

Container-Orchestrierung: Was ist eigentlich Kubernetes?

Diego Wyllie • 28.11.2017, 07:00 Uhr



(Foto: Pikoso.kz / Shutterstock)

Hinweis: Wir haben in diesem Artikel Provisions-Links verwendet und sie durch "*" gekennzeichnet. Erfolgt über diese Links eine Bestellung, erhält t3n.de eine Provision.

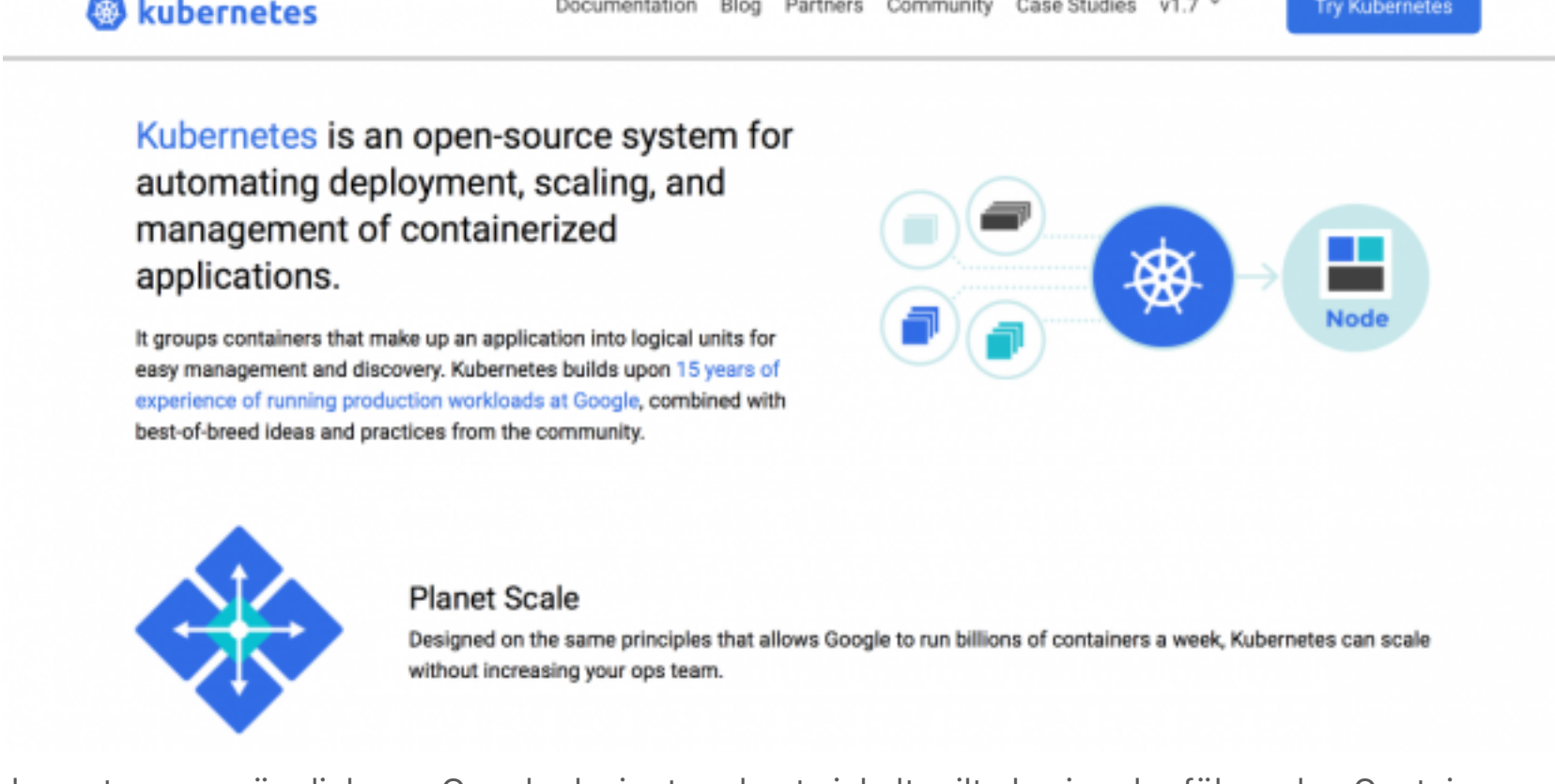
Entwickler, die Docker nutzen und ihre Anwendungen skalieren müssen, brauchen eine effiziente Lösung zum Management der Container-Landschaft.

