

Kubernetes

Eine kompakte Einführung

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

	Einleitung	xvii
1	Einführung	1
1.1	Schnelligkeit	2
1.1.1	Der Wert der Immutabilität	3
1.1.2	Deklarative Konfiguration	4
1.1.3	Selbstheilende Systeme	5
1.2	Ihren Service und Ihre Teams skalieren	6
1.2.1	Entkoppeln	6
1.2.2	Einfaches Skalieren für Anwendungen und Cluster	6
1.2.3	Entwicklungs-Teams mit Microservices skalieren	7
1.2.4	Konsistenz und Skalierung durch Separation of Concerns	8
1.3	Abstrahieren Sie Ihre Infrastruktur	10
1.4	Effizienz	11
1.5	Cloud-natives Ökosystem	12
1.6	Zusammenfassung	13
2	Container erstellen und ausführen	15
2.1	Container-Images	16
2.2	Anwendungs-Images mit Docker bauen	18
2.2.1	Dockerfiles	18
2.2.2	Die Image-Größe optimieren	20
2.2.3	Sicherheit von Images	21
2.3	Multistage Image Build	22
2.4	Images in einer Remote-Registry ablegen	24
2.5	Die Docker Container Runtime	25
2.5.1	Container mit Docker ausführen	26
2.5.2	Die kuard-Anwendung erforschen	26
2.5.3	Den Ressourcen-Einsatz begrenzen	26

2.6	Aufräumen	27
2.7	Zusammenfassung	28
3	Ein Kubernetes-Cluster deployen	29
3.1	Kubernetes auf einem öffentlichen Cloud-Provider installieren	30
3.1.1	Google Kubernetes Engine	30
3.1.2	Kubernetes mit dem Azure Kubernetes Service installieren	30
3.1.3	Kubernetes auf den Amazon Web Services installieren	31
3.1.4	Kubernetes mit minikube lokal installieren	31
3.2	Kubernetes in Docker ausführen	32
3.3	Der Kubernetes-Client	33
3.3.1	Den Cluster-Status prüfen	33
3.3.2	Worker-Knoten in Kubernetes auflisten	34
3.4	Cluster-Komponenten	36
3.4.1	Kubernetes-Proxy	36
3.4.2	Kubernetes-DNS	37
3.4.3	Kubernetes-UI	37
3.5	Zusammenfassung	37
4	Häufige kubectl-Befehle	39
4.1	Namensräume	39
4.2	Kontexte	39
4.3	Objekte der Kubernetes-API anzeigen	40
4.4	Kubernetes-Objekte erstellen, aktualisieren und löschen	41
4.5	Objekte mit einem Label und Anmerkungen versehen	42
4.6	Debugging-Befehle	43
4.7	Cluster-Management	44
4.8	Autovervollständigen von Befehlen	45
4.9	Alternative Möglichkeiten zur Kommunikation mit Ihrem Cluster	45
4.10	Zusammenfassung	46
5	Pods	47
5.1	Pods in Kubernetes	48
5.2	In Pods denken	48
5.3	Das Pod-Manifest	49
5.3.1	Einen Pod erstellen	50
5.3.2	Ein Pod-Manifest schreiben	50

