

Cloud Application Stacks mit Kubernetes

Roadmap

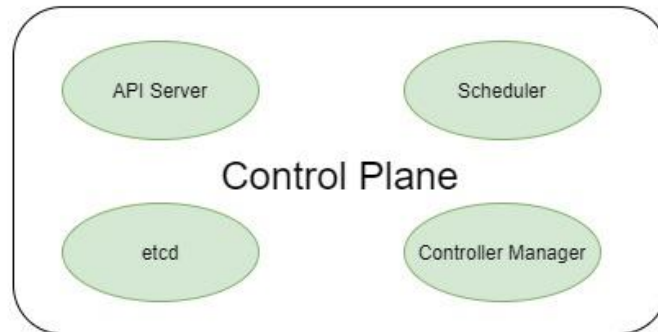
1. Anforderungen
2. Infrastruktur
3. Softwarekomponenten
4. Ablauf des Deployments (Demo)
5. Herausforderungen & Learnings
6. Fazit

Anforderungen

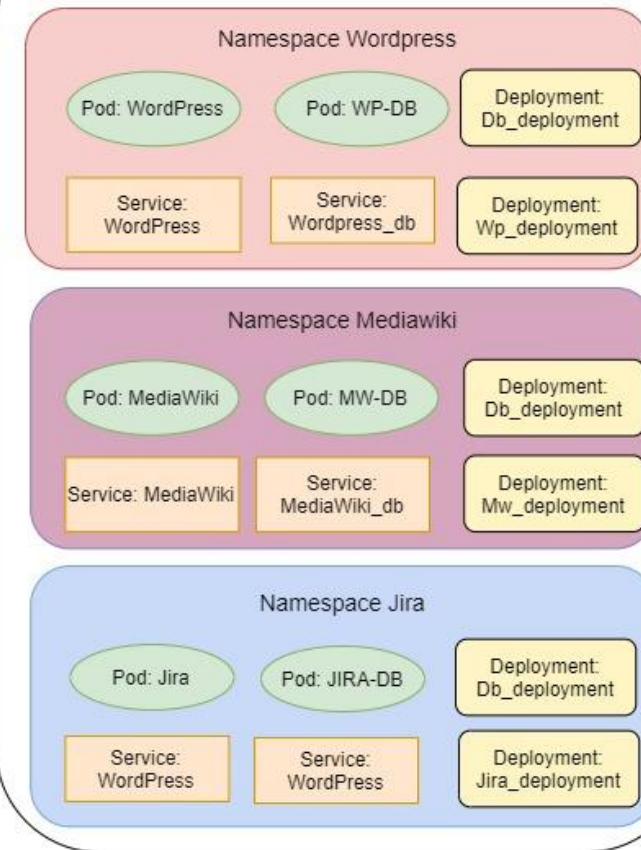
- Bereitstellung einer lauffähigen Umgebung für
 - WordPress
 - MediaWiki
 - Jira
- Implementierung in einer Cloud Umgebung mittels Docker & Kubernetes
- Nutzung von **Minikube** für die lokale Entwicklung
- Automatisierte Installation & Konfiguration über YAML Files

Infrastruktur

Minikube Cluster



Worker Node



Softwarekomponenten



Docker

Minikube

Kubectl
(Kubernetes)

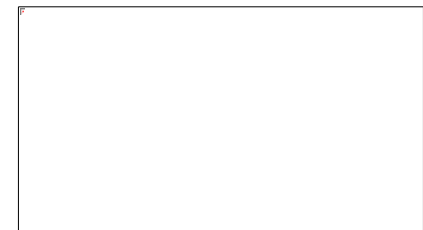
Docker

Rolle von Docker in unserem Projekt:

- Auf dem lokalen Rechner lief Docker als Grundlage für die Container Orchestrierung
- Innerhalb von Docker wurde ein **Minikube Container** erstellt, der das gesamte KubernetesCluster simulierte
- In diesem Minikube Container sind die **Docker Images** der Applikationen sowie der Datenbanken

Aufbau des Systems:

- 1.Lokale Docker Installation: Host für den Minikube Container
- 2.Minikube Container: Kubernetes Umgebung innerhalb von Docker
- 3.Docker in Minikube: Enthält die einzelnen Applikations und Datenbank Images.



Minikube

Rolle von Minikube in unserem Projekt:

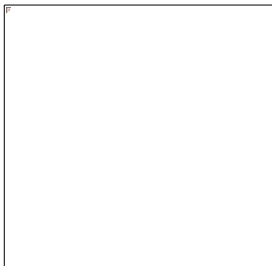
- Minikube diene als Entwicklungsumgebung, um ein Kubernetes Cluster lokal zu simulieren

Warum Minikube?

- Keine Abhängigkeit von externen Cloud-Diensten

Herausforderung:

- Die Konfiguration von Persistent Volumes in Minikube war anfangs komplex, wurde jedoch durch genaue YAML-Definitionen gelöst.



Kubectl

Rolle von Kubectl in unserem Projekt:

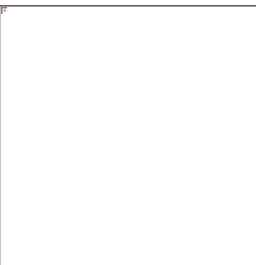
- Kubectl war wichtig für uns, um Kubernetes Ressourcen zu verwalten
- Wir nutzten es für das Erstellen, Testen und Überwachen von Deployments und Pods

Vorteil von Kubectl:

- Es bietet eine schnelle Möglichkeit, Fehler in Kubernetes-Ressourcen zu identifizieren und zu beheben.

Beispielbefehle:

- *kubectl apply -f deployment.yaml* – Startet die in YAML definierten Ressourcen
- *kubectl get pods* – Überprüft den Status der laufenden Pods
- *kubectl describe pod <pod-name>* – Liefert detaillierte Informationen zu einem bestimmten Pod



Ablauf des Deployments

DEMO

Herausforderungen und Learnings

Herausforderungen

- Dass die Datenbank Services und die Applikations Services reibungslos miteinander kommunizieren können
- Fehlersuche & Debugging von Kubernetes Deployments

Learnings

- Besseres Verständnis für Kubernetes & Minikube
- Praktische Erfahrung mit Container Orchestrierung



Q&A

Fragen?