

SCHNELLER, BESSER, DOCKER!

Begriffe wie Docker und Container sind dir schon einmal über den Weg gelaufen, aber du fragst dich - worum geht es hier eigentlich genau? Zu Beginn kann das alles etwas kompliziert wirken, aber keine Sorge, wir antworten Schritt für Schritt auf einige Fragen.

Was es mit Containern auf sich hat, weshalb Docker als isolierte Applikation die bessere Alternative zu virtuellen Maschinen ist und warum du auf Ressourcen sparende Möglichkeiten bei Hosting- und Cloudanwendungen setzen solltest, erfährst du in diesem Blog.

DAS Open-Source-Programm für einen schnelleren Start und bessere Sicherheit

Unsere Welt ist softwaregesteuert und eine einfache Bereitstellung von Software- Anwendungen zu minimalen Kosten ist eine wichtige Anforderung. Tech-Giganten wie Google, Facebook oder Microsoft nutzen Container, um eine sichere Bereitstellung Ihrer Prozesse in den Cloud-Produktionsumgebungen zu ermöglichen. Mit der Softwareplattform Docker bietet sich eine Technik, die die Anwendung von einer Laufzeitumgebung abstrahiert. Zur Laufzeitumgebung gehören Laufzeitbibliothek, Standardbibliotheken, Programmierschnittstellen, Laufzeitvariablen sowie Hard- und Softwarekomponente über Betriebssystemfunktionen. Ähnlich wie bei virtuellen Maschinen, geschieht das durch Virtualisierung.

Docker ist eine Reihe von Platform-as-a-Service (PaaS)-Produkten. Als Platform-as-a-Service bezeichnet man eine Dienstleistung, die in der Cloud eine Computer-Plattform für Entwickler von Webanwendungen zur Verfügung stellt. Es kann sich dabei sowohl um schnell einsetzbare Laufzeitumgebungen (typisch für Webanwendungen), aber auch um Entwicklungsumgebungen handeln. Diese werden mit geringem administrativem Aufwand und ohne Anschaffung der darunterliegenden Hardware und Software genutzt, die sogenannten Container. Diese unterstützen den gesamten Software-Lebenszyklus vom Design über die Entwicklung, den Test, die Auslieferung bis hin zum Betrieb der Webanwendung über das Internet und den dafür ausgerichteten Computersystemen.

Docker makes development efficient and predictable

Docker takes away repetitive, mundane configuration tasks and is used throughout the development lifecycle for fast, easy and portable application development – desktop and cloud. Docker's comprehensive and to end platform includes UIs, CLIs, APIs and security that are engineered to work together across the entire application delivery lifecycle.



Container sind eine einfache Lösung, um Anwendungen in Ihren entsprechenden Umgebungen dynamisch bereitzustellen und zu übertragen, ganz ohne Probleme bei der Kompatibilität zu bekommen. Ebenso sind sie zustandslos, voneinander isoliert und persistieren keine Daten, falls dies nicht gewünscht ist.

Container - Verpackung von Microservices mit klaren Vorteilen

Um Anwendungen in einer losen Umgebung (z.B. Laufzeitumgebung oder Entwicklungsumgebung) zu isolieren, arbeitet Docker mit einem Image-basierten Bereitstellungsmodell. So wird es möglich, eine Anwendung oder ein Paket von Services mit all deren Abhängigkeiten gemeinsam in mehreren Umgebungen zu nutzen.

