

iX *extra* April 2020 **Hosting**

Eine Sonderveröffentlichung der Heise Medien GmbH & Co. KG

Hosting für Entwickler

Docker und Entwicklerwerkzeuge vom Hoster

Im Container

Seite 120

Entwicklertrend Microservices

Seite 122

Entwicklertrend DevOps

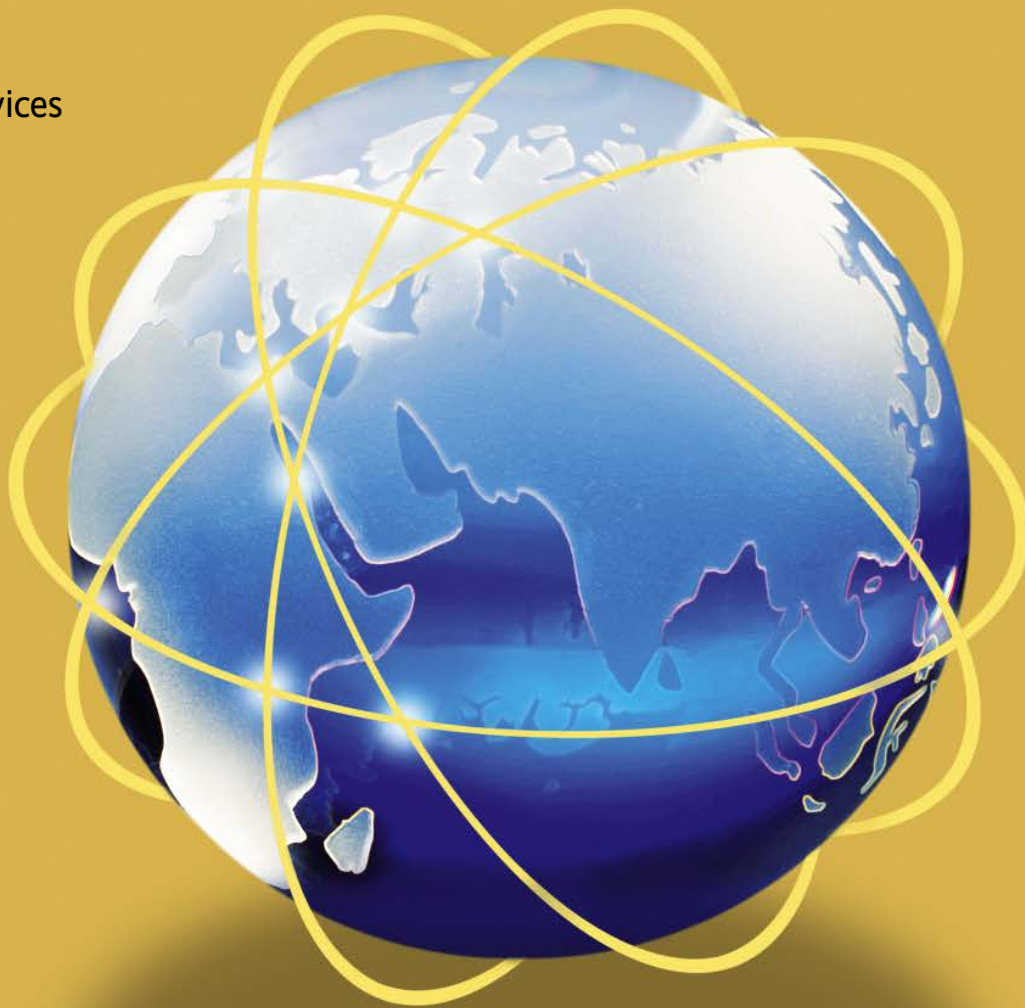
Seite 124

Kubernetes –
der Flottenmanager

Seite 127

Deutsche Hoster mit
Produkten für Entwickler

Seite 128



iX extra zum Nachschlagen:
www.ix.de/extra

Im Container

Docker und Entwicklerwerkzeuge vom Hoster

Softwareentwicklung bedeutet heutzutage das Verteilen von Aufgaben, Werkzeugen und Ressourcen. Von besonderer Bedeutung sind Microservices und Anwendungen in Containern. Hosting-Provider bieten sie sowohl als Entwicklungs- als auch als Produktionsumgebungen an.

Jahrzehntlang waren Großunternehmen stolz auf ihre höchst zuverlässigen Mainframes. Beim Versuch des Umzugs auf moderne Serverarchitekturen zeigte sich aber, dass sich deren monolithische, gewachsene Software nicht so einfach migrieren lässt. Auch deshalb laufen viele Großrechner viel länger als geplant.

Umso wichtiger ist es, neue Anwendungen von Anfang an nach modernen Gesichtspunk-

ten der Softwareentwicklung zu planen. Dazu gehören Microservices, DevOps, agiles Development, Continuous Delivery und Container. Ein erster Schritt bestand in der Virtualisierung. Virtuelle Maschinen abstrahieren Entwicklungs- und Laufzeitumgebungen von der konkreten Hardware und nutzen sie besser aus. Hierzu bilden Virtualisierungsprodukte wie KVM oder Xen ebenso wie die kommerziellen Pendanten von Microsoft und

VMware komplette Computer mit eigenem BIOS nach, die die Software wie einen physischen Server wahrnimmt. Dies stellt ein hohes Maß an Kompatibilität sicher, bedeutet aber auch einen großen Overhead, der sich mit zunehmender Anzahl virtueller Maschinen immer stärker bemerkbar macht, da derselbe Softwarestack zum Einsatz kommt wie auf dedizierten Servern: Betriebssystem, Datenbanken, Skriptsprachen, Bibliotheken und Konfigurationsdateien sind in jeder VM einzeln zu installieren und bei Bedarf zu aktualisieren. Das führt zu den bekannten Schwierigkeiten des Testens einer Produktionsumgebung und birgt Risiken beim Update oder Patchen einzelner Komponenten.

Wie schwer beherrschbar die Abhängigkeiten zwischen den Versionen verschiedener Softwaremodule sind, zeigte der Ausfall des europäischen Satellitennavigationssystems Galileo im September 2019. Die Kontrollzentren sind zwar georedundant ausgelegt, liefern

seinerzeit aber mit unterschiedlichen Softwareversionen – und nach Störungen eines Standorts keine identische Backup-Site zur Verfügung. Updates sind zudem so komplex, dass sie mehrere Tage in Anspruch nehmen. Dies führte dazu, dass Galileo eine Woche lang keine korrekten Positionsdaten liefern konnte.