5.4	Pods starten	51
5.4.1	Pods auflisten	52
5.4.2	Pod-Details	52
5.4.3	Einen Pod löschen	53
5.5	Auf Ihren Pod zugreifen	54
5.5.1	Mehr Informationen aus Logs erhalten	54
5.5.2	Befehle in Ihrem Container mit exec ausführen	55
5.5.3	Dateien von und auf Container kopieren	55
5.6	Health-Checks	55
5.6.1	Liveness-Probe	56
5.6.2	Readiness-Probe	57
5.6.3	Startup-Probe	57
5.6.4	Ausgefeiltere Proben-Konfiguration	58
5.6.5	Andere Arten von Health-Checks	58
5.7	Ressourcen-Management	58
5.7.1	Ressourcen-Anforderungen: Minimal notwendige Ressourcen	59
5.7.2	Den Ressourcen-Einsatz durch Grenzen beschränken	61
5.8	Daten mit Volumes persistieren	62
5.8.1	Volumes in Pods definieren	62
5.8.2	Volumes in Pods nutzen	63
5.9	Fügen Sie alles zusammen	64
5.10	Zusammenfassung	66
6	Labels und Anmerkungen	67
6.1	Labels	67
6.1.1	Labels anwenden	68
6.1.2	Labels anpassen	70
6.1.3	Label-Selektoren	70
6.1.4	Label-Selektoren in API-Objekten	72
6.1.5	Labels in der Architektur von Kubernetes	73
6.2	Anmerkungen	73
6.3	Aufräumen	75
6.4	Zusammenfassung	75
7	Service-Discovery	77
7.1	Was ist Service-Discovery?	77
7.2	Das Service-Objekt	78
7.2.1	Service-DNS	79
7.2.2	Readiness-Checks	80

7.3	Über das Cluster hinausschauen	82
7.4	Load-Balancer-Integration	83
7.5	Weitere Details	85
7.5.1	Endpunkte	85
7.5.2	Manuelle Service-Discovery	86
7.5.3	kube-proxy und Cluster-IPs	87
7.5.4	Umgebungsvariablen zur Cluster-IP	88
7.6	Mit anderen Umgebungen verbinden	89
7.6.1	Mit einer Ressource außerhalb eines Clusters verbinden	89
7.6.2	Externe Ressourcen mit Services innerhalb eines Clusters verbinden	90
7.7	Aufräumen	90
7.8	Zusammenfassung	91
8	HTTP Load Balancing mit Ingress	93
8.1	Ingress-Spec versus Ingress-Controller	94
8.2	Contour installieren	95
8.2.1	DNS konfigurieren	96
8.2.2	Eine lokale hosts-Datei konfigurieren	96
8.3	Ingress verwenden	97
8.3.1	Einfachste Anwendung	97
8.3.2	Hostnamen verwenden	98
8.3.3	Pfade verwenden	100
8.3.4	Aufräumen	101
8.4	Fortgeschrittenere Themen und Probleme mit Ingress	101
8.4.1	Mehrere Ingress-Controller laufen lassen	101
8.4.2	Mehrere Ingress-Objekte	102
8.4.3	Ingress und Namensräume	102
8.4.4	Path Rewriting	103
8.4.5	TLS	103
8.5	Alternative Ingress-Implementierungen	104
8.6	Die Zukunft von Ingress	105
8.7	Zusammenfassung	106
9	ReplicaSets	107
9.1	Reconciliation-Schleifen	108
9.2	Die Verbindung zwischen Pods und ReplicaSets	108
9.2.1	Bestehende Container übernehmen	109
9.2.2	Container in Quarantäne stecken	109
9.3	Mit ReplicaSets designen	109

9.4	Spezifikation eines ReplicaSets	110
9.4.1	Pod-Templates	110
9.4.2	Labels	111
9.5	Ein ReplicaSet erstellen	111
9.6	Ein ReplicaSet untersuchen	112
9.6.1	Ein ReplicaSet über einen Pod finden	112
9.6.2	Eine Gruppe von Pods für ein ReplicaSet finden	113
9.7	ReplicaSets skalieren	113
9.7.1	Imperatives Skalieren mit kubectl scale	113
9.7.2	Deklaratives Skalieren mit kubectl apply	114
9.7.3	Ein ReplicaSet automatisch skalieren	115
9.8	ReplicaSets löschen	116
9.9	Zusammenfassung	116
10	Deployments	117
10.1	Ihr erstes Deployment	118
10.2	Deployments erstellen	120
10.3	Deployments verwalten	121
10.4	Deployments aktualisieren	122
10.4.1	Ein Deployment skalieren	122
10.4.2	Ein Container-Image aktualisieren	123
10.4.3	Rollout-History	124
10.5	Deployment-Strategien	127
10.5.1	Recreate-Strategie	127
10.5.2	RollingUpdate-Strategie	127
10.5.3	Rollouts verlangsamen, um die Service-Qualität sicherzustellen	130
10.6	Ein Deployment löschen	132
10.7	Ein Deployment überwachen	133
10.8	Zusammenfassung	133
11	DaemonSets	135
11.1	Der DaemonSet-Scheduler	136
11.2	DaemonSets erstellen	137
11.3	DaemonSets auf bestimmte Knoten beschränken	139
11.3.1	Knoten mit Labels versehen	139
11.3.2	Knoten-Selektoren	139
11.4	Ein DaemonSet aktualisieren	141
11.5	Ein DaemonSet löschen	142
11.6	Zusammenfassung	142

