

Inhalt

Vorwort 11

TEIL I Einführung

1 Hello World 17

1.1 Docker-Schnellinstallation 17

1.2 Apache mit PHP 8 18

1.3 Node.js 21

1.4 Python 24

1.5 Fazit 25

2 Installation 27

2.1 Viele Wege führen zum Ziel 27

2.2 Docker-Installation unter Windows 31

2.3 Docker-Installation unter macOS 33

2.4 Docker-Installation unter Linux 34

2.5 Rootless Docker 38

2.6 Docker Desktop unter Linux 43

2.7 Podman installieren 45

3 Grundlagen 49

3.1 Grundlagen und Nomenklatur 49

3.2 Container ausführen 55

3.3 Container interaktiv verwenden 58

3.4 Portweiterleitung 65

3.5 Datenspeicherung in Volumes 68

3.6 Volumes mit Namen 75

3.7 Volumes in eigenen Verzeichnissen 76

3.8 Kommunikation zwischen mehreren Containern 78

3.9 Administration 85

4 **Eigene Images** 91

4.1 Hello, Dockerfile! 92

4.2 Dockerfile-Syntax 94

4.3 Ein eigenes Webserver-Image 100

4.4 Images in den Docker Hub hochladen 103

4.5 Multi-Arch-Images 106

4.6 Beispiel: Pandoc- und LaTeX-Umgebung als Image einrichten 109

5 **Container-Setups mit »compose«** 113

5.1 YAML-Syntax 115

5.2 Hello Compose! 116

5.3 Die Datei compose.yaml 123

5.4 Passwörter und andere Geheimnisse 131

5.5 Neue Projekte einrichten (docker init) 132

6 **Tipps, Tricks und Interna** 135

6.1 Docker Desktop und Podman Desktop 136

6.2 Visual Studio Code 139

6.3 Portainer 142

6.4 Pull-Limit im Docker Hub 144

6.5 Unterschiedliche CPU-Architekturen nutzen 149

6.6 Container automatisch starten 152

6.7 Docker-Interna 157

6.8 Podman-Interna 170

7 **Kommandoreferenz** 179

TEIL II Werkzeugkasten

8 **Alpine Linux** 205

8.1 Merkmale 206

8.2 Paketverwaltung mit apk 209

9	Webserver und Co.	213
9.1	Apache HTTP Server	213
9.2	Nginx	219
9.3	Nginx als Reverse Proxy mit SSL-Zertifikaten von Let's Encrypt	222
9.4	Caddy	230
9.5	Node.js mit Express	232
9.6	HAProxy	237
9.7	Traefik-Proxy	239
10	Datenbanksysteme	245
10.1	MySQL und MariaDB	245
10.2	PostgreSQL	251
10.3	MongoDB	256
10.4	Redis	263
11	Programmiersprachen	267
11.1	JavaScript (Node.js)	267
11.2	Java	271
11.3	PHP	274
11.4	Ruby	280
11.5	Python	281
11.6	Go	288
12	Webapplikationen und CMS	297
12.1	WordPress	297
12.2	Nextcloud	306
12.3	Joomla	308

TEIL III Praxis

13	Eine moderne Webapplikation	313
13.1	Die Anwendung	314
13.2	Das Frontend – Vue.js	316
13.3	Der API-Server – Node.js Express	326

13.4	Die MongoDB-Datenbank	336
13.5	Der Sessionspeicher – Redis	341
14	Grafana	343
14.1	Grafana-Docker-Setup	344
14.2	Provisioning	354
14.3	Ein angepasstes Telegraf-Image	357
15	Modernisierung einer traditionellen Applikation	363
15.1	Die bestehende Applikation	364
15.2	Planung und Vorbereitung	366
15.3	Die Entwicklungsumgebung	380
15.4	Produktivumgebung und Migration	381
15.5	Updates	384
15.6	Tipps für die Umstellung	385
15.7	Fazit	386
16	GitLab	387
16.1	GitLab-Schnellstart	389
16.2	GitLab-Webinstallation	390
16.3	HTTPS über ein Reverse-Proxy-Setup	392
16.4	E-Mail-Versand	393
16.5	SSH-Zugriff	396
16.6	Volumes und Backup	397
16.7	Eigene Docker-Registry für GitLab	399
16.8	Die vollständige compose-Datei	401
16.9	GitLab verwenden	403
16.10	GitLab-Runner	407
16.11	Mattermost	410
17	Continuous Integration und Continuous Delivery	417
17.1	Die Website dockerbuch.info mit gohugo.io	418
17.2	Docker-Images für die CI/CD-Pipeline	423
17.3	Die CI/CD-Pipeline	426

18 Sicherheit 437

18.1 Softwareinstallation 437

18.2 Herkunft der Docker-Images 439

18.3 »root« in Docker-Images 442

18.4 Der Docker-Dämon 443

18.5 User Namespaces 445

18.6 cgroups 447

18.7 Secure Computing Mode 448

18.8 AppArmor-Sicherheitsprofile 449

19 Swarm 451

19.1 Docker Swarm 453

19.2 Docker Swarm in der Hetzner-Cloud 458

20 Kubernetes 469

20.1 Minikube 470

20.2 Amazon EKS (Elastic Kubernetes Service) 482

20.3 Microsoft AKS (Azure Kubernetes Service) 486

20.4 Google Kubernetes Engine 495

Index 505