

Vorwort	13
Einleitung	17
1 Einführung	23
Die Entstehung von Docker	23
Das Docker-Versprechen	23
Vorteile des Docker-Workflows	26
Was Docker nicht ist	28
Wichtige Begrifflichkeiten	30
Zusammenfassung	31
2 Docker im Überblick	33
Workflows vereinfachen	33
Umfassender Support und breite Akzeptanz	36
Architektur	38
Das Client-Server-Modell	39
Netzwerkports und Unix-Sockets	40
Stabiles Tooling	40
Dockers Kommandozeilentool	41
Docker-Engine-API	42
Containernetzwerk	42
Docker ausreizen	44
Container sind keine virtuellen Maschinen	45
Beschränkte Isolierung	45
Container sind leichtgewichtig	46
Unveränderliche Infrastruktur	47
Zustandslose Anwendungen	48
Zustände externalisieren	48

Der Docker-Workflow	49
Versionsverwaltung	50
Anwendungen erstellen	52
Testen	53
Paketierung	54
Deployment	54
Das Docker-Ökosystem	55
Zusammenfassung	57
3 Docker installieren	59
Der Docker-Client	60
Linux	60
macOS	63
Microsoft Windows 11	63
Der Docker-Server	65
Linux mit systemd	65
Server, die nicht auf Linux-VMs basieren	65
Installation testen	70
Ubuntu	71
Fedora	71
Alpine Linux	71
Docker-Server erkunden	71
Zusammenfassung	73
4 Docker-Images verwenden	75
Der Aufbau eines Dockerfiles	76
Erstellen eines Image	79
Ausführen eines Image	82
Build-Argumente	82
Umgebungsvariablen als Konfiguration	83
Benutzerdefinierte Base-Images	84
Images speichern	85
Öffentliche Registries	85
Private Registries	86
Authentifizierung	86
Eine private Registry betreiben	91
Images optimieren	95
Den Layer-Cache nutzen	105
Directory Caching	109
Fehlerbehebung bei fehlgeschlagenen Builds	114
Pre-BuildKit-Images debuggen	114
BuildKit-Images debuggen	116
Multi-Architektur-Builds	118
So geht es weiter	123

5	Docker-Container verwenden	125
	Was sind Container?	125
	Die Entstehungsgeschichte der Container	126
	Container erstellen	128
	Grundlegende Konfiguration	128
	Speicher-Volumes	133
	Ressourcen-Quotas	135
	Container starten	144
	Container automatisch neu starten	145
	Container stoppen	146
	Container sofort beenden	148
	Ausführung eines Containers pausieren und fortsetzen	148
	Container und Images aufräumen	150
	Windows-Container	151
	So geht es weiter	155
6	Docker erkunden	157
	Ausgabe der Docker-Version	157
	Informationen über den Server	159
	Image-Updates herunterladen	161
	Container inspizieren	162
	Die Shell erkunden	163
	Ausgabe von Rückgabewerten	164
	In einen laufenden Container gelangen	165
	docker container exec	166
	docker volume	167
	Logging	169
	docker container logs	169
	Fortgeschrittenes Logging	171
	Docker überwachen	173
	Containerstatistiken	173
	Container-Health-Checks	177
	docker system events	180
	cAdvisor	182
	Monitoring mit Prometheus	184
	Weitere Erkundung	187
	So geht es weiter	187
7	Container debuggen	189
	Prozesse anzeigen	190
	Prozesse inspizieren	195
	Prozessverwaltung	196
	Das Netzwerk inspizieren	199

Image-History	202
Inspizieren eines Containers	203
Dateisystem inspizieren	204
So geht es weiter	205
8 Docker Compose	207
Docker Compose konfigurieren	208
Services starten	216
Rocket.Chat	218
Weitere Docker-Compose-Features	227
Die Konfiguration managen	228
Standardwerte	229
Pflichtwerte	231
Die Datei dotenv	231
So geht es weiter	233
9 Der Weg zu Containern in Produktivumgebungen	235
Einstieg in die Produktion	235
Dockers Rolle in Produktivumgebungen	236
Prozesskontrolle	238
Beschränkung der Ressourcen	239
Netzwerke	239
Konfiguration	240
Paketierung und Auslieferung	241
Logging	241
Monitoring	241
Scheduling	242
Service Discovery	244
Fazit zur Produktion	246
Docker und die DevOps-Pipeline	247
Kurzübersicht	247
Externe Abhängigkeiten	251
So geht es weiter	251
10 Skalierung	253
Docker Swarm Mode	254
Kubernetes	265
Minikube	266
In Docker Desktop integriertes Kubernetes	286
Kind	287
Amazon ECS und Fargate	289
Einrichten von AWS	290
Einrichtung von IAM-Rollen	290
Einrichtung der AWS-CLI-Tools	291

Containerinstanzen	293
Tasks	293
Testen des Tasks	301
Task stoppen	301
Zusammenfassung	303
11 Weiterführende Themen	305
Container im Detail	305
Control Groups (cgroups)	306
Namespaces	311
Sicherheitsaspekte	315
UID 0	316
Rootless Mode	319
Privilegierte Container	323
Secure Computing Mode	326
SELinux und AppArmor	331
Wie sicher ist der Docker-Daemon?	332
Erweiterte Konfiguration	333
Netzwerke	334
Storage	341
nsenter	345
Container ohne Shell debuggen	346
Die Struktur von Docker	348
Runtimes austauschen	353
gVisor	353
Zusammenfassung	356
12 Das wachsende Ökosystem	357
Clienttools	357
nerdctl	357
podman und buildah	359
All-in-one-Entwicklungstools	361
Rancher Desktop	361
Podman Desktop	361
Zusammenfassung	363
13 Container in der Produktivumgebung	365
The Twelve-Factor App	366
Codebasis	366
Abhängigkeiten	367
Konfiguration	368
Unterstützende Services	370
Build, Release und Ausführung	371
Prozesse	371

Portanbindung	372
Nebenläufigkeit	372
Austauschbarkeit	373
Gleichstellung von Entwicklungs- und Produktivumgebung	373
Logs	374
Verwaltungsvorgänge	375
»Twelve-Factor«-Zusammenfassung	375
The Reactive Manifesto	375
Reaktionsschnell	376
Belastbar	376
Flexibel	376
Nachrichtengesteuert	376
Zusammenfassung	377
14 Schlusswort	379
Der Blick voraus	379
Herausforderungen	380
Der Docker-Workflow	381
Minimierung der Deployment-Artefakte	382
Speicherung und Abruf optimieren	382
Der Lohn der Mühe	383
Zum Abschluss	384
Index	385