12	Jobs	143
12.1	Das Job-Objekt	143
12.2	Job-Muster	144
12.2.1	Einmalig	144
12.2.2	Parallelism	149
12.2.3	Work-Queues	150
12.3	CronJobs	154
12.4	Zusammenfassung	155
13	ConfigMaps und Secrets	157
13.1	ConfigMaps	157
13.1.1	ConfigMaps erstellen	157
13.1.2	Eine ConfigMap verwenden	158
13.2	Secrets	161
13.2.1	Secrets erstellen	162
13.2.2	Secrets konsumieren	163
13.2.3	Private Docker-Registries	164
13.3	Namensbeschränkungen	165
13.4	ConfigMaps und Secrets managen	166
13.4.1	Ausgabe	166
13.4.2	Erstellen	167
13.4.3	Aktualisieren	167
13.5	Zusammenfassung	169
14	Role-Based Access Control für Kubernetes	171
14.1	Role-Based Access Control	172
14.1.1	Identität in Kubernetes	172
14.1.2	Rollen und Role Bindings verstehen	173
14.1.3	Rollen und Role Bindings in Kubernetes	174
14.2	Techniken zur Arbeit mit RBAC	176
14.2.1	Die Autorisierung mit can-i testen	176
14.2.2	RBAC in der Versionsverwaltung managen	177
14.3	Fortgeschrittene Techniken	177
14.3.1	Cluster-Rollen aggregieren	177
14.3.2	Gruppen für Bindings verwenden	178
14.4	Zusammenfassung	179
15	Service Meshes	181
15.1	Verschlüsselung und Authentifizierung mit Mutual TLS	182
15.2	Traffic Shaping	182
15.3	Introspection	183

15.4	Brauchen Sie wirklich ein Service Mesh?	184
15.5	Introspection einer Service-Mesh-Implementierung	184
15.6	Service-Mesh-Landschaft	185
15.7	Zusammenfassung	186
16	Storage-Lösungen in Kubernetes integrieren	187
16.1	Externe Services importieren	188
16.1.1	Services ohne Selektoren	189
16.1.2	Grenzen für externe Services: Health-Checking	191
16.2	Zuverlässige Singletons ausführen	191
16.2.1	Ein MySQL-Singleton ausführen	192
16.2.2	Dynamisches Volume-Provisioning	195
16.3	Kubernetes-eigenes Storage mit StatefulSets	197
16.3.1	Eigenschaften von StatefulSets	197
16.3.2	Manuell replizierte MongoDB mit StatefulSets	197
16.3.3	Das MongoDB-Cluster automatisch erstellen	200
16.3.4	Persistente Volumes und StatefulSets	203
16.3.5	Zum Abschluss: Readiness-Proben	204
16.4	Zusammenfassung	204
17	Kubernetes erweitern	205
17.1	Was bedeutet das Erweitern von Kubernetes?	205
17.2	Erweiterungspunkte	206
17.3	Patterns für Custom Resources	214
17.3.1	Just Data	214
17.3.2	Compiler	215
17.3.3	Operator	215
17.3.4	Der Einstieg	216
17.4	Zusammenfassung	216
18	Kubernetes über Programmiersprachen steuern	217
18.1	Die Kubernetes-API aus Sicht eines Clients	217
18.1.1	OpenAPI und generierte Client-Bibliotheken	218
18.1.2	Aber was ist mit kubectlx?	218
18.2	Mit der Kubernetes-API programmieren	219
18.2.1	Die Client-Bibliotheken installieren	219
18.2.2	Gegen die Kubernetes-API authentifizieren	220
18.2.3	Zugriff auf die Kubernetes-API	221
18.2.4	Führen wir die Einzelteile zusammen: Pods in Python, Java und .NET auflisten und erzeugen	222

