

THE PRODUCTION-READY DOCKER & LINUX POCKET GUIDE

An Essential Reference for Developers & Students

Complete Production Workflows, Battle-Tested Dockerfiles, Multi-Container Setup, and Security Checklists.

By Ahmed Adawy

© 2026 Ahmed Adawy. All rights reserved.

Free Sample Edition | Full Book Link: <https://leanpub.com/theproduction-readydockerlinuxpocketguide1>

Table of Contents

- **Part I: Linux CLI Essentials & VPS Management**
 - File Navigation & Permissions
 - System Resources & Port Inspection
- **Part II: Docker Core Concepts & Lifecycle Cheat Sheet**
 - Image Building & Container Execution Workflow
- **Part III: Production Dockerfiles** *(Available in Full Edition)*
- **Part IV: Multi-Container Orchestration** *(Available in Full Edition)*
- **Part V: Security Hardening & Maintenance Checklist** *(Available in Full Edition)*

Introduction: The Production-Ready Philosophy

Welcome to *The Production-Ready Docker & Linux Pocket Guide*. In the modern software engineering landscape, a critical gap exists between local development and high-performance production deployment.

This guide is specifically engineered for software developers, computer science students, and DevOps engineers who need actionable, battle-tested server configurations without wading through hundreds of pages of theory.

Part I: Linux CLI Essentials & VPS Management

Essential terminal commands required to manage Linux cloud servers (VPS) smoothly:

File Navigation & Permissions

- `pwd` : Displays current absolute directory path.
- `ls -la` : Lists all files including hidden dotfiles with exact permissions.
- `chmod +x deploy.sh` : Makes script files executable.
- `chown -R www-data:www-data /var/www` : Sets proper owner/group permissions recursively.

System Resources & Port Inspection

- `df -h` : Checks disk space usage in human-readable format (GB/MB).
- `free -m` : Displays total, used, and available RAM memory in Megabytes.
- `htop` : Interactive process monitor for CPU, memory, and running tasks.
- `netstat -tulnp` : Shows active listening ports and process IDs (PID).

Part II: Docker Core Concepts & Lifecycle Cheat Sheet

Command line workflow to manage Docker containers from build to execution:

```
# 1. Build an image with a specific version tag
docker build -t myapp:v1 .

# 2. Run container in background mapping port 8080
docker run -d -p 8080:8080 --name myapp_instance myapp:v1

# 3. Monitor active containers and view real-time logs
docker ps
docker logs -f myapp_instance

# 4. Stop and remove container safely
docker stop myapp_instance && docker rm myapp_instance
```