

Kubernetes

Das Praxisbuch für Entwickler und DevOps-Teams

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

1	Vorwort	11
1.1	Gliederung	13
1.1.1	Was Sie schon wissen sollten	14
1.1.2	Was Sie lernen werden	15
1.1.3	Wichtig zu wissen	16
1.2	Materialien zum Buch	18
2	Einführung in Kubernetes	19
2.1	Grundlagen und Konzepte: Warum überhaupt Container-Cluster?	20
2.1.1	Wieso überhaupt Container?	21
2.1.2	Warum brauchen Sie ein Container-Management-Tool?	25
2.1.3	Von Pets und Cattle	27
2.1.4	Stateless- und Stateful-Applikationen	28
2.1.5	Separation of Concerns	30
2.2	Kubernetes, das Tool der Wahl	34
2.2.1	Wieso Unternehmen auf Kubernetes setzen wollen	34
2.2.2	Versprechen von Kubernetes	37
2.2.3	Wichtige Features	42
2.2.4	Für welche Unternehmen ist Kubernetes sinnvoll	43
2.2.5	Für welche Unternehmen ist Kubernetes nicht sinnvoll	45
2.3	Architektur und Komponenten	46
2.3.1	Master Nodes	47
2.3.2	Worker Nodes	51
2.3.3	API Call Flow	52
2.4	Ein Kubernetes-Cluster auf Ihrem Computer	55
2.4.1	Minikube auf macOS	56
2.4.2	Minikube auf Linux	58
2.4.3	Minikube auf Windows	59
2.4.4	Minikube starten	60
2.4.5	Minikube steuern	61
2.4.6	Mögliche Fehler beim Starten von Minikube	61
2.4.7	Container Registry von Minikube	62

- 2.5 Interaktion mit Kubernetes über die Befehlszeile und das Dashboard 64**
 - 2.5.1 Minikube liefert kubectl mit 64
 - 2.5.2 kubectl installieren 64
 - 2.5.3 Mit Kubeconfig auf das Cluster zugreifen 68
 - 2.5.4 Namespaces 70
 - 2.5.5 kubectl-Befehle 72
 - 2.5.6 Ganz einfach Cluster und Namespaces wechseln 81
 - 2.5.7 Das Kubernetes Dashboard 82

- 2.6 Lens: Die IDE für Kubernetes 85**
 - 2.6.1 Überblick über Lens 86
 - 2.6.2 Vorteile gegenüber dem Kubernetes Dashboard 87
 - 2.6.3 Das Lens-Nachschlagewerk 91

- 2.7 Das Kubernetes-Cluster aus Raspberry Pis 94**
 - 2.7.1 Die richtigen Raspberry Pis auswählen 95
 - 2.7.2 Installation von Kubernetes 97
 - 2.7.3 Kubeconfig des Pi-Clusters nutzen 98

3 Grundlegende Objekte und Konzepte in Kubernetes 101

- 3.1 Pod- und Container-Management 104**
 - 3.1.1 Container Engines 107
 - 3.1.2 Ihr erster eigener Pod 110
 - 3.1.3 Mehrere Container innerhalb eines Pods 113
 - 3.1.4 Kommunikation zwischen den Containern 116
 - 3.1.5 Init-Container 117
 - 3.1.6 Pod-Phasen und Container-Status 120
 - 3.1.7 Die Restart Policy von Pods 122
 - 3.1.8 Wenn es mit dem Pod zu Ende geht 123

- 3.2 Annotations und Labels 126**
 - 3.2.1 Labels und Selektoren anwenden 127
 - 3.2.2 Field-Selektoren 130
 - 3.2.3 NodeSelector 130
 - 3.2.4 Node Affinity und Anti-Affinity 133
 - 3.2.5 Pod Affinity und Anti-Affinity 137
 - 3.2.6 Taints and Tolerations 142
 - 3.2.7 Annotations 145

3.3	Deployments und ReplicaSets	147
3.3.1	Die Rolle von ReplicaSets	149
3.3.2	Erstellen von Deployments	152
3.3.3	Rolling Updates durch das Deployment-Objekt	154
3.3.4	Rollback mithilfe des Deployments	160
3.4	ConfigMaps und Secrets	163
3.4.1	Was sind ConfigMaps?	164
3.4.2	Was sind Secrets?	173
3.5	Kommunikation mit Services und Ingress etablieren	183
3.5.1	Kommunikation zwischen Pods	185
3.5.2	Kommunikation über einen Service	185
3.5.3	Kommunikation über einen Ingress	192

