
Snapd	Latest	Snap package manager for containerized application delivery
LXD	5.21 (via snap)	System container and virtual machine manager
Podman	4.9	Daemonless container engine compatible with OCI standards

One of the most significant changes in Ubuntu 24.04 LTS is the maturation of Netplan as the definitive network configuration tool. Netplan has been the default network configuration layer in Ubuntu Server since 18.04, but version 1.0, shipping with 24.04, marks its graduation to a fully stable API. Netplan uses YAML configuration files stored in `/etc/netplan/` and abstracts the underlying network renderer, which can be either `systemd-networkd` (the default on servers) or `NetworkManager` (the default on desktops). This abstraction means that administrators write human-readable YAML files and let Netplan translate them into the appropriate backend configuration.

A basic Netplan configuration for a server with a static IP address looks like this:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 192.168.1.100/24
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

After creating or modifying a Netplan configuration file, you apply it with:

```
sudo netplan apply
```

To test a configuration before committing it permanently, which is invaluable on remote servers where a misconfiguration could lock you out, use:

```
sudo netplan try
```

This command applies the configuration temporarily and reverts it automatically after 120 seconds unless you confirm it. This is one of those features that separates Ubuntu's approach from many other distributions: thoughtful, administrator-friendly tooling that reduces the risk of catastrophic mistakes.

Note: The `netplan try` command is particularly important when working on remote servers accessed via SSH. If your network configuration change breaks connectivity, the automatic rollback ensures you are not permanently locked out of the machine.

Ubuntu Server Architecture and Design Philosophy

Ubuntu 24.04 LTS Server is designed with a minimalist philosophy. Unlike the desktop variant, the server installation is intentionally lean. The default server installation includes only the packages necessary to boot the system, manage services, configure networking, and provide remote access via SSH. There is no graphical user interface installed by default. This minimalism is not a limitation but a deliberate design choice rooted in security and performance principles. Every additional package installed on a server represents an additional attack surface and an additional consumer of system resources. By starting minimal and allowing administrators to install only what they need, Ubuntu Server ensures that production systems are as lean and secure as possible.

The init system and service manager in Ubuntu 24.04 LTS is systemd, version 255. systemd is responsible for bootstrapping the system after the kernel loads, managing services throughout the system's runtime, handling logging through its journal component, managing network configuration through systemd-networkd, and providing timer-based task scheduling as an alternative to traditional cron jobs. Understanding systemd is not optional for Ubuntu server administrators; it is fundamental.

To check the status of a service, for example the SSH server, you would use:

```
sudo systemctl status ssh
```

To enable a service so that it starts automatically at boot:

```
sudo systemctl enable ssh
```

To start a service immediately without rebooting:

```
sudo systemctl start ssh
```

To view the logs for a specific service using the systemd journal:

```
sudo journalctl -u ssh --since "1 hour ago"
```

The following table explains the most commonly used systemctl subcommands that you will use throughout your work with Ubuntu 24.04 LTS:

Command	**Purpose**	**Example**
`systemctl start <service>`	Start a service immediately	`sudo systemctl start nginx`
`systemctl stop <service>`	Stop a running service	`sudo systemctl stop nginx`
`systemctl restart <service>`	Stop and then start a service	`sudo systemctl restart nginx`
`systemctl reload <service>`	Reload configuration without stopping	`sudo systemctl reload nginx`
`systemctl enable <service>`	Enable a service to start at boot	`sudo systemctl enable nginx`
`systemctl disable <service>`	Prevent a service from starting at boot	`sudo systemctl disable nginx`
`systemctl status <service>`	Display the current status and recent logs	`sudo systemctl status nginx`
`systemctl is-active <service>`	Check if a service is currently running	`systemctl is-active nginx`
`systemctl is-enabled <service>`	Check if a service is enabled at boot	`systemctl is-enabled nginx`
`systemctl list-units --type=service`	List all loaded service units	`systemctl list-units --type=service`
`systemctl daemon-reload`	Reload systemd manager configuration after editing unit files	`sudo systemctl daemon-reload`

Package Management with APT

Ubuntu inherits its package management system from Debian, and the Advanced Package Tool, known as APT, remains the primary mechanism for installing, updating, and removing software on Ubuntu 24.04 LTS. The APT system works with `.deb` packages stored in repositories. By default, Ubuntu 24.04 LTS is configured to use Canonical's official repositories, which are divided into several components.

The following table describes the default repository components:

Repository Component	Description	Support Level
Main	Free and open source software maintained by Canonical	Fully supported with security updates for the LTS lifecycle
Restricted	Proprietary drivers and firmware necessary for hardware support	Supported by Canonical
Universe	Free and open source software maintained by the community	Community supported; ESM coverage available via Ubuntu Pro
Multiverse	Software restricted by copyright or legal issues	Not officially supported

To update the local package index, which should be done before installing or upgrading any packages:

```
sudo apt update
```

To upgrade all installed packages to their latest available versions:

```
sudo apt upgrade
```

To perform a full distribution upgrade that handles dependency changes intelligently:

```
sudo apt full-upgrade
```

To install a specific package:

```
sudo apt install package-name
```

To remove a package while leaving its configuration files in place:

```
sudo apt remove package-name
```

To remove a package along with its configuration files:

```
sudo apt purge package-name
```

To search for a package in the repositories:

```
apt search keyword
```

To display detailed information about a specific package:

```
apt show package-name
```

Note: Always run `sudo apt update` before installing or upgrading packages. This ensures that your local package index reflects the latest state of the repositories. Installing packages without first updating the index can result in dependency errors or the installation of outdated versions.