Hier kommt [Kubernetes](#) ins Spiel. Im Folgenden zeigen wir, was die führende Container-Orchestrierungs-Lösung kann und wie sie in der Praxis eingesetzt wird.

Die Container-Lösung [Docker](#) hat die Art und Weise wie Serveranwendungen bereitgestellt werden revolutioniert. Noch nie ist es nämlich so einfach gewesen, eine Applikation auf verschiedene Plattformen und Umgebungen (beispielsweise Entwicklung, QA, Produktion) zu bringen wie es heutzutage dank Docker der Fall ist. Der Clou: Docker packt die Anwendung und alle für deren Ausführung benötigten Systemkomponenten in einer einzigen Datei, den sogenannten „Container“. Docker-Container sorgen dafür, dass die Anwendung verlässlich läuft, nachdem sie von einer Umgebung in eine Andere versetzt worden ist. Dies vereinfacht nicht nur das Deployment komplexer Anwendungen, die auf verschiedenen Servern laufen, sondern ermöglicht gleichzeitig eine flexiblere Anwendungsinfrastruktur, die sich leichter ändern, erweitern und skalieren lässt. So gilt Docker [als die perfekte Verpackung für Microservices](#), die derzeit stark im Kommen sind. Überall werden mithilfe von Docker große monolithische Systeme, die vor der Cloud-Ära entwickelt wurden, [durch kleine, eigenständige Microservices ersetzt](#).

Container über mehrere Hosts orchestrieren

Mit dem Siegeszug von Containern und Microservices ist rund um Docker ein umfangreiches Ökosystem entstanden, in dem Entwickler viele mächtige Lösungen finden, mit denen sie die Möglichkeiten der innovativen Container-Lösung erweitern können. Eine der Beliebtesten davon ist sicherlich [Kubernetes](#), kurz „K8s“. Dabei handelt es sich um ein Container-Orchestrierungs-System, das Entwickler und IT-Administratoren in die Lage versetzt, Deployment, Betrieb, Wartung und Skalierung von Container-basierten Anwendungen zu vereinfachen und zu automatisieren. Entscheidend dabei: Gruppen von Hosts, auf denen die Container laufen, werden in Clustern von physischen oder virtuellen Maschinen zusammengefasst und als Einheit verwaltet.

