

Unsere
IT-Lösungen werden
Sie begeistern!

Motivation für Kubernetes on-premise Cluster



Die Motivation, die Bereitstellung einer Container-Plattform und eines Kubernetes-Clusters vor Ort in Betracht zu ziehen, ist die Implementierung neuer Unternehmensanwendungen unter Verwendung des „CloudReady“-Microservices-Konzepts. Das Ziel ist dann nicht nur, die Migration von Anwendungen in die öffentliche Cloud zu ermöglichen, sondern auch den Lastausgleich der internen Infrastruktur zu automatisieren und eine Reihe von Prozessen für die Bereitstellung neuer Versionen von Anwendungen zu automatisieren.

Kurz zu Kubernetes

Kubernetes ist ein offenes System zur Orchestrierung der Virtualisierung auf der Ebene des Betriebssystems. Es wurde ursprünglich von Google entwickelt und unterstützt z. B. Docker als untergeordnetes Tool. Wir können auch von virtueller Containerisierung auf Betriebssystemebene sprechen.



Die Vorteile von Kubernetes sind wie folgt:

1 ORCHESTRATION

- Möglichkeit der automatischen Skalierung je nach Last.
 - Breite Skalierungsoptionen durch Hinzufügen zusätzlicher
 - Worker-Knoten zum Kubernetes-Cluster.
 - HA-Konfiguration - bei Ausfall eines Worker-Nodes automatische Pod-Migration (d.h. Container in Kubernetes) auf einen anderen Worker-Node.
 - Health-Check und Selbstheilung - Orchestrate ermöglicht die Einrichtung von Health-Check-Mechanismen für den Start und Betrieb.
 - Traffic-Routing und Lastausgleich - wir definieren die Anwendung als Service neu; sie wird mit Kubernetes-Orchestrierung bereitgestellt. Als Teil der
- Dienstdefinition legen die Anzahl der Replikat fest. Die Kommunikation mit den Diensten erfolgt über die Definition sogenannter Ingresses – also Eingangspunkte zum Dienst. Kubernetes leitet dann Anfragen innerhalb seines Netzwerks weiter.
 - Automatisierte Roll-outs und Roll-backs, bei denen einzelne Pods, die die Anwendung tragen, schrittweise „aufgerüstet“ werden können, so dass keine Ausfallzeiten für den Benutzer entstehen.
 - Canary Deployments – die Möglichkeit, eine neue Version eines Produktionssystems neben der ursprünglichen Version zu testen, und dann einfach zu wechseln.
 - Die Fähigkeit, jeden Dienst mit Hilfe der Kubernetes-Orchestrierung schnell zu erstellen und wieder zu entfernen.

— Automatisierte Bereitstellung über die CI/CD-Pipeline - es gibt CI/CD-Tools, die sich direkt mit der Kubernetes-API verbinden und mithilfe der Orchestrierungskonfigurationsdateien der neuen Version der Docker-Images bereitstellen können.

— Umwandlung der Anwendung in einen „Dienst“.

— Vorbereitung auf die spätere Migration zu ausgelagerten Cloud-Diensten (AWS, MS Azure, Google Cloud).

Kubernetes-Cluster-Konzept on-premise von GEM System



2 IMPLEMENTIERUNG EINES KUBERNETES-CLUSTERS KUBERNETES DER CLUSTER KANN AUF VERSCHIEDENE ARTEN IMPLEMENTIERT WERDEN:

- Kubernetes in der Cloud (MS Azure, AWS, Google Cloud und andere) - wir kümmern uns nicht um die Verwaltung der Plattform, aber wir zahlen dafür.
- Einfaches Kubernetes auf Bare-Metal ist eine saubere Installation, Docker Engine hingegen ist eine ziemlich anspruchsvolle Implementierung der Plattform und in der Praxis auch weniger einfach zu verwalten.
- Vorgefertigte Distributionen mit Kubernetes und unterstützender Infrastruktur (Red Hat OpenShift, Rancher, Kubespray, MetalK8s ...) – in der Größenordnung von einfacher zu implementieren und zu verwalten, aber nur Zahlungen für einige der aufgeführten Distributionen.
- Es gibt auch Mini-Distributionen, die 100% Erfüllung Kubernetes Funktionalität sind, aber zur gleichen Zeit kann mit minimalen Anforderungen für HW ausgeführt werden - geeignet für Entwickler, wenn sie auf ihren Workstations / Laptops entwickeln wollen - z.B. minikube.



3 TYPISCHE ARCHITEKTURKUBERNETES-CLUSTER

- **Front-Reverse-Proxy** (bei Verwendung von No-depot) oder für den Betrieb in der DMZ -> leitet Anfragen aus dem Internet/Intranet an den Kubernetes-Cluster weiter. Sie können einen beliebigen Software-Proxy (Apache HTTPServer, Nginx, HAProxy) oder ein dediziertes Netzwerkelement (F5 Networks, Cisco, usw.) verwenden.
- **Kubernetes-Cluster** – Master-Server, auf denen die grundlegenden Kubernetes-Dienste laufen (kubeproxy, etcd-schlüssel/wert-Datenbank für die Kubernetes-Konfiguration).
- **Kubernetes-Cluster** – Worker-Nodes, auf diesen Servern laufen einzelne Workloads, d. h. einzelne Pods mit Anwendungen.
- **Speicher** – entweder Hardwarespeicher, der mit Kubernetes über APIs zusammenarbeitet, oder softwaredefinierter Speicher (NFS, GlusterFS) und ein mögliches Speicher-Backend (Platten-Array).
- **Hilfssysteme** – Überwachung und Logsammlung - Prometheus, Elastic Stack und andere.
- **CI/CD-Plattform** – Atlassian-Tools, GitLab, Stack und andere.

