

Kubernetes aus Admin-Sicht

fnorsch



15. Januar 2020

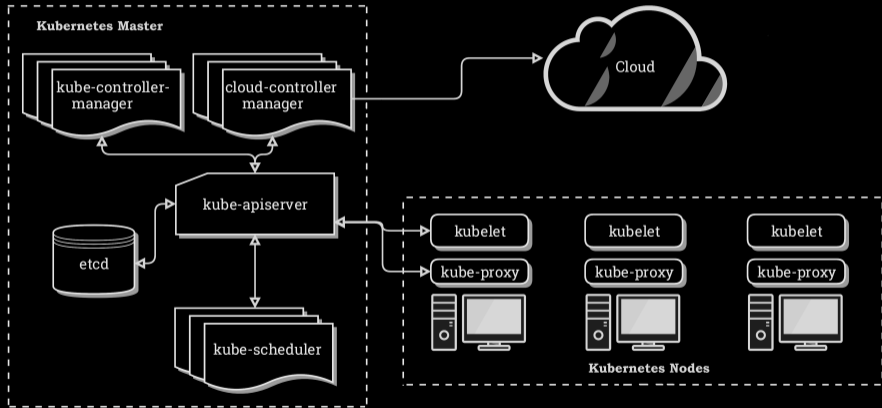
Eine Entwicklerin, ein Administrator und ein GoogleBot/SRE gehen in eine Bar ...

Einleitung

Abgrenzung

- Grober Überblick
- Keine Anleitung, keine Empfehlungen
- Kein Experte für Alles

Architektur



Architektur

Grundsätzliche Objekte

- Pods
- Namespace
- Volume
- Service

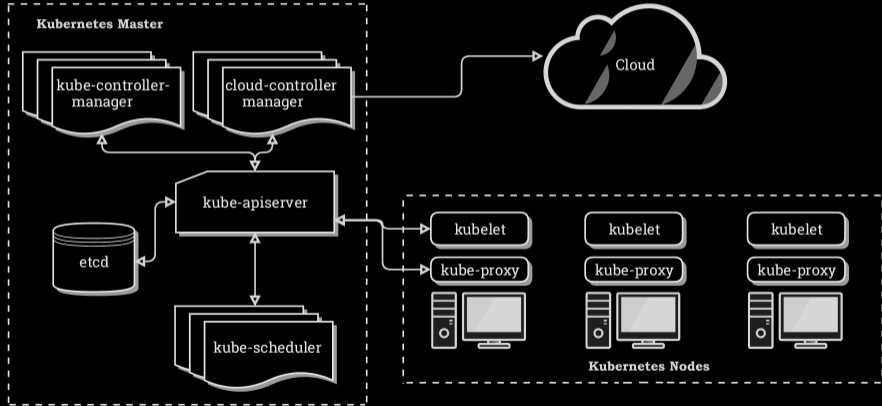
Architektur

Weiteres

- Controller-Objekte (Deployments, Ingress, ...)
- Custom Resource Descriptors (Operators)
- kustomize (und helm ...)

Architektur

Unvollstaendig!



Architektur

Wichtige Infrastruktur

- DNS — extern
- Virtuelle IPs
- LoadBalancer (Layer4++)
- CloudDNS
- Ingress
- Container Storage Interface
- Cluster Networking
- Service Meshes, Logging, Metrics
- (Registry)

- Unterliegende Komponenten
- Konzepte der Grundlagen
- Fiese Details — Kernel vs. libc
- Fehlersuche — Verteilt und über alle Ebenen

Kritik

Allgemeines

- Cloudzentrisch
- Komplex!
- Fehlersuche
- Veröffentlichungspolitik
- Ungetestete Codeverzweigungen

Kritik

Bullshit

Beispiel 1

“With the XXX, organizations can ship software with greater agility. XXX gives platform engineers and operators a self-service edge proxy, reducing toil. At the same time, XXX gives application developers fine-grained control over their edge proxy needs. Built on Kubernetes and XXX Proxy, XXX provides state-of-the-art functionality for observability, resilience, and traffic management.”

Beispiel 2

“Everybody loves using containers. Nobody loves deploying, monitoring, and managing them — especially across multiple public and private clouds. There are simply too many tools and too little standardization. XXX automates the repetitive things and simplifies the complex ones so everyone can just go back to enjoying the magic of containers.”

Kritik

Bullshit

Beispiel 1

“With the XXX, organizations can ship software with greater agility. XXX gives platform engineers and operators a self-service edge proxy, reducing toil. At the same time, XXX gives application developers fine-grained control over their edge proxy needs. Built on Kubernetes and XXX Proxy, XXX provides state-of-the-art functionality for observability, resilience, and traffic management.”

Beispiel 2

“Everybody loves using containers. Nobody loves deploying, monitoring, and managing them — especially across multiple public and private clouds. There are simply too many tools and too little standardization. XXX automates the repetitive things and simplifies the complex ones so everyone can just go back to enjoying the magic of containers.”

Distributionen

- **Unübersichtlicher Markt**
 - Entwickler-Tools (minikube, minishift, k3s, ...)
 - Cloud-Distributionen (GKE, EKS, AKS, ...)
 - Unverändert (rke, kubeadm, ...)
 - und ...

Distributionen

- Unübersichtlicher Markt
- Entwickler-Tools (minikube, minishift, k3s, ...)
- Cloud-Distributionen (GKE, EKS, AKS, ...)
- Unverändert (rke, kubeadm, ...)
- und ...

Distributionen

- Unübersichtlicher Markt
- Entwickler-Tools (minikube, minishift, k3s, ...)
- Cloud-Distributionen (GKE, EKS, AKS, ...)
- Unverändert (rke, kubeadm, ...)
- und ...

Distributionen

- Unübersichtlicher Markt
- Entwickler-Tools (minikube, minishift, k3s, ...)
- Cloud-Distributionen (GKE, EKS, AKS, ...)
- Unverändert (rke, kubeadm, ...)
- und ...

Distributionen

- Unübersichtlicher Markt
- Entwickler-Tools (minikube, minishift, k3s, ...)
- Cloud-Distributionen (GKE, EKS, AKS, ...)
- Unverändert (rke, kubeadm, ...)
- und ...

Distributionen

RedHat OpenShift bzw. okd.io

- *Rolls-Royce* Lösung
- Viele Extras — Multi-Tenant
- okd.io nur mässig brauchbar
- Übernahme von CoreOS
- LTS erhältlich ...

Betrieb

Grundlegendes

- Umgang mit k8s selbst
- Cluster-Admin nicht rausgeben!
- Registry White-Listing
- Monitoring: nur über API oder Container
- Integration in Umgebungen (TLS, External Services, ...)
- Garantien bzgl. Verfügbarkeit

- Docker ist out!
- CoreOS, Atomic Host, RancherOS — Eher kein traditionelles Linux
- Registry mit Security Scan ist wichtig für Drittanbietercontainer
- Wenn möglich, Container immer selbst bauen (Letzte Pakete)
- podman, buildah verwenden zum Bau von Containern

Betrieb

Kubernetes

- Sauber arbeiten — Kein Rollback
- Cluster-Updates konservativ angehen oder mehrere haben

EOT

Danke fuer die
Aufmerksamkeit

Passendes Musikvideo

Musik!