4 Everything as Code: Tools und Prinzipien für Kubernetes Operations 199

4.1	Deklarative Konfigurationen	201
4.2	YAML: Die Sprache für Kubernetes	204
4.2.1	Grundlagen der YAML-Syntax	205
4.2.2	Datentypen in YAML	207
4.2.3	Anker und Aliase	209
4.2.4	Einzeilige YAML-Schreibweise in Dokumentationen	210
4.2.5	Schwächen von YAML	211
4.2.6	Tipps für die Praxis	212
4.3	Versionsverwaltung von Kubernetes-Manifesten	213
4.3.1	Mit Git arbeiten	215
4.3.2	Viele Kubernetes-Manifeste managen	217
4.3.3	Branching-Strategien	221
4.3.4	Aufteilung der Repositories	225
4.4	CI und CD	227
4.4.1	Pipeline-Schritte für Kubernetes	228
4.4.2	Pipeline-Architekturen	233
4.4.3	GitOps	237
4.5	Templating mit Kustomize	239
4.5.1	Grundlagen von Kustomize	241
4.5.2	Ressourcen-Generator	245
4.5.3	Weitere Kustomize-Built-ins	247
4.5.4	Fazit zu Kustomize	248

5 Fortgeschrittene Objekte und Konzepte in Kubernetes 251

- 5.1 DaemonSets** 252
- 5.2 Jobs in Kubernetes** 255
 - 5.2.1 Kubernetes-Jobs in der Praxis 256
 - 5.2.2 Queue Worker mit RabbitMQ 258
 - 5.2.3 Kubernetes-CronJobs 263
- 5.3 Custom Resources und Custom Resource Definitions** 265
 - 5.3.1 Beispiel: Eine Monitoring-CR 266
 - 5.3.2 Validierung in CRD 269
 - 5.3.3 Operatoren 272
- 5.4 Downward-API** 275
- 5.5 Pod Priority und Preemption** 278
- 5.6 Versionierung von Objekten in Kubernetes** 280

6 Stateful-Anwendungen und Storage 283

- 6.1 Stateful-Anwendungen in Kubernetes durch StatefulSets** 285
 - 6.1.1 Pod Management Policy 287
 - 6.1.2 Strategien für Updates 289
 - 6.1.3 Retention Policy für PVCs 291
- 6.2 Persistent Volumes und Persistent Volume Claims** 292
 - 6.2.1 Storage-Typen für PVs 297
 - 6.2.2 CSI-Treiber für externe Speichermedien 301
 - 6.2.3 Storage-Klassen und dynamische PVs 303
 - 6.2.4 PostgreSQL als StatefulSet mit Persistent Volume 306
- 6.3 Ephemeral Volumes** 309
- 6.4 Weitere Features von Volumes** 312
 - 6.4.1 Volume Snapshots 312
 - 6.4.2 Projected Volumes 316

7	Kubernetes-Governance und -Security: Prepare for Production	319
7.1	Pod Security	321
7.2	Pod Security Admission	325
7.3	Admission Controller	326
7.4	Kubernetes-Policies	329
7.5	Policy-Objekte	332
7.6	RBAC für die Zugriffskontrolle	334
7.6.1	Subjects: User, Gruppen und Service Accounts	336
7.6.2	Roles und Role Bindings	338
7.6.3	Fazit	340
8	Anwendungen für Kubernetes entwickeln: Ready for Production	343
8.1	Pod-Ressourcen managen	344
8.2	Readiness, Liveness und Startup Probes	348
8.2.1	Wie Sie Probes definieren	350
8.2.2	Probes an einem Beispiel testen	353
8.3	Skalierung und Lastausgleich	358
8.3.1	Horizontal Pod Autoscaling	359
8.3.2	Vertical Pod Autoscaling	363
8.3.3	Cluster Autoscaler	365
8.4	Monitoring	366
8.4.1	Einführung: Prometheus, Grafana und Alertmanager	367
8.4.2	Monitoring auf dem Pi-Cluster	370
9	Kubernetes mit Helm orchestrieren	381
9.1	Helm, der Kubernetes-Paketmanager	382
9.1.1	Erstes Helm-Chart erstellen	385
9.1.2	Ein Helm-Chart über die CLI deployen	386
9.1.3	Ein Helm-Repository einrichten und verwalten	388

- 9.1.4 Ein Helm-Chart mit Lens deployen 390
- 9.1.5 Helm-Releases aktualisieren und löschen 391
- 9.1.6 Helm-Charts aus einem Repository herunterladen 393
- 9.2 Helm-Charts lesen und entwickeln 393**
 - 9.2.1 Die Templating-Engine und die Sprache der Charts 394
 - 9.2.2 Charts mit Values konfigurieren 399
 - 9.2.3 Bedingungen in Helm-Templates 404
 - 9.2.4 Weitere Operationen und Kontrollstrukturen 406
 - 9.2.5 Helm Diff zum Prüfen von Änderungen 409
- 9.3 Eigene Charts entwickeln 411**
 - 9.3.1 Der Rahmen Ihres Helm-Charts 412
 - 9.3.2 Charts verpacken und in Repository ablegen 414
 - 9.3.3 Abhängigkeiten in Helm-Charts verwalten 417
- 9.4 Fazit 421**

- Index 423