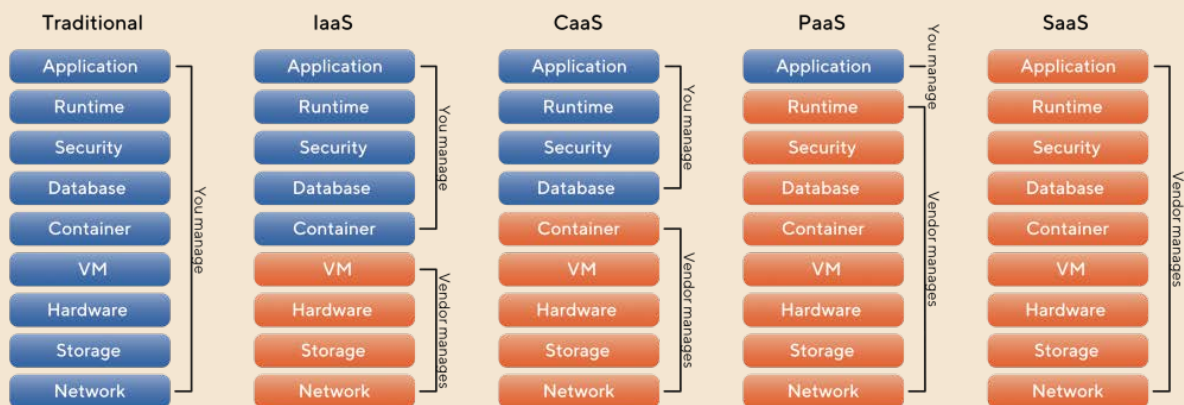
Ladung und Transport

Da einzelne Anwendungen keine komplette virtuelle Maschine benötigen, gingen Hosting-Provider schnell dazu über, vServer oder Shared Webhosting mittels sogenannter Paravirtualisierung zu produzieren, die die Ressourcen effizienter nutzt. Hieraus entwickelten sich virtualisierte Kernel-Ressourcen in Linux-Containern (LXC) und schließlich portable Container, die mittlerweile eine eigene Softwarearchitektur bilden. Was auf den ersten Blick wie ein abgerüsteter virtueller Server aussieht, entpuppt sich als viel mehr, nämlich eine komplette Laufzeitumgebung für eine Applikation. Es gibt dann keine emulierte Serverhardware mit BIOS und eigenem Betriebssystem mehr; vielmehr teilen sich alle Container einen gemeinsamen Host-Kernel und machen einen Hypervisor entbehrlieh (siehe Tabelle „Containertechnik im Vergleich“). Damit reduziert sich der Ressourcenbedarf eines Containers gegenüber einer kompletten VM und die Container-Images fallen deutlich klei-

Containertechnik im Vergleich zu VMs und dedizierten Servern

Kriterium	Container	virtuelle Maschinen	Bare-Metal-Server
zugrunde liegende Plattform	Betriebssystem auf virtuellen oder dedizierten Servern	Hypervisor auf dedizierten Servern	keine
Performance	mittel	mittel	hoch
Zeitbedarf zum Einrichten einer Instanz	Sekunden	Minuten	Stunden
Trennungsebene zwischen Instanzen	Betriebssystem-Kernel	Hypervisor	Hardware
Flexibilität der Konfiguration	hoch	mittel	niedrig
Konsolidierungsdichte einzelner Instanzen	hoch	mittel	keine
Portabilität von Instanzen	Container-Images	VM-Images	ISO-Images, Backup/Restore
Granularität	klein	mittel	groß

Unterschiedliche Betriebsmodelle teilen die Verantwortung zwischen Hosting-Provider und Nutzer auf (Abb. 1).



Quelle: The Cloud Report

ner aus – unter anderem, weil der einige Megabyte große Linux-Kernel entfällt.

Nicht nur der Begriff „Container“ stammt aus der realen Welt, sondern auch das Funktionsprinzip: Container transportieren Güter ganz unterschiedlicher Konsistenz: Schüttgut, Paletten, Säcke oder Fahrzeuge. Sie alle erforderten früher neben viel Handarbeit eine vielfältige Verlade- und Transportinfrastruktur. Die Erfindung des Standardcontainers änderte die Logistikbranche radikal. Zwar gibt es auch hier Spezialanfertigungen (Kühlcontainer, Tankcontainer, Auto- oder Wohncontainer), für Logistiker ergibt sich aber kaum ein Unterschied. Die Hersteller fokussieren sich ganz aufs Entwickeln und Produzieren und die Transportunternehmen bringen Container von A nach B – unabhängig vom Inhalt. Neben

den Abmessungen der Container sind auch Schnittstellen und Dokumentationen standardisiert. Das unterstützt die Automatisierung der gesamten Logistikkette.

Wie im richtigen Leben

Auf die IT übertragen konzentrieren sich Entwickler ganz auf die Anwendungen, ohne Rücksicht auf unterschiedliche Versionen und Updates in der Produktivumgebung zu nehmen. Auf der anderen Seite betreiben Hosting- und Cloud-Provider Container, ohne sich mit Betriebssystemen, Bibliotheken und Softwareaktualisierungen befassen zu müssen.

Ursprünglich für Linux entwickelt, gibt es Containerumgebungen inzwischen auch für Windows und macOS. Bei den

Hostern dominieren aber weiterhin die Linux-Versionen. Container bieten den Applikationen eine vollständige Laufzeitumgebung, die sowohl die Anwendungen als auch die benötigten Systemkomponenten (Libraries) enthält. Damit entsprechen Container einer einfachen Plattform as a Service (PaaS, siehe Abbildung 1).

Jeder Container läuft als eigenständiger Prozess und kann separat gestartet und beendet werden. Die Anwendungsentwicklung folgt dem App-Prinzip einer schnellen Programmierung und Auslieferung kleiner, autonomer Programme (siehe Kasten zu Microservices). Das macht sich bereits beim Start bemerkbar. Wo virtuelle Maschinen ein komplettes Betriebssystem booten, starten Container nur einen Prozess. Zudem muss jede VM regelmäßig mit diversen Updates für

Betriebssystem, Bibliotheken, Datenbanken, Anwendungen etc. versorgt werden. Für Container gestalten sich Änderungen viel einfacher. Sie finden nicht innerhalb der Produktivumgebung statt, sondern werden in einem neuen Image getestet und dann ausgetauscht, was die gesamte Laufzeitumgebung einschließt. Auch müssen sie nicht wie VMs bootfähig sein, sondern nur den Formatvorgaben der jeweiligen Containerumgebung entsprechen. Umgekehrt bleiben nach dem Löschen eines Containers keine Konfigurationseinträge und Codereste im Betriebssystem zurück.

Neben der Portabilität zeichnen sich Container durch Zustandslosigkeit aus: Startet ein Image, beginnt es seine Arbeit wieder wie beim ersten Mal. Das ist Segen und Fluch zugleich. Bereitet ein Container Schwierigkeiten, reicht ein Neustart, um den Ursprungszustand wiederherzustellen. Allerdings gehen damit auch alle Änderungen innerhalb des Containers verloren. Deshalb enthält er in der Regel keine Daten – diese liegen auf separaten virtuellen Datenträgern. Spezielle Data Volume Container können externen Storage mounten und für mehrere andere Container zur Verfügung stellen. Die Orchestrationsplattform Kubernetes kann Container erzeugen, die Zustände – meist Nutzerdaten – speichern (stateful). Konfigurationsdateien verwaltet der Docker-Daemon in einem separaten Volume. So bleiben beim Austausch eines Containers die Konfigurationen erhalten.

