

Container Orchestration

Container automatisiert verwalten und koordinieren

Container haben Art und Weise der Verteilung und des Betriebs von Anwendungen revolutioniert. Die Komplexität ist aber nicht verschwunden, sondern auf die Ebene der Cluster gewandert. Container Orchestration nimmt sich ihr dort an und versucht sie, auf wenige Konfigurationseinstellungen zu reduzieren.

Definition

Container fassen unterschiedliche Software-Artefakte und ihre Abhängigkeiten zu einer standardisierten Einheit zusammen. So entstehen vollständig isolierte Anwendungsplattformen. Das führt zu einem deutlich vereinfachten Verteilungsprozess der Anwendungen. Gleichzeitig eröffnet es die Chance, Anwendungen skalierbar und hochverfügbar bereitzustellen. Container Orchestration bedeutet, alle Aspekte der Verwaltung und Koordination solcher Container ebenfalls zu vereinheitlichen und zu automatisieren.

Das erreicht Container Orchestration durch mehrere Maßnahmen. Zunächst führt es eine **Kontrollebene** (engl. control plane) ein, die aus einem API-Server, einem Scheduler und einem Manager besteht. Die Kontrollebene liest den in einer Konfigurationsdatei deklarierten, erforderlichen oder gewünschten Zielzustand ein. Über der Kontrollebene liegt die **Verarbeitungsebene** (engl. compute plane). Sie setzt sich aus einer beliebigen Anzahl von Servern respektive Nodes zusammen, die wiederum eine beliebige Anzahl Container und damit Anwendungen ausführen. Die Anzahl der Nodes und Container hängt direkt von der bereitgestellten Konfiguration ab.

Die Container-Abbilder stehen in einer **Container Image Registry** bereit, einem Repository, in dem die Abbilder versio-

Cloud Computing

- Container-as-a-Service
- Platform-as-a-Service
- Elastizität
- Verfügbarkeit
- Dezentralisierung

Containerisierung

- Kapselung
- Wiederverwendbarkeit
- Abhängigkeiten reduzieren

niert hinterlegt sind. Zudem ist eine von den Containern und Ebenen losgelöste **Persistenz** wesentlich, um anfallende oder benötigte Daten langfristig speichern zu können.

Die Kommunikation zwischen den Nodes und dem Client erfolgt letztendlich über einen **Proxy**, der verschiedene Aufgaben wie Lastverteilung und Verbindungssicherung übernimmt.

Infrastructure as Code

- automatisieren
- versionieren
- nachvollziehbar
- revisionsssicher

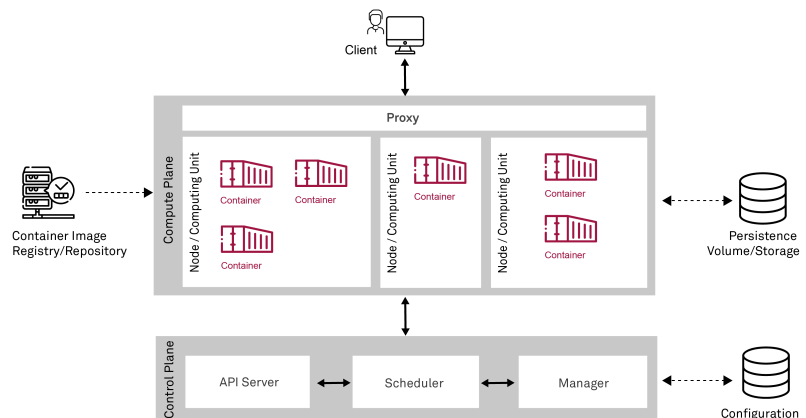
Cloud-Native Architecture

- Microservices
- Skalierbarkeit

COR

Referenzszenario

Eine oder mehrere Anwendungen sollen in Container verpackt hochverfügbar und hochskalierbar bereitgestellt werden. Entscheidend ist, dass jederzeit die passende Anzahl von Anwendungen und Instanzen läuft. Die vorhandenen Ressourcen sollen oder müssen so effizient wie möglich genutzt werden, bei entsprechendem Bedarf aber jederzeit ergänzt werden können.



Das Unternehmen setzt dazu ein automatisiertes Deployment mitsamt aller Schnittstellen, Quellen und weiterer unterstützender Infrastruktur auf.