18.2.5	Objekte erstellen und patchen	224
18.2.6	Kubernetes-APIs auf Änderungen belauschen	226
18.2.7	Mit Pods interagieren	228
18.3	Zusammenfassung	230
19	Anwendungen in Kubernetes absichern	231
19.1	SecurityContext verstehen	231
19.1.1	Herausforderungen beim SecurityContext	237
19.2	Pod Security	238
19.2.1	Was ist Pod Security	238
19.2.2	Pod-Security-Standards anwenden	239
19.3	Managen von Service-Accounts	242
19.4	Role-Based Access Control	242
19.5	RuntimeClass	243
19.6	NetworkPolicy	244
19.7	Service Mesh	248
19.8	Security-Benchmark-Tools	248
19.9	Image-Sicherheit	249
19.10	Zusammenfassung	250
20	Policy und Governance für Kubernetes-Cluster	251
20.1	Warum Policy und Governance wichtig sind	251
20.2	Genehmigungsablauf	252
20.3	Policy und Governance mit Gatekeeper	253
20.3.1	Was ist der Open Policy Agent?	254
20.3.2	Gatekeeper installieren	254
20.3.3	Policies konfigurieren	256
20.3.4	Constraint Templates verstehen	259
20.3.5	Constraints erstellen	259
20.3.6	Audit	260
20.3.7	Mutation	262
20.3.8	Datenreplikation	264
20.3.9	Metriken	266
20.3.10	Policy-Bibliothek	266
20.4	Zusammenfassung	266
21	Anwendungen auf mehrere Cluster deployen	267
21.1	Bevor Sie überhaupt anfangen	268
21.2	Ganz oben mit einem Load-Balancing-Ansatz beginnen	270

21.3	Anwendungen für mehrere Cluster bauen	271
21.3.1	Replizierte Silos: Das einfachste regions- übergreifende Modell	273
21.3.2	Sharding: Regionale Daten	274
21.3.3	Mehr Flexibilität: Microservice Routing	275
21.4	Zusammenfassung	276
22	Organisieren Sie Ihre Anwendung	277
22.1	Leitprinzipien	277
22.1.1	Dateisysteme als Source of Truth	277
22.1.2	Die Rolle des Code-Reviews	278
22.1.3	Feature Gates	279
22.2	Ihre Anwendung in der Versionsverwaltung managen	280
22.2.1	Struktur im Dateisystem	280
22.2.2	Regelmäßige Versionen managen	281
22.3	Ihre Anwendung für Entwicklung, Testen und Deployment strukturieren	283
22.3.1	Ziele	283
22.3.2	Verlauf eines Releases	284
22.4	Ihre Anwendung durch Templates parametrisieren	286
22.4.1	Mit Helm und Templates parametrisieren	286
22.4.2	Dateisystem-Layout zur Parametrisierung	287
22.5	Ihre Anwendung weltweit deployen	287
22.5.1	Architekturen für ein weltweites Deployment	288
22.5.2	Ein weltweites Deployment implementieren	289
22.5.3	Dashboards und Monitoring für weltweite Deployments	290
22.6	Zusammenfassung	291
A	Ein eigenes Kubernetes-Cluster bauen	293
A.1	Teileliste	293
A.2	Images flashen	294
A.3	Erstes Booten: Master	295
A.3.1	Das Netzwerk einrichten	295
A.3.2	Eine Container Runtime installieren	298
A.3.3	Kubernetes installieren	299
A.3.4	Das Cluster aufsetzen	299
A.4	Zusammenfassung	300
	Index	301