

Containerbasiertes PaaS für individuelle VPS Infrastrukturen

Studenten



Stefan Meyer



Jan Meier

Ausgangslage: Die zunehmende Komplexität von Software-Deployments auf eigenen Infrastrukturen stellt Entwickler vor Herausforderungen. Während Platform-as-a-Service (PaaS)-Lösungen diese Komplexität reduzieren können, sind kommerzielle Cloud-PaaS-Angebote wie Azure App Service, AWS Beanstalk oder Digital Ocean App Platform oft kostenintensiv. Gleichzeitig mangelt es vielen Softwareentwicklern an fundierten Kenntnissen in der Serveradministration, um eine eigene Lösung auf einem Virtual Private Server (VPS) aufzusetzen. Ziel dieser Arbeit ist die Entwicklung einer kostengünstigen und einfach zu bedienenden Self-Hosted PaaS-Lösung für VPS-Infrastrukturen, um Build- und Deployment-Prozesse sowie die Orchestrierung von laufenden Container-Anwendungen zu vereinfachen. Darüber hinaus soll das Tool in einem Cluster aus mehreren VPS betrieben werden können, um auf unterschiedliche Skalierungsanforderungen reagieren zu können.

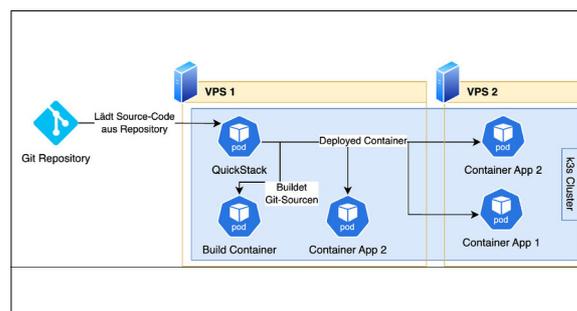
Vorgehen / Technologien: Eine Marktanalyse bestehender PaaS-Lösungen und VPS-Anbieter wurde durchgeführt, um Kosten und Funktionsumfang zu vergleichen. Anschliessend wurden verschiedene Container-Orchestrierungstools evaluiert. Die Wahl fiel aufgrund des geringen Ressourcenbedarfs, der einfachen Installation und der integrierten Komponenten wie Traefik, Ingress und ServiceLB auf die Kubernetes-Distribution k3s. Als Shared Storage Provider für persistente Daten wurde Longhorn basierend auf den mit Fio durchgeführten Performancetests und der einfacheren Inbetriebnahme ausgewählt. QuickStack selbst wurde mit dem Fullstack-Framework Next.js, SQLite als Datenbank und verschiedenen JavaScript-Bibliotheken entwickelt. Die Architektur orientiert sich an Clean Code Prinzipien und verwendet Kubernetes und Longhorn APIs. Kaniko wird als Container Build Tool und Registry als Container Registry verwendet. Wie in der Grafik dargestellt (Abb. Architektur QuickStack), wird der Quellcode aus dem Git Repository geladen und dem Build Container zur Verfügung gestellt. Nach dem Build wird der Container in den k3s Cluster deployed und kann über den QuickStack Pod orchestriert werden. Die Container der Applikation werden je nach Konfiguration auf die verschiedenen VPS-Nodes des k3s Clusters verteilt.

Ergebnis: QuickStack bietet eine benutzerfreundliche Weboberfläche zur Verwaltung von Anwendungen und deren Deployments auf einem k3s-Cluster. Kernfunktionen sind die einfache Installation per Kommandozeile, Deployments aus Git-Repositories, Ressourcenüberwachung, SSL-Zertifikatsverwaltung, Live-Log-Streaming, Webterminal-Zugriff, Zwei-Faktor-Authentifizierung und die flexible Anbindung von persistenten Volumes. Damit vereinfacht QuickStack das Applikationsmanagement auf

VPS-Infrastrukturen erheblich und bietet eine kostengünstige Alternative zu kommerziellen PaaS-Lösungen. Usability-Tests zeigten eine hohe Benutzerfreundlichkeit. Zukünftig könnte QuickStack um automatisierte Backups, weitere Container-Debugging-Tools und Deployment-Templates für Datenbanken und gängige Open-Source-Anwendungen erweitert werden.

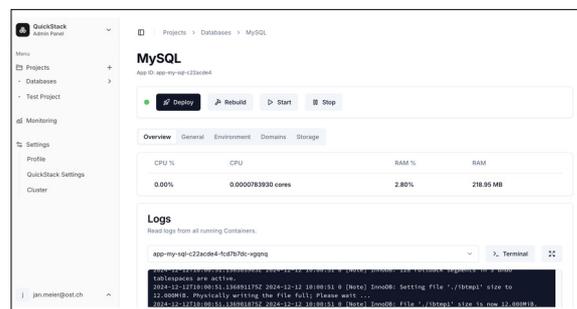
Architektur QuickStack (vereinfacht)

Eigene Darstellung



QuickStack UI

Eigene Darstellung



Referenten

Prof. Mirko Stocker,
Dominic Klinger

Themengebiet

System-Software,
Software, Networks,
Security & Cloud
Infrastructure