

[cloud.google.com](https://cloud.google.com)

# Container und ihre Vorteile | Google Cloud

14-17 Minuten

---

## Google-Ansatz

Bei Google wird von Gmail über YouTube bis hin zur Suche alles in Containern ausgeführt. Dank der Containerisierung arbeiten unsere Entwicklerteams schneller als zuvor und können Software effizient bereitstellen. Wir starten pro Woche mehrere Milliarden Container. In den letzten zehn Jahren haben wir viel über die Ausführung von Containerarbeitslasten in der Produktion gelernt und [dieses Wissen an die Community weitergegeben](#) – von den Anfängen, als wir [cgroups zum Linux-Kernel](#) beisteuerten, bis hin zur Entscheidung, Designs unserer internen Tools als [Open-Source-Projekt Kubernetes](#) bereitzustellen. Dieses Know-how ist nun in der [Google Cloud Platform](#) vereint, sodass Entwickler und Unternehmen jeder Größenordnung problemlos die neuesten Containerinnovationen nutzen können.



Container-Grundlagen: Was sind Container?

Container bieten einen logischen Mechanismus der Paketerstellung, der darauf beruht, dass Anwendungen von ihrer Ausführungsumgebung abstrahiert werden. Mit dieser Entkopplung können containerbasierte Anwendungen einfach und konsistent bereitgestellt werden, unabhängig davon, ob es sich bei der Zielumgebung um ein privates Rechenzentrum, die öffentliche Cloud oder auch um den persönlichen Laptop eines Entwicklers handelt. Die Containerisierung sorgt dafür, dass sich Entwickler auf ihre Anwendungslogik und -abhängigkeiten konzentrieren können, während sich operative IT-Teams auf die Bereitstellung und Verwaltung konzentrieren dürfen, ohne sich Gedanken über Anwendungsdetails wie spezifische Softwareversionen und Konfigurationen machen zu müssen.

Container werden oft mit virtuellen Maschinen (VMs) verglichen. Möglicherweise sind Sie bereits mit VMs vertraut: Bei diesen handelt es sich um ein Gastbetriebssystem wie Linux oder Windows, das auf einem Host-Betriebssystem mit virtualisiertem Zugriff auf die zugrunde liegende Hardware ausgeführt wird. Wie bei virtuellen Maschinen können Sie auch mithilfe von Containern Ihre Anwendung mit Bibliotheken und anderen Abhängigkeiten zusammenpacken. Dadurch wird eine isolierte Umgebung für die Ausführung Ihrer Softwaredienste geschaffen. Wie Sie im Schaubild unten sehen, hören hier die Gemeinsamkeiten aber auch schon auf. Container ermöglichen Entwicklern und operativen IT-Teams ein einfacheres Arbeiten mit einer Vielzahl von Vorteilen.





Virtuelle Maschinen enthalten die folgenden Einheiten, die zur besseren Verdeutlichung übereinander gestapelt sind: Anwendung, Binärprogramm/Bibliotheken, Gastbetriebssystem, Hypervisor, Hostbetriebssystem, Infrastruktur. Container enthalten die folgenden Einheiten, die zur besseren Verdeutlichung übereinander gestapelt sind: Anwendung, Binärprogramm/Bibliotheken, Containerlaufzeit, Hostbetriebssystem, Infrastruktur.

### Vorteile von Containern

Statt einer Virtualisierung der gesamten Hardware wie bei virtuellen Maschinen erfolgt bei Containern eine Virtualisierung auf Betriebssystemebene, bei der mehrere Container direkt auf dem Kernel des Betriebssystems ausgeführt werden. Das bedeutet, dass Container weitaus schlanker sind: Sie teilen sich den Kernel des Betriebssystems, starten schneller und verwenden nur einen Bruchteil des Speichers im Vergleich zum Booten eines vollständigen Betriebssystems.

Containerformate gibt es viele. Ein beliebtes Open-Source-Containerformat ist Docker, das auf der Google Cloud Platform und von Google Kubernetes Engine unterstützt wird.