Container und virtuelle Maschinen folgen zwar einer unterschiedlichen Architektur, lassen sich aber auch gemeinsam einsetzen, denn außer auf Bare-Metal-Hardware lassen sich Containerumgebungen auch in virtuellen Maschinen installieren. So laufen VMs und Docker-Instanzen parallel auf derselben Infrastruktur (siehe Abbildung 2). Die Anbieter realisieren das auf unterschied-

Entwicklertrend Microservices

Hinter dem Konzept von Microservices steht eine neuartige Architektur für den Aufbau komplexer Softwareanwendungen, die sich aus mehreren kleineren Modulen zusammensetzen. Jeder dieser Services hat eine dedizierte Aufgabe und nutzt eine genau definierte Schnittstelle zum Kommunizieren mit den anderen Komponenten. Auf diese Weise lassen sich einzelne Funktionen unabhängig voneinander entwickeln, testen und gegebenenfalls ersetzen, ohne andere Module zu beeinflussen. Auch die genutzten Programmiersprachen und Plattformen können unterschiedlich sein. Damit ist zum einen eine Parallelisierung bei der Softwareentwicklung möglich, zum anderen ein hoher Wiedernutzungsgrad einzelner Funktionen.

Microservices sollen vor allem den Aufwand für die Tests nach Änderungen im Quellcode reduzieren. In monolithischen Architekturen sind auch nach kleinen Korrekturen sämtliche Funktionen wiederholt zu testen. Andernfalls besteht die Gefahr, dass ein kleiner Fehler die gesamte

Anwendung in Mitleidenschaft zieht.

Als Microservices können beispielsweise in einem Onlineshop Kunden- und Artikelverwaltung, Warenkorb, Zahlungsvorgang und Suchfunktion separat gestaltet werden. Jedes Modul lässt sich einzeln skalieren oder ersetzen. Steigt die Anzahl unterschiedlicher Artikel stark, erhält dieses Modul mehr Ressourcen oder neue Instanzen. Beim Wechsel des Zahlungsanbieters sind alle anderen Services davon nicht betroffen. Das Platzieren von Anzeigen obliegt in der Regel nicht dem eigenen Code, sondern wird von Spezialisten zugekauft. Auch der Ausfall eines einzelnen Service muss nicht den gesamten Shop lahmlegen: Wenn die Suchfunktion temporär nicht zur Verfügung steht, können die Kunden trotzdem weiter einkaufen. Die Skalierung jedes einzelnen Moduls wird als Scale-out bezeichnet. Monolithische Software lässt sich nur durch eine erhebliche Aufrüstung von Prozesskernen, Hauptspeicher und SSDs beschleunigen (Scale-up). Bei allen Vorteilen der

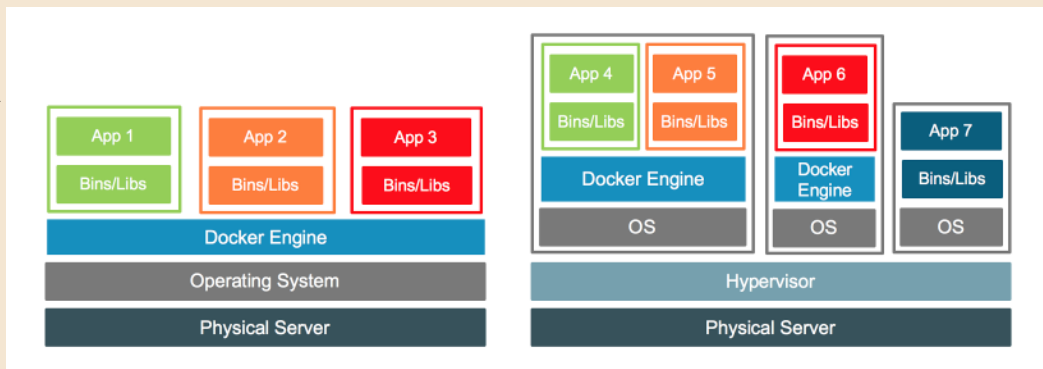
Microservices: Solch ein verteiltes System mit vielen Schnittstellen birgt auch eine hohe Komplexität mit ganz eigenen Herausforderungen.

Microservices werden oft direkt mit Containern assoziiert, es gibt aber auch andere Konzepte hierfür. Ein Beispiel ist AWS Lambda, das es erlaubt, in Java, JavaScript oder Python geschriebene Funktionen einzeln zu starten.

Diese Vorgehensweise nennt sich Serverless Computing und gilt als besonders geeignete Plattform für Microservices. Ähnlichkeiten ergeben sich auch zu serviceorientierten Architekturen (SOA), die insbesondere von Service Providern dafür beworben werden, Dienste zu standardisieren und zentral bereitzustellen. Während sich hinter SOA aber ein unternehmensweites Datenmodell mit Schnittstellendefinitionen und zentralen Steuerungsmechanismen verbirgt, kapseln Microservices nur einzelne Dienste und nutzen einfache Kommunikationsprotokolle wie RESTful HTTP.

liche Weise; hostNET etwa möchte die Vorteile von Container-Images auf virtuelle Maschinen übertragen. Hierzu stehen vorgefertigte Hosting-Scenes zur Verfügung, die Betriebssystem, Skriptsprache, Datenbank, Webserver und je nach Bedarf weitere Serverkomponenten enthalten. Sie lassen sich auf unterschiedliche Weise kombinieren und als vorgefertigtes Image einspielen (siehe Kasten zu DevOps).

Quelle: docker.com



Vergleich der Softwarestacks von Containern mit virtuellen Maschinen: Innerhalb einer VM kann auch eine Docker-Umgebung laufen (Abb. 2).

Alle wollen Docker

Die bekannteste Containerumgebung ist die Open-Source-Software Docker. Ihre Bedeutung lässt sich an der Unterstützung durch alle Schwergewichte der Branche ablesen – von Amazon und Google über IBM und Microsoft bis zu VMware (siehe Tabelle zu CaaS-Plattformen). Daneben etablieren sich reine Do-

cker-Hoster wie DigitalOcean, Jelastic, Quay.io oder Orchard. In Deutschland sind solche Start-ups traditionell rar, aber es gibt auch hier Beispiele wie Giant Swarm, Loodse und sloppy.io. Pivotal ist zwar in Kalifornien beheimatet, betreibt aber auch ein Büro in Berlin.

Dank des hohen Automatisierungsgrades einer Docker-Umgebung können gerade kleine Hoster sie schnell an-

bieten. Doch auch bei den klassischen Providern werden Docker-Umgebungen immer mehr zu einem Standardprodukt. Die hohe Bedeutung der Containertechnologie auch für die großen Hardware- und Softwarehersteller zeigen die Akquisitionen der Kubernetes-Spezialisten Heptio und StackPoint durch VMware und NetApp (jetzt NetApp Kubernetes Service, NKS).