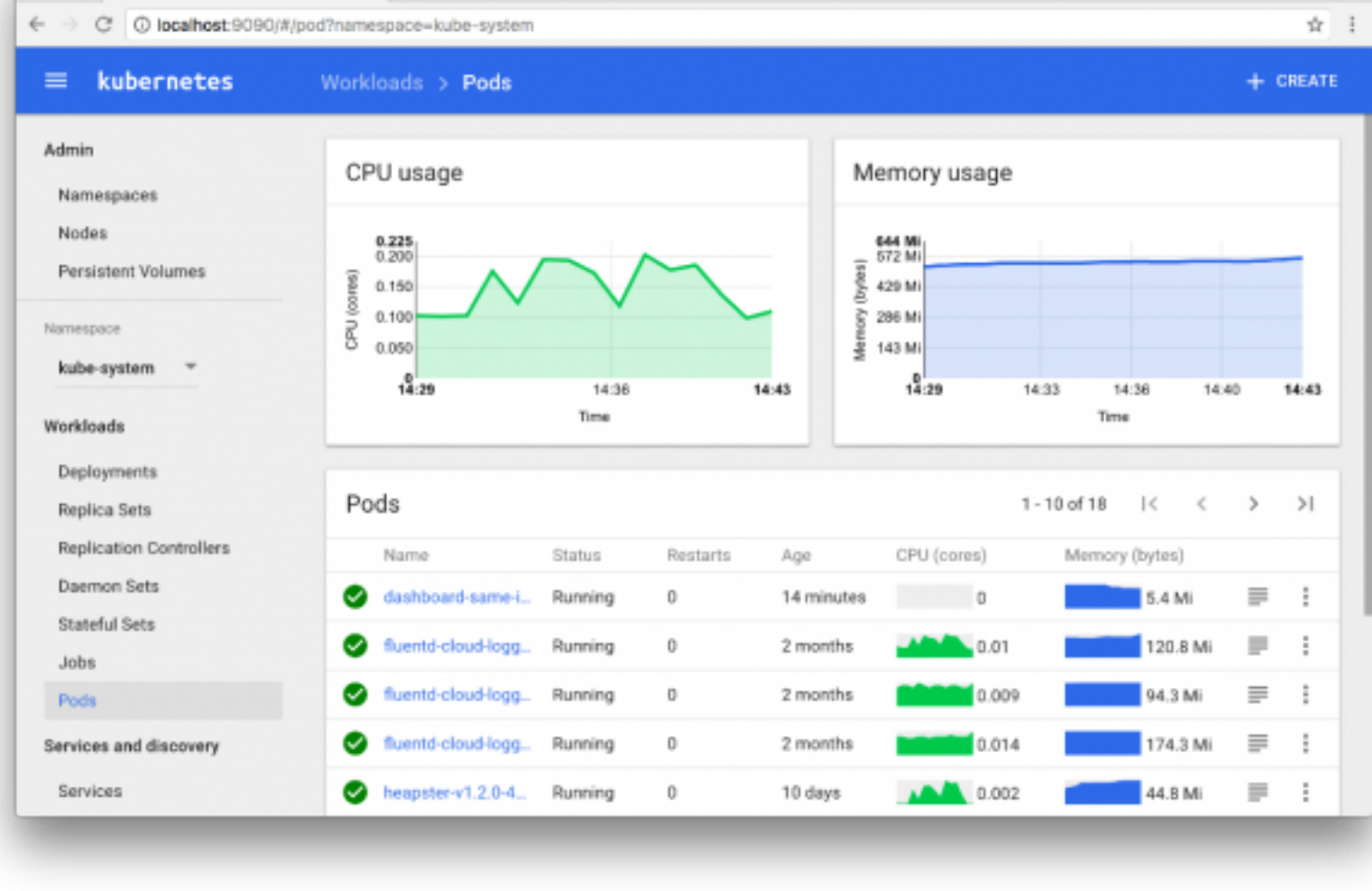


Kubernetes, ursprünglich von Google design und entwickelt, gilt als eine der führenden Container-Orchestrierungslösungen, die derzeit um die Gunst der Entwickler kämpfen.

Kubernetes wurde ursprünglich von Google design und entwickelt. 2015 kam die erste Version auf den Markt und der Internetrieser spendete das Projekt als Open-Source-Software der neu gegründeten [Cloud Native Computing Foundation](#), in der mittlerweile so gut wie alle wichtigen Cloud-Player [vertreten sind](#). Die Orchestrierung mittels Kubernetes wird heute von führenden Cloud-Plattformen wie [Azure *](#) (Microsoft), [Bluemix](#) (IBM) und [Openshift](#) (Red Hat) unterstützt und von vielen namhaften Unternehmen, darunter The New York Times, Phillips, SAP und SaaS-Anbieter Box und Github, produktiv eingesetzt.