In addition to APT, Ubuntu 24.04 LTS fully supports Snap packages through the Snapd daemon. Snaps are containerized software packages that bundle the application along with all of its dependencies, providing isolation from the host system and automatic updates. Many server applications, including LXD, MicroK8s, and various database engines, are distributed as Snaps. To install a Snap package:

```
sudo snap install package-name
```

To list installed Snaps:

```
snap list
```

To update all installed Snaps:

```
sudo snap refresh
```

Ubuntu 24.04 LTS in the Cloud and Virtualization Landscape

Ubuntu 24.04 LTS is not just a server operating system that you install on bare metal. It is the most widely used guest operating system on public cloud platforms, and it has been optimized for cloud deployment from the ground up. Canonical publishes official Ubuntu images for all major cloud providers, and these images are tuned for optimal performance in virtualized and cloud environments.

The cloud-init tool, included by default in Ubuntu 24.04 LTS cloud images, is responsible for configuring a new cloud instance on its first boot. cloud-init reads configuration data from the cloud provider's metadata service and uses it to set the hostname, configure networking, create user accounts, inject SSH keys, run custom scripts, and perform any other initialization tasks specified by the administrator. This enables fully automated, hands-free server provisioning at scale.

A basic cloud-init configuration file, typically provided as user data when launching a cloud instance, might look like this:

```
#cloud-config
hostname: webserver01
manage_etc_hosts: true
users:
  - name: admin
    groups: sudo
    shell: /bin/bash
    sudo: ALL=(ALL) NOPASSWD:ALL
    ssh_authorized_keys:
      - ssh-rsa AAAAB3... your-public-key-here
package_update: true
package_upgrade: true
packages:
  - nginx
  - ufw
  - fail2ban
runcmd:
  - systemctl enable nginx
  - systemctl start nginx
  - ufw allow 'Nginx Full'
  - ufw enable
```

This single YAML file, when provided as user data to a cloud instance, will configure the hostname, create an administrative user with SSH key-based authentication, update all packages, install Nginx along with a firewall and intrusion prevention tool, and configure the firewall to allow web traffic. The entire process happens automatically during the first boot without any manual intervention.

Practical Exercise: Verifying Your Ubuntu 24.04 LTS Installation

Now that you have a thorough understanding of what Ubuntu 24.04 LTS is and what it includes, let us walk through a practical exercise to verify and explore a fresh installation. This exercise assumes you have access to an Ubuntu 24.04 LTS server, whether it is a physical machine, a virtual machine, or a cloud instance.

Step 1: Verify the Ubuntu version and release information.

```
lsb_release -a
```

The output should display:

```
Distributor ID: Ubuntu
```

```
Description:    Ubuntu 24.04 LTS
Release:        24.04
Codename:       noble
```

You can also check the contents of the os-release file:

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

Step 2: Verify the kernel version.

```
uname -r
```

The output should show a kernel version beginning with 6.8, such as `6.8.0-31-generic`.

Step 3: Check the system architecture.

```
uname -m
```

This will return `x86_64` for 64-bit Intel/AMD systems or `aarch64` for ARM-based systems.

Step 4: Review the systemd version.

```
systemctl --version
```

Step 5: Check the default Python version.

```
python3 --version
```

Step 6: Verify that the package repositories are properly configured.

```
apt policy
```

This command displays the configured repositories and their priorities.

Step 7: Check the available disk space and mounted filesystems.

```
df -h
```

Step 8: Review the system's memory usage.

```
free -h
```

Step 9: Check the Ubuntu Pro status.

```
pro status
```

Step 10: View the system boot time and uptime.

```
uptime
```

By completing these ten steps, you have verified the fundamental characteristics of your Ubuntu 24.04 LTS installation and confirmed that the system is ready for the configuration, hardening, and service deployment tasks covered in the chapters ahead.

Summary and Looking Forward

Ubuntu 24.04 LTS "Noble Numbat" stands on twenty years of continuous development, community collaboration, and enterprise hardening. It ships with Linux kernel 6.8, Python 3.12, systemd 255, and a carefully selected set of default tools that provide a secure and minimal foundation for server workloads. Its Long Term Support guarantee of five years of standard maintenance, extendable to twelve years through Ubuntu Pro, makes it suitable for the most demanding production environments. Its first-class support for cloud deployment through cloud-init, its mature network configuration through Netplan, and its robust package management through APT and Snap ensure that administrators have the tools they need to manage infrastructure at any scale.