Sobald diese Basis geschaffen ist, entscheidet fortan die Orchestrierung selbstständig, auf welchen Nodes der Berechnungsebene die Container gestartet und ausgeführt werden, um eine optimale Lastverteilung zu erreichen. Sie fügt Ressourcen bedarfsgerecht hinzu oder entfernt sie, um eine optimale Balance zwischen angeforderter Dienstleistung und bereitgestellten Mitteln zu erreichen.

Potenzial

Container Orchestration verschiebt die Verantwortung für den Betrieb von Anwendungen noch stärker in Richtung Entwicklung, als es Container allgemein bereits tun. Das verschafft dem Unternehmen eine deutlich höhere Flexibilität und Dynamik durch immer umfassendere Automatisierung.

Reifegrad

Einige Anbieter haben bereits Standards und Schnittstellen definiert, die sich teilweise noch in Beta- oder gar Alpha-Stadien befinden. Mitunter gibt es auch konkurrierende Lösungen. Mittelfristig wird es mit wachsender Praxiserfahrung zu einer Konsolidierung des Marktes und der Lösungen kommen. Einem produktiven und stabilen Betrieb steht prinzipiell nichts im Wege, solange es akzeptabel

ist und vorerst bleibt, dass Teile der APIs noch in Bewegung sind.

Marktübersicht

Kubernetes dominiert den Markt. Alle großen Infrastrukturanbieter wie Amazon, Microsoft oder Google setzen auf Kubernetes. Sie entwickeln das Produkt teilweise aktiv weiter und heben die Relevanz von Kubernetes durch Dienstbezeichnungen wie Azure Kubernetes Service, Elastic Kubernetes Service oder Cloud Kubernetes deutlich hervor. Demgegenüber steht eine kleinere Fraktion, die nicht auf Kubernetes, sondern Docker Swarm oder Nomad setzt.

Eine Vielzahl auf Kubernetes zugeschnittener Lösungen erweitert dessen Ökosystem und steigert seine Beliebtheit. Für die Verwaltung kommen etwa Kubernetes oder Kublr zum Einsatz, Monitoring und Logging übernehmen Produkte wie Grafana oder Dynatrace, und Haproxy oder Nginx kümmern sich ums Load Balancing. Wiederum andere Lösungen erhöhen die Sicherheit, etwa Yubico oder Blackduck, oder unterstützen allgemein die Entwicklung von auf Kubernetes ausgerichteten Anwendungen.

Alternativen

Anstelle Anwendungen in Container zu verpacken, lassen sich auch klassische Virtualisierungslösungen einsetzen, meistens auf einem Unterbau aus Infra-

structure-as-a-Service (IaaS). Diese Lösungen sind weniger flexibel und zudem ressourcenhungriger, schließlich beinhalten sie nicht nur eine Laufzeitumgebung, sondern auch ein vollwertiges Betriebssystem.

Anstatt die verfügbaren Ressourcen durch die ideale Verteilung der Container maximal auszureizen, lässt sich ebenso auf vertikale Skalierung setzen. Das bedeutet schlicht, zusätzliche oder leistungsfähigere Hardware zu installieren. Das erhöht wiederum die Investitions- und Betriebskosten signifikant und hat harte technologische Grenzen.

Fazit

- + bietet hohe Verfügbarkeit von mehreren Applikationsinstanzen
- + ist hoch skalierbar
- + optimiert die Auslastung der Systemlandschaft
- + vereinfacht das Deployment durch eine standardisierte API
- + ist durch Infrastructure as Code und Continuous Deployment hoch automatisiert
- erfordert fundiertes Expertenwissen
- setzt an die Laufzeitumgebung angepasste Anwendungen voraus
- verwaltet nur die verfügbaren Ressourcen, aber nicht die Hardware
- befindet sich noch in der Weiterentwicklung



Buzzword Factor (Ent./Customer)

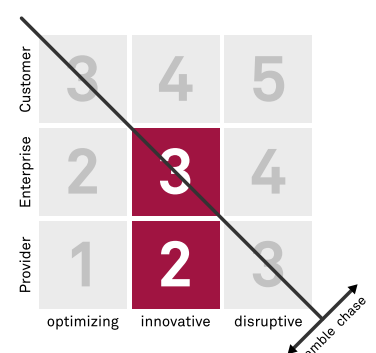
1 low	2 medium	3 high
----------	-------------	-----------

Entry Barrier (Provider)

1 low	2 medium	3 high
----------	-------------	-----------

Benefit Level (Provider)

1 low	2 medium	3 high
----------	-------------	-----------



<https://msg.direct/techrefresh>

Stand: September 2020

msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München | Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113 | www.msg.group | info@msg.group