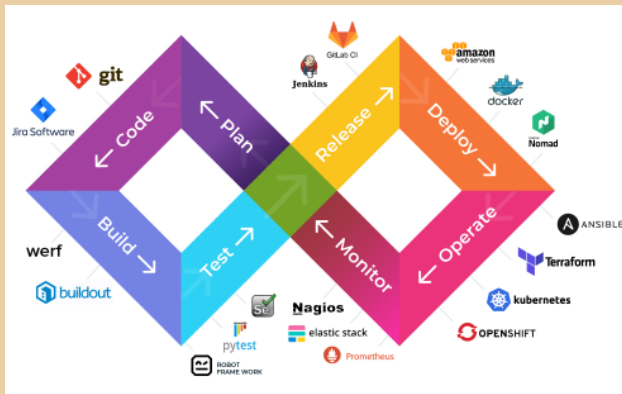
Die Docker-Infrastrukturen von Amazon, Google und Microsoft stehen zwar in Konkurrenz zu Services lokaler Provider, ergänzen sie aber auch. Einige Hoster produzieren die Laufzeitumgebungen nicht mehr selbst, sondern nutzen vorhandene Plattformen. So setzt root360 auf AWS Elastic Container und Kubernetes Service. Damit verschmelzen Hosting-Services

Entwicklertrend DevOps

In der Vergangenheit galt die strikte Trennung von Softwareentwicklung und -betrieb als Gesetz. Schließlich sind die Ziele gegensätzlich: viele schnelle Updates auf der einen Seite, um neue Funktionen zu implementieren und die Bedienbarkeit zu verbessern – stabiler Betrieb mit möglichst wenigen Änderungen auf der anderen. Im App-Zeitalter

wird diese Arbeitsteilung zunehmend kontraproduktiv. Das Buzzword heißt DevOps – eine Kombination aus Development und Operations.

DevOps steht für den nahtlosen Übergang der Programmentwicklung in den Betrieb. Ein zentraler Gedanke besteht in der Automatisierung bisher manueller Tätig-



Toolchain: DevOps begleiten den gesamten Entwicklungs- und Betriebsprozess einer Software (Abb. 3).

Quelle: Cusy

Entwicklertask	Softwaretool
Projektmanagement	Jira Software
Dokumentenmanagement	Confluence
Versionsverwaltung	Gitblit
Code-Review	Gerrit
Testen/Deployment	Jenkins
Log-Management und Analyse	Sentry
Helpdesk	Jira Service Desk
Webanalyse	Matomo

keiten. Deshalb wird DevOps oft gleichgesetzt mit Tools für Versionskontrolle, Konfigurationsmanagement, Testing, Feature- und Fehler-Tracking, Softwareverteilung und Monitoring. Diese Tools entlang des Prozesses der Softwareentwicklung nennen sich DevOps-Toolchain.

In gleicher Weise steht DevOps aber auch für neue Prozesse und organisatorische Veränderungen. So profitiert DevOps besonders von Containern, da sich Testen, Softwareverteilung und Konfigu-

rationsmanagement vereinfachen. Mit dem Verschmelzen der Softwareumgebungen für Entwicklung und Betrieb wird die Trennung gleichsam aufgehoben. Da DevOps-Tools nahezu vollständig der Open Source entstammen, stellen viele Hosters sie als Managed Services zur Verfügung. Ganz auf Entwicklerwerkzeuge hat sich Cusy spezialisiert. Alle Tools lassen sich einzeln als virtuelle Maschinen buchen, können aber über ein integriertes LDAP-Directory zusammenarbeiten (siehe Abbildung 3).

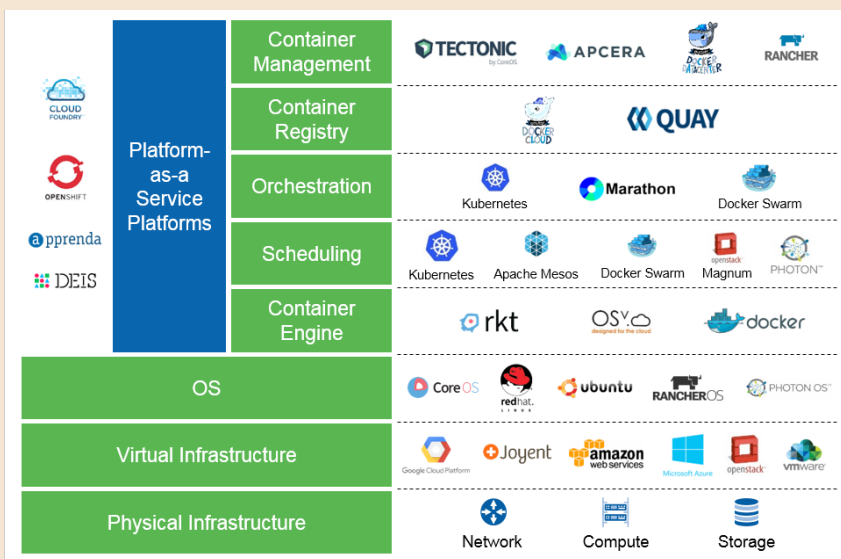
nach und nach mit Cloud-Infrastrukturen.

Wer wissen will, wo die Daten gespeichert werden, muss die Dokumentation der Anbieter sorgfältig lesen. Die Container bei Giant Swarm liegen in Amazons Cloud und bei Loodse auch auf Servern von Digital-

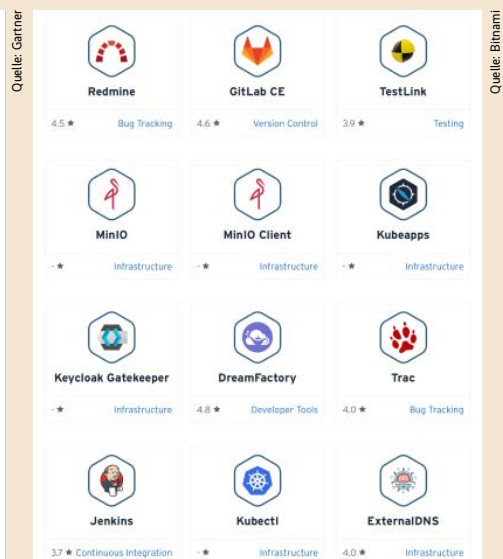
Ocean, während Netways und sloppy.io eigene Server betreiben. Ionos – so der neue Name der 1&1-Hosting-Sparte – hostet seine Containercluster in eigenen Rechenzentren und lässt den Kunden die Wahl der Standorte. Mittels Cloud Panel lassen sich Anwendungen über

mehrere Server (Nodes) hinweg verteilen, deren Nutzung minutengenau abgerechnet wird. Mindestens drei Nodes bilden einen Cluster und lassen sich frei über die IONOS-Rechenzentren in Deutschland, Großbritannien und den USA verteilen.

Auch wenn Docker inzwischen zum vorherrschenden Standard geworden ist, existiert eine Reihe anderer Containerprodukte, etwa Photon von VMware oder rkt von CoreOS aus der Open Source. Erste Entwickler sehen den Zenit der Docker-Technologie bereits



Rund um Docker hat sich ein eigenes Ökosystem von Werkzeugen entwickelt, die Administratoren großer Installationen unterstützen (Abb. 4).



Fertige Entwickler-Images lassen sich als VM oder Container herunterladen (Abb. 5).