Container, Pods, Nodes und mehr

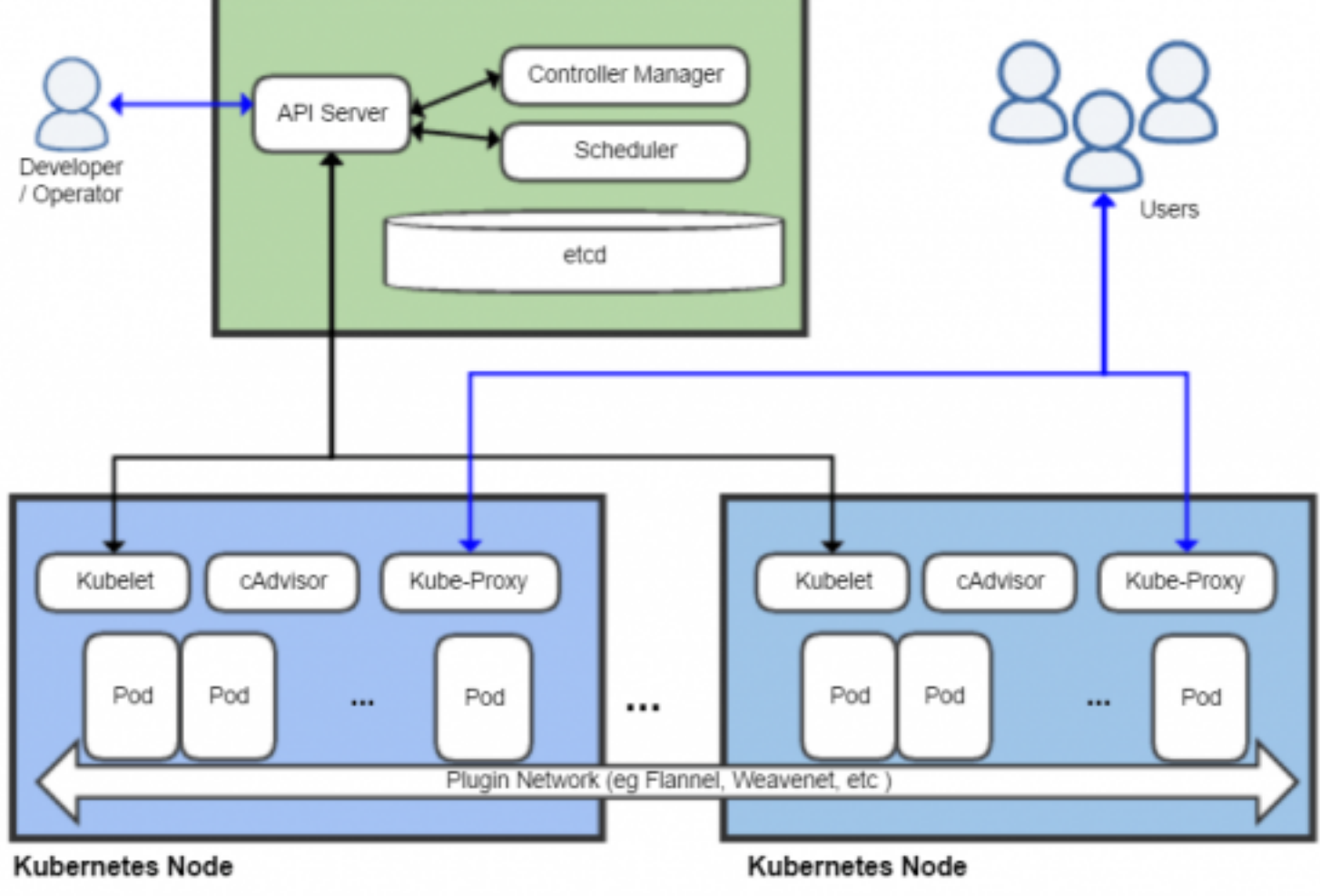
Kubernetes stellt eine umfangreiche Plattform dar, die aus zahlreichen Komponenten und Tools besteht – beispielsweise Kommandozeile, Web UI und APIs. Diese sorgen dafür, dass Administratoren komplexe Container-Cluster auf effiziente und professionelle Art und Weise verwalten können – und das über verschiedene Umgebungen wie Public, Private oder Hybrid Clouds hinweg. Mehrere Container werden dabei in einen „Pod“ zusammengefasst. Pods stellen eine Basiseinheit dar, die einen oder mehrere Container hostet, die sich Ressourcen teilen und auf der gleichen physischen oder virtuellen Maschine – „[Nodes](#)“ genannt, früher „Minions“ – ausgeführt werden. Alle Container in einem Pod teilen sich die IP-Adresse, IPC, Hostname und andere Ressourcen. Für jeden Pod kann Kubernetes automatisiert Maschinen mit ausreichend Rechenkapazität finden und die zusammenhängenden Container starten. Damit wird den gruppierten Containern eine Abstraktionsebene hinzugefügt, die dem Administrator hilft, seinen Workload zu planen und diesen Containern die notwendigen Dienste wie Networking und Storage zu liefern.



Neben vielen Kommandozeilen-Tools bietet Kubernetes ein Web-Dashboard, das die Verwaltung von Deployments, Nodes, Pods und vielem mehr vereinfacht.

