

Inhaltsverzeichnis

	Vorwort	13
	Über die Autoren	17
	Einleitung	19
1	Einführung	23
1.1	Die Entstehung von Docker	23
1.2	Das Docker-Versprechen	23
	1.2.1 Vorteile des Docker-Workflows	26
1.3	Was Docker nicht ist	28
2	Docker im Überblick	31
2.1	Workflows vereinfachen	31
2.2	Umfassender Support und breite Akzeptanz	34
2.3	Architektur	35
	2.3.1 Das Client/Server-Modell	35
	2.3.2 Netzwerkports und Unix-Sockets	36
	2.3.3 Stabile Toolboxes	36
	2.3.4 Dockers Kommandozeilentool	37
	2.3.5 API (Application Programming Interface)	37
	2.3.6 Containervernetzung	38
2.4	Docker ausreizen	39
	2.4.1 Container sind keine virtuellen Maschinen	40
	2.4.2 Container sind leichtgewichtig	41
	2.4.3 Unveränderliche Infrastruktur	41
	2.4.4 Beschränkte Isolierung	42
	2.4.5 Zustandslose Anwendungen	43
	2.4.6 Zustände externalisieren	44
2.5	Der Docker-Workflow	45
	2.5.1 Versionsverwaltung	45
	2.5.2 Anwendungen erstellen	47
	2.5.3 Testen	47
	2.5.4 Paketerstellung	48

2.5.5	Deployment	49
2.5.6	Das Docker-Ökosystem	49
2.6	Zusammenfassung	52
3	Docker installieren	53
3.1	Wichtige Begriffe	53
3.2	Der Docker-Client	54
3.2.1	Linux	55
3.2.2	Mac OS X 10	58
3.2.3	Microsoft Windows	59
3.3	Der Docker-Server	60
3.3.1	Linux mit systemd	61
3.3.2	Linux mit upstart	61
3.3.3	Linux mit init.d	61
3.3.4	VM-basierte Server	62
3.4	Installation testen	74
3.4.1	Ubuntu	74
3.4.2	Fedora	74
3.4.3	CentOS	74
3.5	Zusammenfassung	75
4	Docker-Images verwenden	77
4.1	Der Aufbau eines Dockerfiles	77
4.2	Erstellen eines Images	81
4.3	Ausführen eines Images	87
4.3.1	Umgebungsvariablen	88
4.4	Benutzerdefinierte Base-Images	89
4.5	Images speichern	90
4.5.1	Öffentliche Registries	90
4.5.2	Private Registries	91
4.5.3	Authentifizierung	92
4.5.4	Mirroring einer Registry	96
4.5.5	Weitere Ansätze bei der Auslieferung von Images	100
5	Docker-Container verwenden	103
5.1	Was sind Container?	103
5.1.1	Die Entstehungsgeschichte der Container	104
5.2	Container erstellen	106
5.2.1	Grundlegende Konfiguration	106

5.2.2	Speichervolumen	111
5.2.3	Ressourcenkontingente	114
5.3	Container starten	122
5.4	Container automatisch neu starten	124
5.5	Container stoppen	125
5.6	Container sofort beenden	126
5.7	Ausführung eines Containers pausieren und fortsetzen	127
5.8	Container und Images aufräumen	128
5.9	Wie geht es weiter?	130
6	Docker erkunden	131
6.1	Ausgabe der Docker-Version	131
6.2	Informationen über den Server	133
6.3	Image-Updates herunterladen	134
6.4	Container inspizieren	135
6.5	In einen laufenden Container gelangen	137
6.5.1	docker exec	137
6.5.2	nsenter	138
6.6	Die Shell erkunden	142
6.7	Rückgabe von Ergebnissen	142
6.8	Docker Logs	144
6.9	Docker überwachen	149
6.9.1	Containerstatistiken	150
6.9.2	Docker Events	154
6.9.3	cAdvisor	156
6.10	Weitere Erkundung	160
7	Container ausliefern	161
7.1	Deployment	161
7.1.1	Toolklassen	162
7.1.2	Orchestrierungstools	163
7.1.3	Distributed Schedulers	164
7.1.4	Deployment: Fazit	166
7.2	Container testen	166
7.2.1	Kurzübersicht	167
7.2.2	Externe Abhängigkeiten	170
8	Container debuggen	173
8.1	Prozesse anzeigen	173
8.2	Prozesse inspizieren	178

8.3	Prozessverwaltung	180
8.4	Das Netzwerk inspizieren	181
8.5	Image History	182
8.6	Inspizieren eines Containers	183
8.7	Dateisystem inspizieren	185
8.8	So geht es weiter	186
9	Skalierung	187
9.1	Docker Swarm	188
9.2	Centurion	194
9.3	Amazon EC2 Container Service	200
	9.3.1 Einrichtung von IAM-Rollen	200
	9.3.2 Einrichtung der AWS-CLI-Tools	201
	9.3.3 Containerinstanzen	203
	9.3.4 Tasks	207
	9.3.5 Testen des Tasks	213
	9.3.6 Task stoppen	214
9.4	Zusammenfassung	216
10	Weiterführende Themen	217
10.1	Austauschbare Backends	217
	10.1.1 Execution Driver	218
	10.1.2 Storage	220
10.2	Container im Detail	224
	10.2.1 Control Groups (cgroups)	224
	10.2.2 Kernel- und Benutzernamensräume	228
10.3	Sicherheitsaspekte	233
	10.3.1 Wie sicher ist Ihr Container?	233
	10.3.2 Wie sicher ist der Docker-Daemon?	240
10.4	Vernetzung	242
11	Container in der Produktivumgebung	249
11.1	The Twelve-Factor App	250
	11.1.1 Codebasis	250
	11.1.2 Abhängigkeiten	250
	11.1.3 Konfiguration	253
	11.1.4 Unterstützende Dienste	255
	11.1.5 Build, Release und Ausführung	256
	11.1.6 Prozesse	256

11.1.7	Portanbindung	257
11.1.8	Nebenläufigkeit	257
11.1.9	Austauschbarkeit	258
11.1.10	Gleichstellung von Entwicklungs- und Produktivumgebung	258
11.1.11	Logs	259
11.1.12	Verwaltungsvorgänge	259
11.1.13	»Twelve-Factor«-Zusammenfassung	260
11.2	The Reactive Manifesto	260
11.2.1	Reaktionsschnell	260
11.2.2	Belastbar	261
11.2.3	Flexibel	261
11.2.4	Nachrichtengesteuert	261
11.3	Fazit	261
12	Schlussfolgerungen	263
12.1	Herausforderungen	263
12.2	Der Docker-Workflow	264
12.3	Minimierung der Deployment-Artefakte	265
12.4	Speicherung und Abruf optimieren	265
12.5	Der Lohn der Mühe	266
12.6	Schlusswort	267
	Stichwortverzeichnis	269