Docker Learning Guide - Table of Contents

Complete Docker Documentation

This comprehensive Docker guide covers everything from basics to advanced topics with detailed examples and real-world use cases.

Documentation Files

01 - Docker Introduction & Basics

Topics Covered: - What is Docker? - Why We Need Docker? - What is a Container? - Before Docker - Traditional Problems - History of Docker - What is DevOps?

Key Highlights: - Understanding containerization - Docker vs Traditional deployment - "It works on my machine" problem solved - DevOps culture and practices

---

02 - Docker Architecture & Components

Topics Covered: - Architecture of Docker - Architecture of Docker Engine - Docker vs Virtual Machines - Docker Runtime - Docker Engine Components - Open Container Initiative (OCI) - Layers in Docker

Key Highlights: - Docker client-server architecture - How Docker Engine works internally - Container runtime (containerd, runc) - Why containers are lighter than VMs - Image layer system explained

---

03 - Docker Installation

Topics Covered: - Docker Installation on Windows - Docker Installation on macOS - Docker Installation on Linux - Docker Desktop - Post-Installation Setup - Verification and Testing

Key Highlights: - Step-by-step installation for all platforms - WSL 2 setup for Windows - Apple Silicon (M1/M2/M3) considerations - Troubleshooting common installation issues - Docker Desktop features and settings

---

04 - Docker CLI & Commands

Topics Covered: - Docker CLI Overview - How the CLI Works - Docker Image Commands - Docker Container Commands - Docker Network Commands - Docker Volume Commands - Docker System Commands - Additional Docker Commands

Key Highlights: - Complete command reference - Running containers in interactive mode - Accessing containers locally - Managing images and containers - System cleanup and maintenance - Docker commit explained

---

05 - Docker Images, Dockerfile & Building

Topics Covered: - What is a Dockerfile? - What is a Docker Image? - Difference Between Dockerfile and Image - How to Create a Dockerfile - Dockerfile Instructions (FROM, RUN, COPY, CMD, etc.) - How to Build Docker Images - Building Real Projects (React, Node.js, Python, Java) - Best Practices - Multi-Stage Builds

Key Highlights: - Creating Dockerfiles from scratch - Building optimized images - Layer caching and optimization - Real-world project examples - Security best practices - Multi-stage builds for smaller images

---

06 - Docker Volumes, Networks & Compose

Topics Covered: - Docker Volumes - Mount Binds in Docker - Docker Networks - What is Docker Compose - Docker Compose Examples - Predefined Images - Push and Pull Images (DockerHub)

Key Highlights: - Persistent data with volumes - Development with bind mounts - Container networking and communication - Multi-container applications with Compose - Working with Docker Hub - Real-world compose examples (MERN, WordPress, Microservices)

Learning Path

Beginner Level (Week 1-2)

1. Start with **File 01** - Understand what Docker is and why it exists
2. Read **File 02** - Learn Docker architecture
3. Follow **File 03** - Install Docker on your system
4. Practice **File 04** - Master basic CLI commands

Intermediate Level (Week 3-4)

5. Study **File 05** - Create Dockerfiles and build images
6. Build real projects from examples
7. Learn **File 06** - Work with volumes, networks, and Compose

8. Create multi-container applications

Advanced Level (Week 5+)

9. Multi-stage builds for production
10. Docker security best practices
11. CI/CD integration
12. Container orchestration (Kubernetes)

Quick Reference

Most Used Commands

```
# Images
docker pull <image>
docker build -t <name> .
docker images
docker rmi <image>

# Containers
docker run <image>
docker ps
docker stop <container>
docker rm <container>
docker logs <container>
docker exec -it <container> bash

# Networks
docker network ls
docker network create <network>
docker network connect <network> <container>

# Volumes
docker volume create <volume>
docker volume ls
docker volume inspect <volume>

# Compose
docker compose up -d
docker compose down
docker compose logs -f
```

Topics by Timeline Reference

Based on your original video timestamps:

Docker Basics (00:00 - 17:38)

- **File 01**: Introduction, What is Docker, Why Docker, Containers
- **File 02**: Architecture, Docker vs VM, Flow of Docker

Installation (17:38 - 30:44)

- **File 03**: Windows, Mac, and Linux installation

Building & Images (30:44 - 01:35:43)

- **File 04**: Docker CLI commands
- **File 05**: Creating Dockerfile, Building images, Managing containers
- **File 05**: Pre-defined images, Interactive mode, Push to DockerHub

Advanced Topics (01:38:21 - 03:03:45)

- **File 06**: Docker Volumes
- **File 06**: Mount Binds
- **File 06**: Working with APIs, Multiple Containers
- **File 06**: Docker Networks
- **File 06**: Docker Compose

Practical Examples Included

Complete Projects

- Python Flask Application
- Node.js Express Server
- React Web Application
- Java Spring Boot Application
- Full-Stack MERN Application
- WordPress with MySQL
- Microservices Architecture

Database Examples

- PostgreSQL
- MySQL
- MongoDB
- Redis

Real-World Scenarios

- Development environments with hot-reload
- Multi-tier applications
- CI/CD pipelines
- Data persistence
- Network isolation
- Scaling applications

Learning Tips

1. **Practice Along**: Don't just read - type every command
2. **Build Projects**: Apply knowledge to real projects
3. **Understand Why**: Don't memorize commands, understand concepts
4. **Experiment**: Break things and fix them
5. **Use Version Control**: Keep your Dockerfiles in Git
6. **Check Official Docs**: Reference Docker documentation
7. **Join Community**: Engage with Docker community

Additional Resources

- Official Docker Documentation
- Docker Hub
- Docker Samples
- Play with Docker
- Docker Cheat Sheet

Checklist for Mastery

☐ Understand containerization concept
☐ Install Docker on your system
☐ Run your first container
☐ Create a Dockerfile
☐ Build a custom image
☐ Push image to Docker Hub
☐ Use volumes for persistence
☐ Create custom networks
☐ Write docker-compose.yml
☐ Deploy multi-container app
☐ Implement multi-stage builds
☐ Apply security best practices
☐ Set up development environment with Docker

☐ Create production-ready images

Next Steps After Completing This Guide

1. **Container Orchestration**
   - Learn Kubernetes
   - Study Docker Swarm
   - Explore container orchestration patterns
2. **CI/CD Integration**
   - Integrate Docker with Jenkins
   - Use GitHub Actions for Docker builds
   - Set up automated deployments
3. **Production Deployment**
   - Cloud platforms (AWS ECS, Azure Container Instances, Google Cloud Run)
   - Monitoring and logging
   - Security scanning and hardening
4. **Advanced Topics**
   - Docker plugins
   - Custom network drivers
   - Docker secrets and configs
   - Performance optimization

Need Help?

If you encounter issues: 1. Check the troubleshooting sections in each file 2. Review Docker official documentation 3. Search Docker forums and Stack Overflow 4. Join Docker community Slack

---

Congratulations!

You now have access to a complete Docker learning resource covering all essential topics with practical examples. Take your time, practice regularly, and build real projects to solidify your understanding.

Happy Dockerizing!

Last Updated: October 2025 Documentation Version: 1.0