Automatisiertes Deployment und Management

Kubernetes läuft aufgesetzt auf ein Betriebssystem und interagiert mit den Pods von Containern, die auf den Nodes laufen. Der sogenannte „Kubernetes Master“ spielt dabei eine zentrale Rolle. Er empfängt Befehle vom Administrator und leitet diese an die untergeordneten Nodes weiter. Diese Übergabe arbeitet mit einer Vielzahl von Diensten, um automatisch zu entscheiden, welcher Node am besten für die Aufgabe geeignet ist. Der Master weist dann Ressourcen zu und legt die Pods in diesem Node fest, die die gewünschte Aufgabe durchführen sollen.



Kubernetes basiert auf einer durchdachten Systemarchitektur, die die Orchestrierung von Containern über mehrere Maschinen ermöglicht.

Wenn Kubernetes einen Pod für einen Node einplant, wird „Kubelet“ auf diesem Node Docker anweisen, die spezifischen Container zu starten. Kubelet erfasst dann kontinuierlich den Status dieser Container von Docker und sammelt diese Informationen im Master. Anhand dieser Metriken ist Kubernetes in der Lage, Container, die fehlerhaft sind oder überhaupt nicht reagieren, automatisiert neuzustarten, zu ersetzen und zu stoppen – Stichwort „Self-Healing“. Zu den weiteren zentralen Features von Kubernetes zählen zudem Load-Balancing, horizontales Skalieren, automatisierte Rollouts, sowie professionelles [Konfigurationsmanagement](#).

Kubernetes löst monolithische Systeme ab

Durch die Flexibilität, die Container und deren Orchestrierung mit Kubernetes ermöglichen, können Entwickler-Teams eigenverantwortlicher und agiler arbeiten – und somit neue Funktionen, Services und Fixes schneller ausrollen – ohne Downtime. In der Praxis haben sich viele namhafte Unternehmen und Startups deshalb dafür entschieden, monolithische Legacy-Systeme durch Container-Cluster zu ersetzen. Ein gutes Beispiel dafür ist [Github](#). Wie das Entwickler-Team hinter dem populären Code-Hosting-Dienst neulich [in seinem Blog schreibt](#), wurde das über acht Jahre lang entwickelte monolithische System, auf dem die Github-Website und -API basieren, durch Kubernetes ersetzt. Einzelne Entwickler-Teams können nun kleinere, spezielle Services leichter und vor allem wesentlich schneller als bisher selbst ausrollen, heißt es in dem Blog-Post.

Ein weiterer SaaS-Player, der sich für Kubernetes entschieden hat, um die Agilität seiner Entwickler-Teams zu erhöhen, ist Box. In einem aufwändigen Projekt, das sich über eineinhalb Jahre erstreckte, hat der Content-Management-Anbieter sein auf PHP basiertes System, das lange vor Amazon-Web-Services und Container entwickelt wurde, komplett durch eine auf Kubernetes basierende Microservices-Infrastruktur ersetzt. Während Entwickler früher bis zu sechs Monate warten mussten, um einen neuen Service produktiv zu deployen – das Unternehmen betrieb ihre eigene Server-Hardware –, dauert das heute dank der neuen Microservices-Infrastruktur auf Basis von Containern und Kubernetes im Durchschnitt [weniger als eine Woche *](#).

Fazit

Kubernetes verspricht Entwicklern, IT-Administratoren und DevOps-Teams Entwicklung, Deployment und Betrieb komplexer Server-Anwendungen, die eine hohe Last bewältigen und schnell und automatisiert skalieren müssen, radikal zu optimieren. Das Konzept setzt sich in der Praxis immer weiter durch – und das sowohl bei Startups als auch bei großen Konzernen. Github und Box zeigen eindrucksvoll, wie komplexe Monolithen aus der PC-Ära sich komplett durch eine moderne Infrastruktur ersetzen lassen, die auf Container und Microservices basiert und sich mit Kubernetes professionell verwalten lässt. Wie beide Beispiele zeigen, ist der Umstieg sicherlich nicht leicht. Die Lernkurve bei der Implementierung von Docker und Kubernetes ist hoch. Doch früher oder später werden sich Softwarehersteller mit diesem innovativen Architektur- und Entwicklungsmodell auseinandersetzen müssen, wenn sie dem Konkurrenzdruck, der in der Branche herrscht, standhalten wollen. Denn die Flexibilität und Agilität, die diese modernen Technologien mit sich bringen, werden verstärkt zu einem entscheidenden Wettbewerbsvorteil.

[Zur Startseite](#)