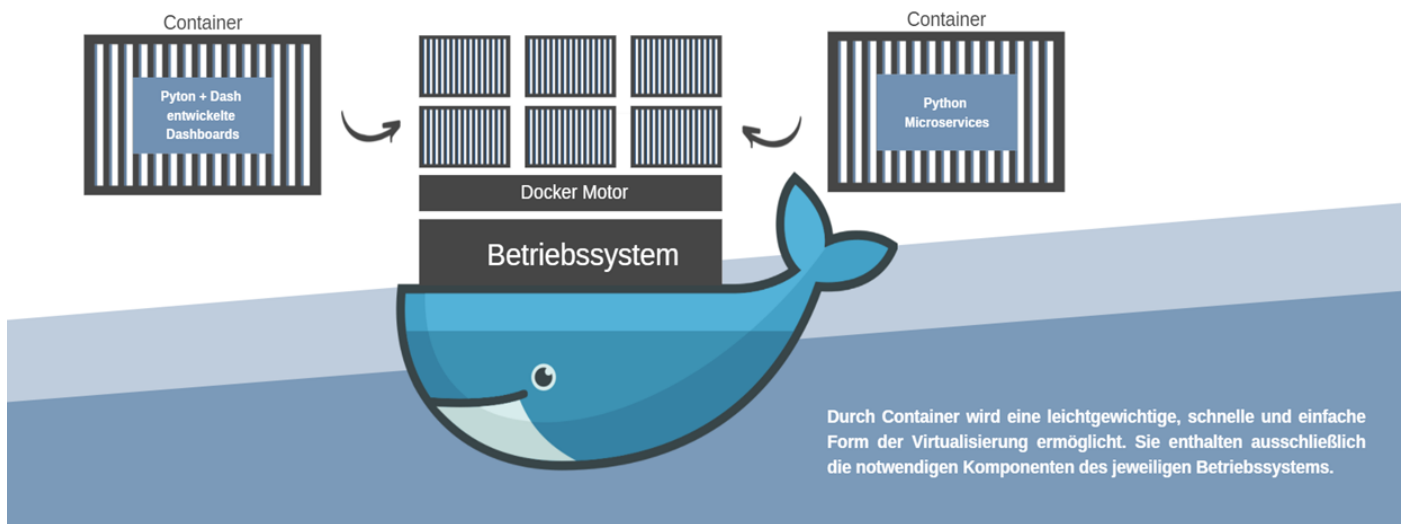
Einfach erklärt: **Der Zweck eines Containers ist die Fähigkeit, mehrere Prozesse und Apps getrennt voneinander sowie getrennt vom Host-System, betreiben zu können.** Daraus ergibt sich mehr Sicherheit bei der Nutzung und ohne die Applikation zu unterbrechen, wird die Infrastruktur gleichzeitig besser und Ressourcen sparend genutzt. Mit Ressourcenkontingenten ist es sogar möglich, die Menge an Arbeitsspeicher zu begrenzen, die ein Container aufnehmen kann. Dies verbessert die Effizienz der Umgebung und verhindert außerdem ein Ungleichgewicht bei der Verteilung dieser Ressourcen auf ihre Umgebung.

Anwendungen können mit einem deutlich verringerten Zeitaufwand bereitgestellt werden und aufeinanderfolgende Versionen eines Containers lassen sich sogar nachverfolgen. Unterschiede können untersucht werden und zudem kann auf vorherige Versionen zurückgegriffen werden.

Auf den Punkt gebracht - Vorteile der Docker-Anwendung:

- Schnelle Anwendungsbereitstellung
- Portabilität auf allen Maschinen
- Einheitliche Umgebung
- Versionskontrolle und Wiederverwendung von Komponenten
- Leichtgewichtigkeit und minimaler Aufwand bei Rollouts
- Einfachere Wartung einzelner Module
- Hohe Sicherheit bei der Anwendung
- Optimale Nutzung der Infrastruktur
- Registry: The Registry ist eine zustandslose, hoch skalierbare serverseitige Anwendung, die Docker-Images speichert und verteilt.
Die Registry ist Open-Source und steht unter der Apache-Lizenz.

Fun-Fact: Der Begriff Container ist tatsächlich an den physischen Container, wie wir ihn zum Beispiel von Schiffen kennen, angelehnt. Von außen betrachtet, ist ein Schiffscontainer mit standardisierten Schnittstellen versehen und kann dadurch auf jedem dafür ausgerichteten Schiff befördert werden – unabhängig von seinem Inhalt. Genau so läuft das bei unseren Docker-Containern auch ab. Abgegrenzt vom Inhalt, kann der Container somit auf jedem dafür ausgerichteten Computersystem ausgeführt werden.



Docker Container in der Anwendung inklusive Betriebssystem, Docker Motor und den entsprechenden Applikationen

Die Alternativen zu Docker? Wir möchten nicht behaupten, dass wir keine wirklichen Alternativen kennen, denn für Entwickler gibt es zum Beispiel virtuelle Python Environments. Diese decken jedoch nur einen kleinen Teil von dem ab, was uns Docker bieten kann. Docker als Webanwendung macht unsere Arbeit effizient und vorhersagbar.

Deine ersten Docker-Basics hast du jetzt hinter dir. Selbstverständlich handelt es sich hier um ein sehr umfangreiches Thema, schau dir daher gerne unseren Artikel **Docker vs. virtuelle Maschine** an. In diesem Blog hast du eine Übersicht darüber bekommen was Docker ist, wofür wir Container nutzen und was der große Vorteil dieses Plattform-as-a-Service Produktes ist.

Hast du noch Fragen, Anregungen oder möchtest einfach Hallo sagen? Kontaktiere uns gerne über Email: info@supperundsupper.com

