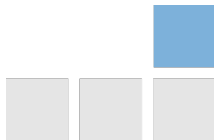


Intern: Kubernetes

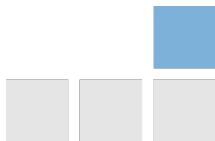
Production-Grade Container Orchestration



Felix Greb ■ 10.12.2018

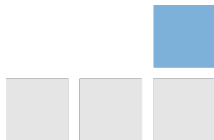


- Allgemeines
- Features
- Aufbau und Komponenten
- Kubernetes Objekte
- Kubernetes Cluster bei Tralios IT GmbH





- Kubernetes auch bekannt als K8s wurde ursprünglich von Google entwickelt, inzwischen von der Cloud Native Computing Foundation
- Version 1.0 wurde am 21. Juli 2015 veröffentlicht
- Release Cycle: Alle 3 Monate
- Bekannteste Services die Kubernetes anbieten: AWS (EKS), Microsoft Azure, Google (GKE), IBM Cloud, Alibaba Cloud





Google Trends für Kubernetes

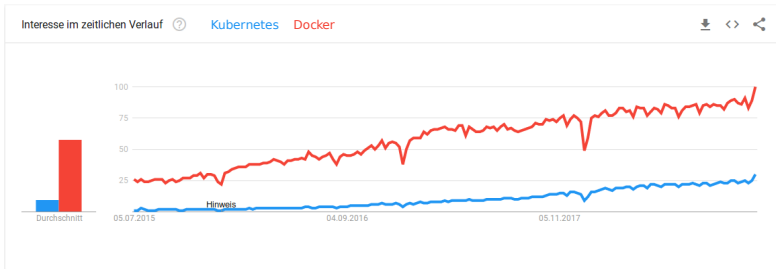


Abbildung: <https://trends.google.com>

Features

- Automatisiertes Deployment, Skalierung, Management
- Service discovery and load balancing
- Storage orchestration
- High Availability durch automatische Neustarts und Rescheduling
- Seit Version 1.8 mit Role Based Access Control (RBAC)

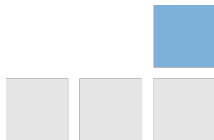


Master Komponenten - Control Plane

- kube-apiserver
Stellt API Funktionen für alle anderen Dienste zur Verfügung, z.B.: für den Zugriff per kubectl
- etcd
Persistenter Storage in welchem alle Informationen zum Cluster gespeichert sind
- kube-scheduler
Entscheidet auf welchem Knoten ein neuer Pod laufen wird
- kube-controller-manager
Legt Objekte, die in Etcd definiert sind, an bzw. gibt den Befehl zum löschen

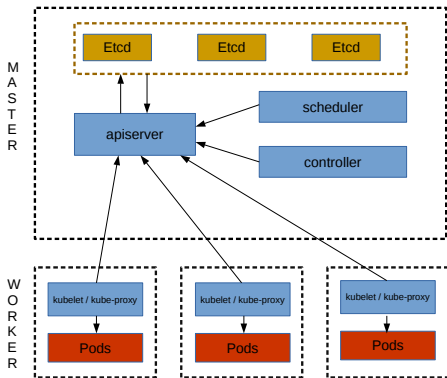


- kubelet
Gibt den aktuellen Knoten Status an die API weiter, startet und beendet Pods
- kube-proxy
Verbindung zwischen Service und User, kümmert sich um Forwarding
- Container Runtime
Die verwendete Container Runtime, in vielen Fällen Docker, es sind aber auch andere, wie beispielsweise rkt oder runc möglich

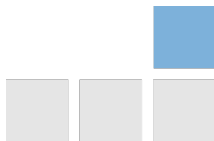




Komponenten Übersicht



- DNS
Kubernetes liefert einen internen DNS mit
- Web UI (Dashboard)
Ermöglicht eine schnelle Übersicht über den Cluster, einzelne Namespaces und Probleme
- Overlay Network
Wir setzen hierfür Flannel ein, andere Möglichkeiten sind z.B.: Calico, Weave oder Romana





- Pod

Die kleinste Objekt Einheit, ein Pod beinhaltet mindestens einen Container

- Service

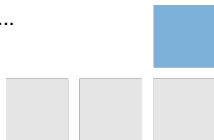
Verknüpfung zwischen dem internen Pod und einem externen Zugriff

- Volume

Speicher für Pods, kann auch ein Persistent Volume sein

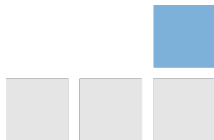
- Namespace

Voneinander abgetrennte Bereiche mit eigenen Pods, Services, ...





- ReplicaSet
Definiert wie oft ein Pod repliziert werden soll, meist durch Deployments genutzt
- Deployment
Managed Pods und dazugehörige Replicasets
- StatefulSet
Für Stateful Applications, in der Regel mit persistentem Speicher und einzigartigem definierten Netzwerk



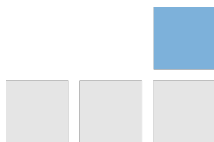


- Daemonset

Definiert Pods, welche auf jedem Knoten laufen, automatisiert up- / down-Scaling bei Knoten Änderungen

- Job

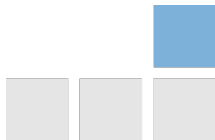
Erstellt einen Pod und wartet bis dieser seine Aufgabe erfolgreich abgeschlossen hat





Beispiel YAML File für ein Deployment

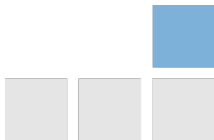
Siehe grafana.yaml



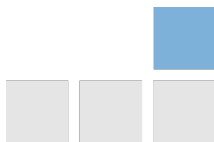


Extras in unserem Cluster

- Rook
Stellt Ceph Block Storage im Cluster zur Verfügung
- MetalLB
Load Balancer um automatisiert IP Adressen bestimmten Services zuzuweisen
- Traefik
Reverse Proxy und Load Balancer
- Prometheus und Grafana
Monitoring für Kubernetes Ressourcen
- Istio (zumindestens in der Zukunft)
Mächtiges Tool zur Datenaggregation



**Verwendung von kubectl, Kubernetes Dashboard, Demo eines Deployments,
Skalierung und Ausfall eines Knotens**



- Logo: <https://github.com/kubernetes/kubernetes/blob/master/logo/logo.pdf>
- Inhalt: <https://kubernetes.io/docs/>