### Vorteile des Sandboxing

Beim Ausführen in einer Sandbox werden Anwendungen in Containern getrennt voneinander gespeichert, es sei denn, sie sind ausdrücklich miteinander verknüpft. Daher müssen Sie sich keine Gedanken über konfliktverursachende Abhängigkeiten oder Ressourcenbeschränkungen machen, da Sie für jeden Dienst einen

bestimmten Ressourcengrenzwert festlegen. Außerdem entsteht dadurch eine zusätzliche Sicherheitsebene, da Ihre Anwendungen nicht direkt auf dem Host-Betriebssystem ausgeführt werden.



### Konsistente Umgebung

Mit Containern können Entwickler vorhersehbare Umgebungen erstellen, die von anderen Anwendungen getrennt sind. Container können auch Softwareabhängigkeiten enthalten, die für die Anwendung notwendig sind, z. B. spezifische Versionen von Laufzeiten für Programmiersprachen und andere Softwarebibliotheken. Aus Sicht des Entwicklers ist die gesamte Struktur konsistent, unabhängig davon, wo die Anwendung letztendlich bereitgestellt wird. All dies führt zu besserer Produktivität: Entwickler und operative IT-Teams verbringen weniger Zeit mit der Fehlerbehebung und Diagnose der Unterschiede zwischen Umgebungen und haben mehr Zeit für die Programmierung von neuen Funktionen für Nutzer. Außerdem entstehen weniger Programmfehler, da Entwickler bereits in Entwicklungs- und Testumgebungen Annahmen treffen können, bei denen sie sich sicher sein können, dass diese auch in der Produktion zutreffen.



### Überall ausführbar

Container können praktisch überall ausgeführt werden. Dadurch wird die Entwicklung und Bereitstellung auf Linux-, Windows- und Mac-Betriebssystemen, virtuellen Maschinen oder Bare-Metal-Computern, Entwicklerrechnern oder lokalen Rechenzentren und natürlich auch in der öffentlichen Cloud erheblich vereinfacht. Das

weit verbreitete [Docker-Image-Format](#) für Container schafft zusätzliche Portabilität. Sie können Container überall dort verwenden, wo Sie Software ausführen möchten.



## Isolation

Container ermöglichen die Virtualisierung von CPU, Speicher und Netzwerkressourcen auf Betriebssystemebene, sodass Entwickler eine Sandbox-Ansicht des Betriebssystems erhalten, das dadurch logisch von anderen Anwendungen isoliert ist.

	<b>Container – Vorteile</b>	<b>Virtuelle Maschine (VM) – Vorteile</b>
<b>Konsistente Laufzeitumgebung</b>	<i>check</i>	<i>check</i>
<b>Anwendungs-Sandboxing</b>	<i>check</i>	<i>check</i>
<b>Geringe Speichergröße</b>	<i>check</i>	
<b>Niedriger Verwaltungsaufwand</b>	<i>check</i>	

## Vom Code zur Anwendung

Container ermöglichen es Ihnen, die Anwendung und ihre Abhängigkeiten zusammen in ein kurzes Manifest zu packen, das der Versionskontrolle unterliegt und einfach von Entwicklern in Ihrem Team oder Maschinen in Ihrem Cluster repliziert werden kann.

Wie bei Softwarebibliotheken, die Codeabschnitte zusammenpacken, damit Entwickler Logik wie die

Nutzerauthentifizierung oder die Sitzungsverwaltung "wegabstrahieren" können, kann die Anwendung komplett in Container gepackt werden. Dadurch werden in diesem Fall Betriebssystem, Rechner und Code wegabstrahiert. In Verbindung mit einer dienstbasierten Architektur bedeutet dies für Entwickler weniger Arbeitsaufwand für die komplette Einheit. Dies wiederum führt zu erhöhter Produktivität, Flexibilität und Geschwindigkeit. Die Entwicklung, Durchführung von Tests, Bereitstellung und allgemeine Verwaltung von Anwendungen werden erheblich vereinfacht.

### Von monolithischer zu dienstbasierter Architektur

Container eignen sich optimal für dienstbasierte Architekturen. Im Gegensatz zu monolithischen Architekturen, bei denen alle Teile einer Anwendung miteinander verknüpft sind – von den E/A über die Datenverarbeitung bis hin zum Rendering –, werden diese Komponenten bei einer dienstbasierten Architektur getrennt. Durch eine Trennung und Arbeitsteilung werden Ihre Dienste auch dann weiter ausgeführt, wenn andere fehlschlagen. Damit bleibt Ihre gesamte Anwendung zuverlässiger.