überschritten. Als Nachfolger hoch gehandelt wird Podman, das teilweise Docker-kompatibel ist, aber durch die Nutzung von Fork-Prozessen und den Verzicht auf einen Daemon sicherer sein soll. Das neue Red Hat Enterprise Linux 8 ersetzt Docker bereits durch Podman, SUSE Linux stellt beides bereit.

Einer der Gründe, Docker-Alternativen zu entwickeln, ist die Unsicherheit um die Firma Docker Inc., Erfinder und Lizenzhalter der Docker-Entwicklungen. Ende 2019 wurde der Geschäftsbereich für Unternehmenskunden an den Cloud-Dienstleister Mirantis verkauft. Der Wechsel betrifft neben 1000 Enterprise-Kunden und 300 Mitarbeitern die Docker-Enterprise-Engine, Teile der Registry und einige Tools – darunter die Orchestrierungsplattform Docker Swarm, die in den nächsten beiden Jahren infolge der allgemeinen „Kubernetesisierung“ eingestellt wird.

Quelle: [81]

Container-as-a-Service-Plattformen im Vergleich			
	Amazon	Google	Microsoft
CaaS	Amazon EC2 Container Service (ECS)	Google Container Engine (GKE)	Microsoft Azure Container Service (ACS)
technische Grundlage (Rechenleistung)	EC2-Instanzen	Google Compute Engine (GCE)	virtuelle Maschinen
Objektspeicher	Amazon S3 (Simple Storage Service)	Google Cloud Storage	Blob Storage
Orchestrierungstool	proprietäres Verwaltungstool	Kubernetes	Mesosphere DC/OS, Docker Swarm oder Azure Kubernetes Service (AKS)
Verwaltung der Containerplattform	ECS-GUI und CLI	Kubernetes-Dashboard und Google Cloud (Cluster-Management)	Docker-Tools und Azure Cluster-Management
Containerformat	Docker-Container	Docker-Container	Docker-Container
Netzwerkfunktionen	kein Overlay Networking via libnetwork	Networking auf Basis von Kubernetes	Overlay-Networking via libnetwork und Container Network Model (CNM)
Integration in externe Speichersysteme	Docker-Volume-Support	Kubernetes Persistent Volume	Docker Volume Driver via Azure File Storage
Registry	Docker Registry, Amazon EC2 Container Registry	Docker Registry, Google Container Registry	Docker Registry, Azure Container Registry
Identity- und Access-Management	ja	ja	ja
Hybrid-Cloud-Unterstützung	nein	ja	ja
Logging und Monitoring	ja	ja	ja
automatische Skalierung	ja	ja	ja
Schnittstellen zu anderen Cloud-Diensten	Elastic Load Balancing, Elastic Block Store, Virtual Private Cloud, AWS IAM, AWS CloudTrail, AWS CloudFormation	Cloud IAM, Stackdriver Monitoring, Stackdriver Logging, Container Builder	Azure-Portal, Azure Resource Manager (ARM), Azure Active Directory, Azure Stack, Microsoft Operations Management Suite (OMS)

Development

Zahlreiche Tools und Anwendungen für Entwickler und IT Spezialisten.



<p>» Elasticsearch Server</p> <p>Enterprise Suchfunktionalität für höchste Ansprüche</p> <p>» mehr</p>	<p>» GitLab Server</p> <p>Das beliebteste Versionskontrollsystem mit einer wirklich benutzerfreundlichen Oberfläche</p> <p>» mehr</p>	<p>» Grails Plattform</p> <p>Eine optimierte Plattform für Ihre Groovy Eigenentwicklung</p> <p>» mehr</p>
<p>» Java Plattform</p> <p>Betreiben Sie mühelos Ihre professionelle Java-Applikation</p> <p>» mehr</p>	<p>» osticket</p> <p>osTicket ist ein populäres und beliebtes Open Source Support Ticket System</p> <p>» mehr</p>	<p>» OTRS Community Edition</p> <p>Die umfangreiche Open Source Support und Ticketing Lösung</p> <p>» mehr</p>
<p>» PHP Plattform</p> <p>Eine leistungsfähige PHP Plattform für anspruchsvolle Eigenentwicklungen</p> <p>» mehr</p>	<p>» Ruby on Rails Plattform</p> <p>Entwickeln Sie high-end Anwendungen mit Ruby on Rails</p> <p>» mehr</p>	<p>» SaaS Web Mail Gateway</p> <p>Automatisiert mit Ihrer gehosteten Anwendung E-Mails zustellen</p> <p>» mehr</p>

Container als Universalwerkzeuge

Die Vorteile der Containertechnologie lesen sich, als ob die Quadratur des Kreises gelungen sei: schnellere Entwicklung und bessere Testmöglichkeiten, aber auch einfacherer Betrieb und weniger Fehlerquellen. Dem stehen natürlich Limitierungen gegenüber. Nicht alle Aufgaben lassen sich in Microservices zerlegen. Virtuelle Maschinen mit einem eigenen Betriebssystem-Kernel isolieren die Anwendungen besser voneinander, und umfangreiche Anwendungen skalieren besser auf dedizierter Hardware.

Container werden die Virtualisierung somit nicht ersetzen, sondern eher für spezifische Aufgaben ergänzen. Während sich Virtualisierung auf die Infrastruktur bezieht und Server, Storage und Netz als Services bereitstellt, sind Container anwendungsorientiert und kapseln einzelne Funktionen.

Der Betrieb einer größeren Anzahl von Containern ist inzwischen gut möglich; Orchestrierungswerkzeuge unterstützen sowohl die Automatisierung wiederkehrender Aufgaben als auch eine übersichtliche Administration mittels grafischer Oberflächen. Was Container zu leisten vermögen, demonstriert Netflix. Nach eigenen Angaben

Softwareentwicklung as a Service: Programmierwerkzeuge stehen auch als Managed Applications zur Verfügung (Abb. 6).

Gegenüber virtuellen Maschinen punktet Containervirtualisierung nicht nur durch bessere Performance und geringere Ressourcenanforderungen. Als Schnittstellenstandard erlaubt Docker auch die einfache Portierung von Anwendungen und Daten zwischen heimischem Rechenzentrum und Providern sowie zwischen verschiedenen Plattformen, die Docker unterstützen. Entsprechend ihrer Linux-Herkunft können Docker-Container recht einfach auf der Kommandozeile verwaltet und durch Skripte gesteuert werden. Inzwischen stehen aber auch grafische Oberflächen zur Verfügung, etwa DockStation, Portainer oder (für Windows und macOS) Kitematic, das inzwischen Teil der Docker-Toolbox ist. Besonders Portainer ist einen Blick wert, da es selbst in einem Container läuft.

Aufgrund seiner Einfachheit ist rund um Docker ein ganzes Open-Source-Ökosystem entstanden (siehe Abbildung 4). So sind im Internet vorkonfigurierte Docker-Images verfügbar, die die Softwareentwicklung enorm beschleunigen. Als Bei-

spiele seien Bitnami, Turnkey Linux und LinuxServer.io genannt, die als Plattformen für VM-Images gestartet sind und sich inzwischen auch für Container eignen. Viele Anwendungen stehen sowohl als virtuelle Maschinen als auch als Container zur Verfügung, was den Umstieg oder Tests erleichtert.

