Container-Based Deployment

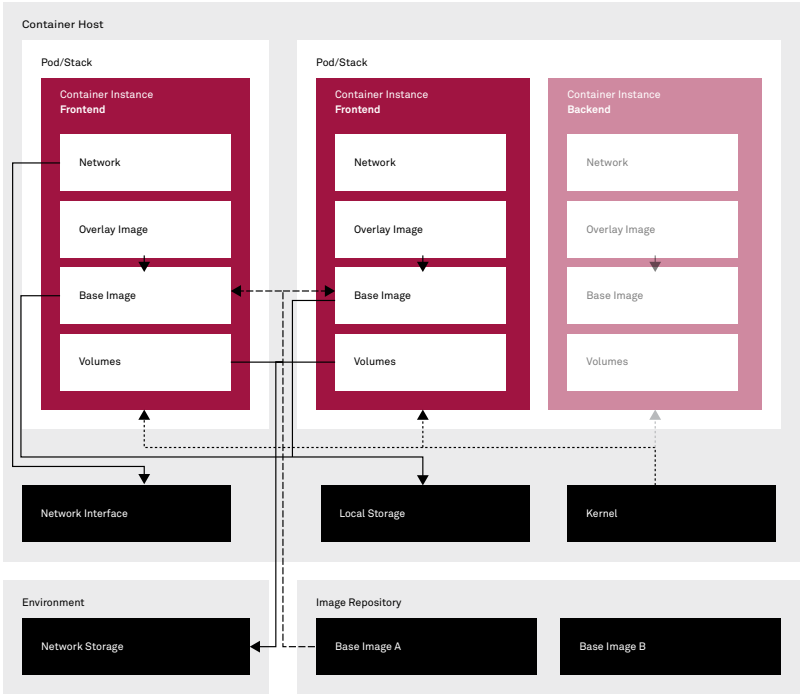
Software hoch flexibel betreiben

Eine kurze Auslieferzeit ist ein entscheidender Wettbewerbsvorteil, zwingt aber zu immer kürzeren Releasezyklen und flexiblen Infrastrukturlösungen. Container-Based Deployment schafft voneinander separierte Laufzeitumgebungen mit standardisierten Schnittstellen zu den Systemressourcen. Dadurch vereinfacht und beschleunigt es die Installation von Software erheblich.

Definition

Container kombinieren eine Anwendung mit all ihren Daten und Abhängigkeiten wie Frameworks, Libraries und Tools zu einer isolierten Komponente. Die Komponente lässt sich beliebig oft replizieren und auf verschiedenen Plattformen ausführen. Container-Based Deployment nutzt die Technologie der Container, um zwei divergierende Anforderungsprofile abzudecken: die des Entwicklers und die des Administrators.

Entwickler haben eine Innensicht auf Container. Innerhalb des Containers stellt ihnen ein Base Image eine einheitliche Laufzeitumgebung für ihre Anwendung oder ihren Service zur Verfügung. Die auf dem Wirtsystem laufenden Container nutzen alle verfügbaren Ressourcen wie Dateiystem, Prozessor, Hauptspeicher und Netzwerk isoliert voneinander und teilen sich lediglich den Betriebssystem-Kernel. Ihren maximalen Nutzen entwickeln Container dann, wenn sie zustandslos (engl. stateless) arbeiten können. Das erfordert unter anderem, dass sie ihre dauerhaft benötigten Daten außerhalb des Containers halten, etwa in Datenbanken oder auf Netzwerkspeichern.



Administratoren haben eine Außensicht auf den Container. Außerhalb des Containers sind die Schnittstellen für Netzwerk und Speicher klar definiert und vereinheitlicht, so dass sich die Installation und Deinstallation von Software durch den Einsatz der Container-Technologie massiv vereinfacht. Herausforderungen wie Sicherheitsaspekte, Datensicherung, Lastverteilung, persistente Speicherung, Netzwerkarchitektur und Überwachung lassen sich für alle Anwendungen zudem generisch und einheitlich lösen. Das schafft einen hohen Grad an Geschlossenheit und führt schlussendlich zu einer hohen Wirtschaftlichkeit.

Referenzszenario

Steht ein Unternehmen bei Fragen des Infrastrukturbetriebs vor einer klassischen Make-or-Buy-Entscheidung, dann lässt sich mit Container-Based Deployment diese Bewertung vereinfachen. Fragen wie „Selbst machen oder einkaufen?“ und „Selbst betreiben oder auslagern?“ rücken

nahezu vollständig in den Hintergrund. Stattdessen schafft Container-Based Deployment einen Grad an Abstraktion, durch den sich selbst komplexere Setups mit kalkulierbarem Aufwand in variierenden Umgebungen beliebig oft und automatisiert reproduzieren lassen.