Die Einteilung in Komponenten ermöglicht eine schnellere und zuverlässigere Entwicklung, da eine kleinere Codebasis einfacher zu verwalten ist. Außerdem sind bestimmte Eingaben und Ausgaben leichter zu testen, weil Dienste getrennt voneinander ausgeführt werden.

Container eignen sich optimal für dienstbasierte Anwendungen, da Sie den Zustand jedes Containers überprüfen, einzelne Dienste auf bestimmte Ressourcen begrenzen und sie unabhängig voneinander starten und stoppen können.

Durch die Abstraktion von Code können einzelne Dienste wie Blackboxes gehandhabt werden. Damit wird der Raum, um den sich der Entwickler kümmern muss, weiter reduziert. Wenn Entwickler an voneinander abhängigen Diensten arbeiten, können sie einfach einen Container für diesen speziellen Dienst erstellen, ohne dass sie eine ordnungsgemäße Umgebung einrichten und vorab Fehler beheben müssen.

Kubernetes für produktionskonforme Container-Orchestrierung

Wir hatten so viel Erfolg mit unserem internen [Cluster-Management-System Borg](#), dass wir diese Erfahrungen in unser [Open-Source-Projekt Kubernetes](#) eingebracht haben. Jetzt können Sie und andere Unternehmen unsere jahrzehntelange Erfahrung nutzen. Kubernetes, auch bekannt als "k8s", bietet eine automatische Container-Orchestrierung bzw. Verwaltung Ihrer Maschinen und Dienste. Dies führt zu einer verbesserten Zuverlässigkeit sowie einer Reduzierung des Zeitaufwands und der Ressourcen, die Sie für die Entwicklung benötigen.

Kubernetes vereinfacht alles im Zusammenhang mit der Bereitstellung und Verwaltung Ihrer Anwendung. Mit Kubernetes werden Rollouts und Rollbacks automatisiert, sodass der Status Ihrer Dienste überwacht wird und fehlerhafte Rollouts verhindert werden. Außerdem werden regelmäßig Systemdiagnosen Ihrer Dienste durchgeführt und fehlgeschlagene oder gestoppte Container neu gestartet. Den Kunden werden nur Dienste angezeigt, wenn bestätigt wurde, dass diese auch erfolgreich gestartet wurden. Zusätzlich werden Ihre Dienste mithilfe von Kubernetes je nach Nutzung automatisch hoch- oder herunterskaliert, um sicherzustellen, dass nur die Dienste ausgeführt werden, die Sie benötigen. Wie bei Containern ist es mit

Kubernetes möglich, Cluster deklarativ zu verwalten, damit die Einrichtung des Clusters der Versionskontrolle unterliegt und einfach zu replizieren ist.

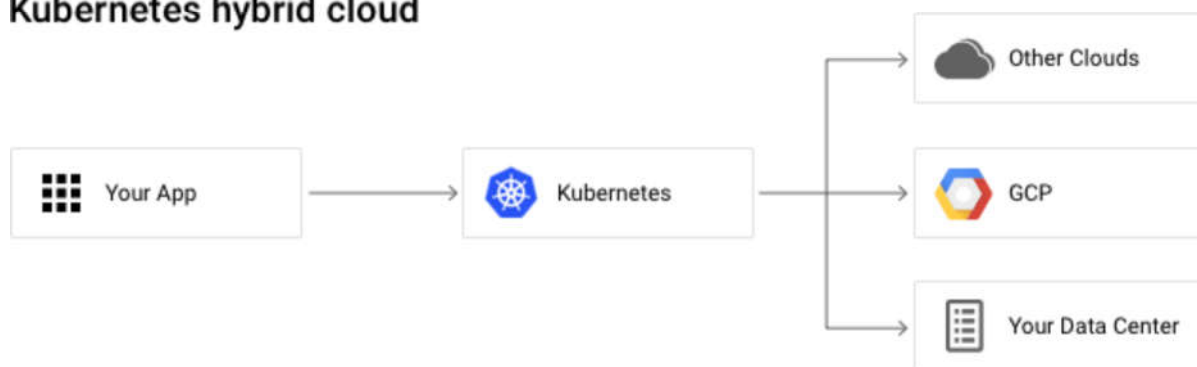
### Kubernetes-Features

- *check* Automatische Einführungen und Rollbacks
- *check* Monitoring des Dienststatus
- *check* Autoscaling von Diensten
- *check* Deklarative Verwaltung
- *check* Überall bereitstellbar, auch in Hybridbereitstellungen