Bitnami gehört mittlerweile zu VMware und bietet neben dem Download von Images auch die Möglichkeit der Instal-

lation in einer Cloud-Umgebung (siehe Abbildung 5). Zur Auswahl stehen AWS, Google Cloud, Microsoft Azure und Oracle. Eine dedizierte Softwareinstanz widerspricht zwar dem ursprünglichen Cloud-Paradigma der geteilten Ressourcen, passt aber ins Bild der Konvergenz von Diensten, die auf ganz unterschiedliche Weise produziert werden: on Premises, Hosted, as a Service und in der Cloud.

Docker-Container lassen sich mithilfe einer grafischen Oberfläche anlegen und verwalten (Abb. 7).

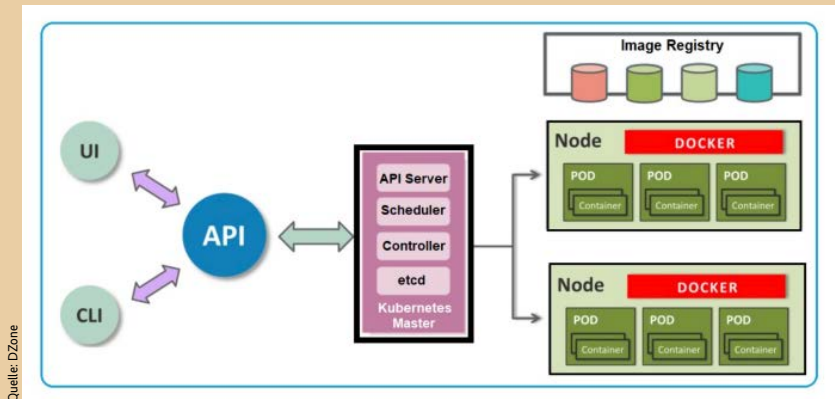
Kubernetes – der Flottenmanager

Kubernetes (griechisch für Steuermann) ist ein Orchestrierungssystem, das den automatisierten Betrieb von Linux-Containern ermöglicht. Es stammt ursprünglich von Google und wird seit 5 Jahren von der Cloud Native Computing Foundation (CNCF) unter Open-Source-Lizenz weiterentwickelt. Kubernetes dient als Plattform zum Bereitstellen, Skalieren und Betreiben von Anwendungscontainern. So lässt sich eine große Anzahl an Containern nach vordefinierten Regeln über verschiedene Infrastrukturen verteilen und managen, beispielsweise im eigenen Rechenzentrum, beim Host und in der Cloud. Kubernetes lässt sich sowohl über Skripte als auch über die grafische Weboberfläche bedienen. Dadurch erübrigen sich viele manuelle Prozesse; Verfügbarkeit und Skalierbarkeit steigen.

Kubernetes ermöglicht die vollständige Verwaltung des Lebenszyklus containerisierter Anwendungen. Administratoren können festlegen, wie diese ausgeführt werden und miteinander interagieren. Es steuert auch die Netz-anbindung durch Verteilung von IP-Adressen und DNS-Namen für eine Gruppe an Containern. So kann das Load Balancing automatisiert erfolgen.

Eine weitere Aufgabe von Kubernetes besteht im Neustart ausgefallener Container und in der Verlagerung von Workloads bei

Mit Kubernetes lassen sich Tausende von Docker-Instanzen verwalten (Abb. 8).



Quelle: DZone

Störungen von Knoten. Je nach CPU-Auslastung und Workload lassen sich zusätzliche Instanzen erzeugen. Darüber hinaus sind Funktionsprüfungen einzelner Applikationen möglich.

Kubernetes ist auf die Orchestrierung großer Mengen von Containern ausgerichtet. Hierzu werden sie in Pods zusammengefasst – einer zusätzlichen Abstraktionsebene, die hilft, Workloads zu planen und die notwendigen Dienste wie Storage und Vernetzung bereitzustellen. Unter einem Pod versteht man eine Gruppe eng gekoppelter Container, die auf derselben Hardware laufen und direkt auf dieselben Ressourcen zugreifen. Alle Container in einem solchen Pod teilen sich IP-Adressen, Ports, Volumes und lokalen Cache (siehe Abbildung 8).

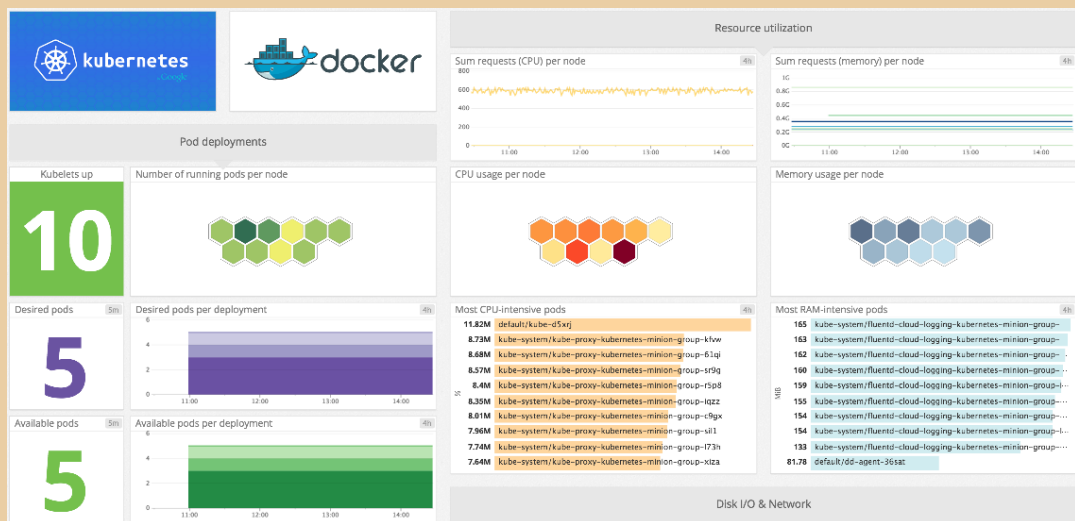
Pods ermöglichen eine Abstrahierung von Netzwerk und Storage und tragen dazu bei, dass

sich Gruppen von Containern im Cluster einfach verschieben lassen. Bei einem Ausfall startet der Replication Controller den Pod automatisch auf einem anderen Host.

Kubernetes verteilt die Last über die Pods und stellt sicher, dass die richtige Anzahl an Containern für die gegebenen Workloads zur Verfügung steht. Zu diesem Zweck kann man definieren, wie viele Instanzen eines Pods jeweils aktiv sein sollen; der Replication Controller kümmert sich um die automatische horizontale Skalierung.

Da Container nur den Programmcode enthalten, werden alle Daten auf klassischem Storage gespeichert. Kubernetes verwaltet hierfür gleichermaßen lokalen Speicher, öffentliche Cloud-Plattformen wie Google Cloud oder AWS, Netzwerk-Filesysteme wie NFS oder Objektspeicher mit Ceph.