In the chapters that follow, we will build upon this foundation by walking through the installation process in detail, configuring networking, hardening the system against threats, deploying services, managing storage, implementing monitoring, and ultimately preparing you to manage Ubuntu 24.04 LTS servers with confidence and competence in production environments. Every concept introduced in this chapter will be revisited and expanded upon as we move from understanding to implementation.

Chapter 2: Installation and Initial Configuration

Setting up an Ubuntu 24.04 LTS server properly from the very beginning is one of the most critical steps in building a reliable, secure, and performant production environment. A poorly configured installation can lead to security vulnerabilities, performance bottlenecks, and administrative headaches that compound over time. This chapter walks through every stage of the installation process, from obtaining the installation media to completing the initial configuration that transforms a bare machine into a ready-to-manage server. Whether you are deploying Ubuntu on bare metal hardware, within a virtual machine, or provisioning cloud instances, the principles and practices covered here will serve as the foundation for everything that follows in this book.

Understanding Ubuntu 24.04 LTS Server Edition

Before touching any installation media, it is important to understand what makes Ubuntu 24.04 LTS Server distinct from its desktop counterpart and from other Ubuntu releases. The "LTS" designation stands for Long Term Support, which means Canonical commits to providing security updates, bug fixes, and maintenance for a full five years from the release date, with the option to extend that support to ten years through Ubuntu Pro. This makes the 24.04 LTS release ideal for production server environments where stability and predictability are paramount.

Ubuntu Server 24.04 LTS ships without a graphical desktop environment by default. The installation image is lean, containing only the essential packages needed to boot a functional server. This minimal approach is deliberate. Every additional package installed on a server represents a potential attack surface and a resource consumer. By starting minimal, administrators retain full control over exactly what runs on their systems.

The 24.04 release, codenamed "Noble Numbat," introduces several significant changes that affect server administration. The Linux kernel ships at version 6.8, bringing improved hardware support, enhanced security features, and better performance for modern workloads. The default Python version moves to 3.12, the OpenSSL library is updated to version 3.2, and systemd reaches version 255. These updates are not merely cosmetic; they affect how services are configured, how applications are deployed, and how the system behaves under load.

Component	Version in Ubuntu 24.04 LTS	Key Changes
Linux Kernel	6.8	Improved io_uring, better Rust support, enhanced security modules
systemd	255	New service management features, improved boot performance
OpenSSL	3.2	Deprecated legacy algorithms, stronger default cipher suites
Python	3.12	Performance improvements, new syntax features
GCC	13.2	Better optimization, improved C23 support
cloud-init	24.x	Enhanced multi-cloud support, improved networking configuration

Netplan	1.0	Stable API, YAML-based network configuration
AppArmor	4.0	Improved policy management, better container support

Obtaining the Installation Media

The first practical step is downloading the official Ubuntu Server 24.04 LTS ISO image from Canonical's website. The server installation image is approximately 2.5 GB in size and is available for the amd64 (x86_64) and arm64 architectures. Always download the image from the official source at `https://ubuntu.com/download/server` or from verified mirror sites to ensure the integrity of the installation media.

After downloading the ISO, verifying its integrity is not optional in a production context. Canonical provides SHA256 checksums signed with their GPG key. The verification process involves two steps: confirming the checksum of the downloaded file matches the published checksum, and verifying that the checksum file itself was signed by Canonical.

```
# Download the SHA256SUMS file and its GPG signature
wget https://releases.ubuntu.com/24.04/SHA256SUMS
wget https://releases.ubuntu.com/24.04/SHA256SUMS.gpg

# Import Canonical's signing key
gpg --keyid-format long --keyserver hkp://keyserver.ubuntu.com --
recv-keys 0x46181433FBB75451

# Verify the signature on the checksum file
gpg --keyid-format long --verify SHA256SUMS.gpg SHA256SUMS

# Verify the ISO against the checksum
```

```
sha256sum -c SHA256SUMS 2>&1 | grep ubuntu-24.04-live-server-
amd64.iso
```

The output of the last command should display "OK" next to the filename. If it shows any other result, the downloaded file is corrupted or has been tampered with, and you must download it again.

For creating bootable USB media from another Ubuntu or Linux system, the `dd` command remains the most reliable method. This command writes the ISO image directly to the USB device at the block level, bypassing the filesystem entirely.

```
# Identify the USB device (be absolutely certain of the device
name)
lsblk

# Write the ISO to the USB device (replace /dev/sdX with your
actual device)
sudo dd if=ubuntu-24.04-live-server-amd64.iso of=/dev/sdX bs=4M
status=progress conv=fsync
```

Note: The `dd` command is sometimes called "disk destroyer" for good reason. Writing to the wrong device will irreversibly overwrite its contents. Always double-check the target device with `lsblk` before executing the command. The device should be specified as `/dev/sdX`, not `/dev/sdX1`. You are writing to the entire device, not a partition.

Walking Through the Installation Process

Ubuntu Server 24.04 LTS uses the Subiquity installer, which replaced the older Debian Installer (d-i) starting with Ubuntu 18.04 for the live server image. Subiquity provides a text-based, menu-driven interface that guides you through each config-