Kubernetes kann überall eingesetzt werden, sodass eine Orchestrierung von lokalen Bereitstellungen über öffentliche Clouds bis hin zu Hybridbereitstellungen vorgenommen werden kann.

Dadurch erreicht Ihre Infrastruktur die Nutzer an deren Standorten, Ihre Anwendungen haben eine höhere Verfügbarkeit und Ihr Unternehmen kann genau die richtige Balance von Sicherheits- und Kostenaspekten finden – alles abgestimmt auf Ihre speziellen Bedürfnisse.

### Kubernetes hybrid cloud



### Cluster bei Google

Kubernetes wird am besten auf der Google Cloud Platform ausgeführt. [Google Kubernetes Engine](#) ist die führende verwaltete Kubernetes-Lösung, die eine schnelle Einrichtung und



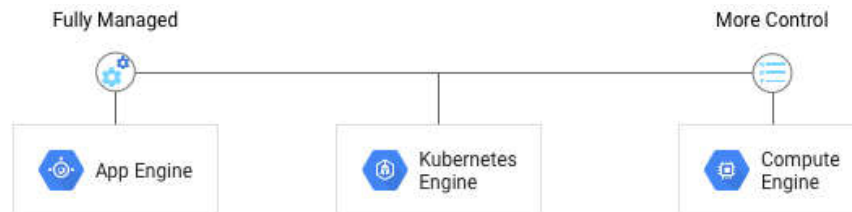
Produktionsbereitschaft ermöglicht.

Kubernetes Engine wird vollständig durch Reliability Engineers von Google verwaltet, die sich mit Containern am besten auskennen. Dadurch werden Cluster immer verfügbar und aktuell gehalten. Kubernetes Engine lässt sich nahtlos in alle GCP-Dienste einbinden, z. B. [Stackdriver](#) Monitoring, Diagnostics und Logging, [Identity and Access Management](#) sowie die erstklassige [Netzwerkinfrastruktur](#) von Google.

Features von Kubernetes Engine

- *check* Verwaltetes Open-Source-Kubernetes
- *check* SLA mit 99,5 % sowie Hochverfügbarkeit mit integrierten Bereitstellungen über mehrere Zonen
- *check* Nahtlose Integration weiterer GCP-Dienste
- *check* Branchenführendes Preis-Leistungs-Verhältnis
- *check* Flexibel und interoperabel mit lokalen Clustern und anderen Cloud-Anbietern
- *check* Verwaltete Infrastruktur auf Google-Niveau

Darüber hinaus sind noch weitere Optionen verfügbar. Die Google Cloud Platform bietet Ihnen ein volles Funktionsspektrum für die Ausführung Ihrer Container. Egal, ob Sie eine vollständig verwaltete Umgebung mit [Google Cloud Run](#) nutzen, Ihre Cluster mit Kubernetes Engine verwalten oder die Infrastruktur selbst in [Google Compute Engine](#) bereitstellen und von einem erstklassigen Preis-Leistungs-Verhältnis profitieren möchten – auf der Google Cloud Platform finden Sie die optimale Lösung für die Containerausführung.



## Vollständige Container-Lösung

Zusätzlich stellt die Google Cloud Platform Tools bereit, mit denen Container von der Entwicklungsphase in die Produktionsphase überführt werden können. [Cloud Build](#) und [Container Registry](#) bieten Docker-Image-Speicher und -Verwaltung basierend auf den hohen Sicherheitsstandards und dem erstklassigen Netzwerk von Google. Mit [Container-Optimized OS](#) von Google erhalten Sie ein schlankes, überaus sicheres Betriebssystem mit vorinstallierter Docker- und Kubernetes-Laufzeit. Die gesamte Containerverwaltung kann auf der GCP durchgeführt werden.