Obwohl Kubernetes das beherrschende Orchestrierungstool für Docker-Container ist, existieren auch Alternativen. Die bekanntesten sind Docker Swarm und Amazon ECS. Weniger verbreitet sind Nomad und Consul (bei Neofonie), wobei sich Funktionen und Einsatzszenarien unterscheiden. Ergänzend oder alternativ können auch weitere Automatisierungstools zum Einsatz kommen, beispielsweise Jenkins oder Ansible (Abbildung 9). Umgekehrt ist Kubernetes nicht zwingend auf Docker als Container-Run-time angewiesen, sondern kann auch Alternativen wie Container Runtime Interface (CRI-O) nutzen – vorausgesetzt, sie entsprechen den Vorgaben der Open Container Initiative (OCI). Diese gliedern sich in eine Image- und eine Runtime-Spezifikation und sollen einen offenen Containerstandard als Gegenstück zu Docker etablieren.



Quelle: GitHub

Datadog ermöglicht das Monitoring von Kubernetes-Installationen in Echtzeit (Abb. 9).

Deutsche Hosting-Provider mit Containerservices und anderen Produkten für Entwickler*

Hoster	Website	Produkt
1&1 IONOS	www.ionos.de	Container-Cluster, Managed Kubernetes
b'nerd	bnerd.com	Managed Kubernetes
Centron	www.centron.de	Container-Hosting Takeoff / Economy / First Class
Claranet	www.claranet.de	Docker-Container, Kubernetes, DevOps und ITIL
Cusy	www.cusy.io	DevOps-Toolchain as a Service
Datis	www.datis.de	Container-Hosting
Dembach Goo Informatik	www.dg-i.net	DG-i Container-Plattform (DCP)
Dogado	www.dogado.de	Jelastic Cloud-Hosting: Cloud-Umgebung für Docker, Java-, PHP-, Node.js-, RUBY-Applikationen und VMs
First Colo	www.first-colo.netwww.accelerated.de	Kubernetes as a Service
Giant Swarm	giantswarm.io	Managed Kubernetes
Gridscale	gridscale.io	Plattform as a Service
Hetzner	www.hetzner.com	Dedicated oder Managed Server für Docker
hostNET	www.hostnet.de	Docker-Hosting-Scene, Hosting-Scenes für VMs
Hostserver	www.hostserver.de	Docker-Server
iNNOVO Cloud	www.innovo-cloud.de	Kubernetes as a Service, OpenStack as a Service
InterNetX	www.internetx.com	Kubernetes
I.T.E.N.O.S.	www.itenos.de	PaaS (Containervirtualisierung)
KAMP	www.kamp.de	Kamp Virtual-Core
Laendle Hosting	www.laendle-hosting.de	Docker-Hosting
Leaseweb	www.leaseweb.com	Apache CloudStack mit Container Service
LCube	www.lcube-webhosting.de	Jenkins Server, GitLab Server
levigo systems	hosting.levigo.de	Container-Hosting
Loodse	loodse.com	Kubernatic Container Engine (Cloud oder on Premises)
managedhosting.de	www.managedhosting.de	PaaS for Cloud Native Apps
maxcluster	www.maxcluster.de	Entwicklertools (Monitoring- und Profiling-Tools)
Mittwald	www.mittwald.de	SPACES (basiert auf Kubernetes)
mobilitics	www.mobilitics.de	Docker-Container als App
mpex	www.mpex.de	Managed Hosting (Docker ready)
Neofonie	www.neofonie.de	Docker-Schulung und Projektierung, Hosting in Zusammenarbeit mit mpex und GiantSwarm, Docker-Orchestrierung mit Consul
Nepomuk	www.nepomuk.software	Kubernetes Starter / Produktion / Custom
Netways	www.netways.de	Kubernetes as a Service, Entwicklertools als Managed Service (GitLab, Puppet)
Nordcloud	nordcloud.com	Kubernetes, DevOps
Noris	www.noris.de	Docker-Container, Managed Kubernetes, OpenShift
OVH	www.ovh.de	Container und Orchestrierung
Pivotal	www.pivotal.io	Pivotal Container Service
PlusServer	www.plusserver.de	Managed Kubernetes, OpenShift
ProtoSoft	www.protosoft.de	Docker-Container-Hosting (CaaS)
root360	www.root360.de	Managed Docker und Kubernetes
SaaS-Secure.com	de.saas-secure.com	Projekt- und Softwaremanagement (Redmine, Git, SVN)
SaaS Web Internet Solutions	www.saasweb.net	Managed Application Hosting (Development)
ScaleUp	www.scaleuptech.com	Managed Kubernetes
servinga	www.servinga.com	Kubernetes Hosting
sloppy.io	sloppy.io	Docker-Hosting, Managed Docker-Hosting
SysEleven	www.syseleven.de	MetaKube: Kubernetes as a Service, DevOps-Automatisierung
Strato	www.strato.de	Dedicated Linux-Server, ServerCloud und V-Server Windows sind Docker ready
Telekom Deutschland	cloud.telekom.de	AppAgile PaaS, Cloud Container Engine (CCE)
teuto.net	www.teutostack.de	Managed Kubernetes
UBL	www.ubl-is.de	Container as a Service
Uptime Informations-Technologie	www.uptime.de	Kubernetes as a Service
VPS2day.com	www.vps2day.com	Kubernetes Hosting
Windcloud	www.windcloud.org	Managed Kubernetes

*basierend auf den Angaben der Anbieter, ohne Anspruch auf Vollständigkeit

startet der Streaminganbieter täglich eine halbe Million Container und 200 000 Cluster. Zu diesem Zweck hat das Unternehmen eigens die Containermanagementplattform Titus entwickelt, die im Quellcode auf GitHub bereitsteht (siehe Kasten zu Kubernetes).

Hoster stellen Entwicklungswerkzeuge in ganz unterschiedlicher Form zur Verfügung. Auf dedizierten oder virtuellen Servern können die Kunden sie selbst installieren und betreiben. Zur Unterstützung liefert Hetzner eine Anleitung zum Installieren von Docker auf CentOS.

Viele Softwarepakete stehen auch vorinstalliert oder vorkonfiguriert zur Verfügung (siehe Abbildung 6). Noch einfacher nutzbar ist ein Managed Service oder SaaS, wie ihn Netways für Kubernetes und spezielle Entwicklertools wie GitLab anbietet.

Wider den Vendor Lock-in

Docker-Hosting ähnelt dem Angebot der Virtual Private Server (VPS): Der Hoster betreibt ein Portal, über das Kunden Software-Images hochladen und starten. Trotz ähnlichen Funktionsumfangs unterscheiden sich die Oberflächen von Anbietern. Giant Swarm und sloppy.io bieten neben einem WebGUI auch ein Command Line Interface (CLI) an, was besonders Admins und Entwickler zu schätzen wissen. Loodse setzt auf die haus-eigene Software Kubernatic und Dogado nutzt die Tools und die Oberfläche der kalifornischen Firma Jelastic (Abbildung 7).