Potenzial

Die ständige Verfügbarkeit von Anwendungen mitsamt einer hohen Elastizität gegenüber Lastschwankungen entwickelt sich derzeit zur Muss-Anforderung. Container-Based Deployment schafft dafür den Unterbau. Cluster-Lösungen wie Kubernetes oder Docker Swarm benötigen Container-Based Deployment, um Ressourcen dynamisch bereitstellen zu können, Zero Downtime und Scale Out technisch zu realisieren und vor allem kaufmännisch erreichbar werden zu lassen.

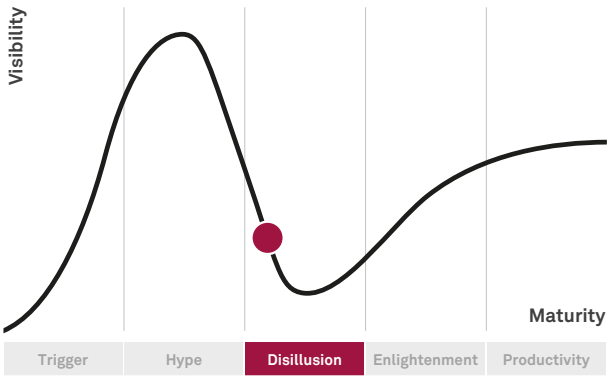
Darüber hinaus schafft die Container-Technologie durch ihre standardisierten Schnittstellen erstmals die Möglichkeit, vollständige fachliche Anwendungen kostengünstig und mitsamt ihren Abhängigkeiten zwischen Servern umzuziehen. Unternehmen profitieren von mehr Markttransparenz, so dass sie in eine komfortableren Position gegenüber den Betreibern der Cloud-Infrastrukturlösung stehen.

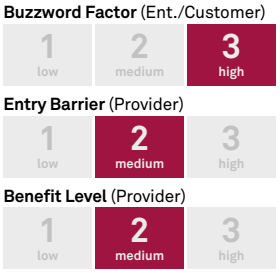
Reifegrad

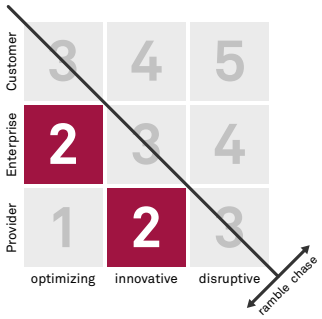
Container-Based Deployment berührt Technologien, die sich durch alle Betriebssystemebenen bis hinunter zum Kernel erstrecken. Die Einzelkomponenten haben einen hohen Reifegrad. Im Vergleich zu Infrastucture-as-a-Service- sind Container-as-a-Service-Produkte allerdings noch nicht weit verbreitet.

Marktübersicht

Das Format der Container und ihre Laufzeitumgebung sind inzwischen durch die Open Container Initiative (kurz OCI) offen standardisiert. Mit Docker (containerd) – der führenden Lösung auf dem Markt – und Kubernetes (CRI-O) bauen die beiden wichtigsten Protagonisten auf diesem Standard auf. Gleiches gilt für Kata, das vor allem die gegenseitige Abschottung der Container fokussiert, sowie Balena, das Container auf eingebettete Systeme und das Internet of Things überträgt.







Vorteilhaft ist, dass offene Standards und Open Source den Markt des Container-Based Deployments dominieren. Ein Vendor-Lock-in tritt somit nicht auf, weil keiner der Anbieter das Thema vereinnahmen und eine marktbeherrschende Stellung einnehmen konnte. Das ist auch unwahrscheinlich, denn die Open Container Initiative treibt sämtliche Schlüsseltechnologien entscheidend voran.

Alternativen

Bis zu einem bestimmten Grad lässt sich die Funktionalität von Container-Based Deployment auch durch klassische Vollvirtualisierung sowie verschiedene Paketformate umsetzen.

Pro	Contra
automatisierte Reproduzierbarkeit	zusätzliche Komplexität
Separation of Concerns	veränderte Angriffsvektoren aufgrund veralteter Container-Images
leichtgewichtiger als Vollvirtualisierung	geringere Parametrierbarkeit
Sicherheit	schwergewichter als Software-Paketierung
verringert Restriktionen hinsichtlich der Entwicklung der Software	auch kleine Korrekturen erfordern den kompletten Austausch des Containers

https://msg.direct/techrefresh

Stand: November 2019

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