Viele Hosting-Angebote sind ganz auf Softwareentwickler ausgerichtet. So werben die Provider damit, dass nur die wirklich genutzten Ressourcen zu bezahlen sind und nicht die reservierten. Einrichtungsarbeiten sind in der Regel nicht nötig und die Vertragslaufzeiten flexibel. Hoster stellen sich auch darauf ein, dass Entwick-

Glossar: Docker-Container

Ein **Image** ist eine Datei, die einen inaktiven, statischen (read-only) Snapshot enthält, aus dem sich laufende Containerinstanzen erzeugen lassen. Neben dem Programmcode und allen benötigten Bibliotheken stellt das Image Metadaten zur Verfügung, die Anforderungen und Fähigkeiten der jeweiligen Anwendung beschreiben. Images dienen dazu, Container zu speichern und zu übertragen – beispielsweise zwischen Entwicklungs- und Produktionssystem.

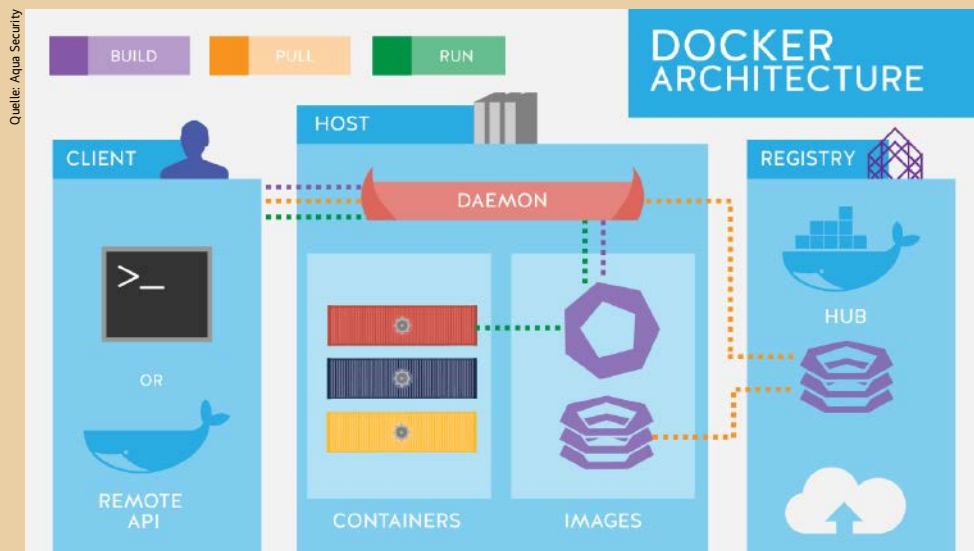
Eine **Registry** speichert als zentrales Software-Repository Images, verwaltet sie und stellt sie zur Ausführung bereit. Unternehmen können eine private Registry implementieren; darüber hinaus stellen Cloud-Registries Images weltweit zur Verfügung. Die größte öffentliche Bibliothek ist Docker Hub mit mehr als 100 000 Images von kommerziellen Softwareentwicklern und aus der Open Source. Sie wird von Docker Inc. betrieben, erhält aber immer mehr Konkurrenz von anderen öffentlichen Registries wie der Entwicklerplattform GitHub.

Die **Docker Engine** ist eine Laufzeitumgebung für die einheitliche Ausführung von Docker-Containern. Der Kernprozess hierfür ist der Docker-Daemon, der Images, Netzressourcen und Storage Volumes verwaltet. Neben dem Daemon zum Ausführen des Programmcodes stellt die Docker-Engine auch eine REST-API und ein Command Line Interface (CLI) zur Kommunikation mit Applika-

tionen zur Verfügung. Bei Docker Inc. gibt es die Docker Engine in einer kostenpflichtigen Enterprise- und einer freien Community-Version.

Eine Instanz der Docker Engine, der **Docker Host**, kann mit anderen Daemons kommunizieren, um eine über mehrere Sites verteilte Installation als einheitliches Ganzes darzustellen. Beim Start eines

Containers erzeugt der Docker-Host eine Laufzeitumgebung in Form von Instruktionen, die im Build File hinterlegt sind. Sie berücksichtigt Komponenten, die im Vorfeld bereitstehen müssen, sowie zusätzliche Befehle für den Start des Containers. Hierfür ist keine Handarbeit nötig. Der Docker Host kann sich benötigte Images direkt von einer vorgeinstellten Registry herunterladen.



Mit Docker lassen sich Anwendungen so einfach wie Dateien bereitstellen (Abb. 10).

ler andere Anforderungen an Vertragswerke und Zahlungsmöglichkeiten stellen als die klassischen IT-Abteilungen: Bei sloppy.io genügen eine E-Mail-Adresse und eine Kreditkarte zum Starten von Containern; VPS2day erlaubt die Bezahlung per Bitcoin, Litecoin oder Altcoin.

Darüber hinaus stellen Hoster dedizierte Containerlaufzeitumgebungen für einzelne Kunden zur Verfügung. So gliedert Centron sein Angebot in drei Pakete mit bis zu 300 Containern, die über unterschiedlich viele Prozesskerne und Speicherausstattung verfügen. Noris lässt den Kunden die Wahl, Managed Kubernetes Cluster als Public Infrastructure in den Noris-Rechenzentren zu nutzen oder eine kundenspezifische Plattform mit eigenen Servern aufzubauen. Einige Provider wollen Preise

nur auf Anfrage nennen. Sie zielen damit eher auf Produktivumgebungen großer Kunden.

Auch wenn sich die Oberflächen von Hoster zu Hoster unterscheiden (siehe Tabelle mit

der Marktübersicht), wirken Docker und Kubernetes als Open-Source-Werkzeuge dem Vendor Lock-in entgegen. Sollte eines Tages ein Wechsel des Providers anstehen, wird

dies also zu einem vergleichsweise einfachen Unterfangen. (un@ix.de)

Uwe Schulze
ist freier Autor in Berlin.

In iX extra 6/2020

Cloud-Computing: Kubernetes auf Expansionskurs

Die Containerorchestrierungstechnologie Kubernetes hat sich unglaublich schnell von der internen Verwendung bei Google über die Open-Source-Inkubation in der Cloud Native Computing Foundation zum unverzichtbaren Standard bei den namhaften Betreibern von Cloud-Infrastrukturen entwickelt.

Als ausgereifte Technologie befindet sich Kubernetes derzeit in einer neuen Phase wachsender Akzeptanz bei Unternehmen aller Art.

Aber Adoptionstrends zeichnen nur einen Teil des Kubernetes-Bildes. Die Frage für CIOs, die sich für eine Strategie zur digitalen Transformation entscheiden müssen, lautet nicht, ob Kubernetes als Rückgrat der Cloud-nativen Infrastruktur ihres Unternehmens dienen soll, sondern wie, wie schnell und mit wem.

Start-ups, Open-Source-Initiativen und die großen IT-Konzerne tragen gemeinsam zu einem umfassenden

Ökosystem bei, das sich um das Kernprojekt von Kubernetes dreht und die vielfältigen Anforderungen und Einschränkungen all seiner potenziellen Kunden erfüllen kann. Die Marktübersicht gibt einen Überblick über die wichtigsten Teilnehmer und Konzepte.

Weitere iX extras erscheinen in der iX 10/2020 (Security: Trends und Produkte zur it-sa) sowie in der iX 11/2020 (Hosting: Managed Services).