

Boost your APEX Deployment and Provisioning with Docker

Steven Grzbielok
MT AG
Frankfurt am Main

Schlüsselworte

APEX, Docker, Virtualisierung, Deployment

Einleitung

Docker ist eine Virtualisierungsplattform mit dessen Hilfe sich Linux Container einfach erstellen und verwalten lassen. Docker soll vor allem durch eine optimierte Architektur sowie bessere Performance, Skalierbarkeit und Portabilität neue Möglichkeiten hinsichtlich virtueller Serverlandschaften ermöglichen. Dieser Vortrag beleuchtet die Möglichkeiten, die für APEX in Verbindung mit Docker gegeben sind und zeigt mögliche Arten der Integration für diese beiden Technologien auf.

Grundlagen

Warum mit Docker beschäftigen?

Docker setzt auf eine Microservice-Architektur. Die Verwendung einer Microservice Architektur macht Software robuster, da über Sprachenunabhängige Schnittstellen kommuniziert wird und die einzelnen Microservices unabhängig und weitestgehend entkoppelt voneinander sind. Durch diese Architektur fällt der Einstieg in Continuous Delivery leichter, da einzelne Microservices einzeln in die Produktion überführt werden können. Docker verpackt einzelne Microservices in Container und somit in modulare virtuelle Umgebungen. Diese Container zeichnen sich besonders durch ihr einfaches Deployment aus und somit lässt sich die gesamte Umgebung leicht auf einen anderen Host deployen. Des Weiteren ist Docker ebenfalls für umfangreiche und skalierbare Serverlandschaften relevant, mit deren Hilfe auch die Performance einer APEX Anwendung bei hoher Auslastung profitieren kann.

Virtualisierung

Virtualisierung beschreibt die Unterteilung eines physischen Systems in verschiedene kleine Systeme durch die Einführung virtueller Hardware. Die Systemumgebung wird abstrahiert für das virtuelle System, sodass die virtuelle Maschine unabhängig von der konkreten zugrundeliegenden Hardware betrieben wird. So laufen virtuelle Maschinen meist innerhalb eines normalen Betriebssystems und werden von diesem gehostet. Durch die Unabhängigkeit von der Hardware wird die virtuelle Maschine skalierbar und mobil. Die verfügbaren Ressourcen können dynamisch erhöht oder verringert werden und die virtuelle Maschine kann jederzeit auf einem anderen physischen System gehostet werden.

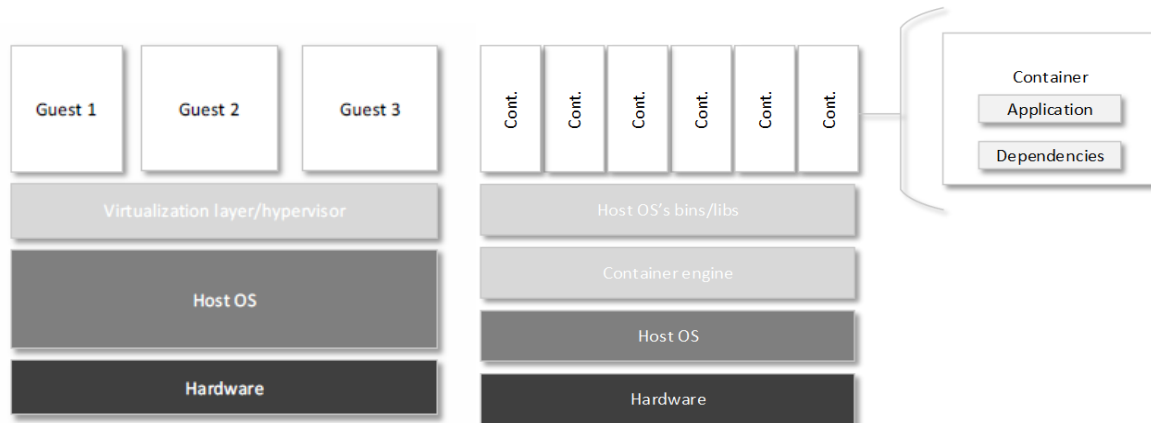


Abb. 1: Container als neue Architektur bei der Virtualisierung

Während bei der herkömmlichen Virtualisierung das gesamte Betriebssystem der virtuellen Maschine virtuell gehostet wird, verfolgen Container einen anderen Ansatz. Bei Containern wird lediglich ein virtuelles Betriebssystem gehostet. Virtuelle Systeme werden hier als Container geladen und teilen sich das selbe Betriebssystem bzw. dessen Kernel. Die Virtualisierungsplattform verwaltet die verschiedenen Abhängigkeiten und Berechtigungen durch Features wie z.B. Namespaces. Die zugeteilten Ressourcen je Container können mit Controlgroups verwaltet werden. Zu beachten ist, dass die Container mit dem virtuellen Betriebssystem übereinstimmen müssen, d.h. sie benötigen den selben Kernel. Bspw. ein Ubuntu kann nur auf einem Linux System gehostet werden. Um dennoch einen Ubuntu Container unter Windows zu hosten, wird ein virtuelles Linux System benötigt.

Docker

Docker ist eine der bekanntesten Virtualisierungsplattformen und verspricht besonders einfache Handhabung und erleichtert durch verschiedene Features die Arbeit der Entwickler. Zu Beginn unterstützte Docker lediglich Linux Container, jedoch werden durch die Zusammenarbeit von Docker und Microsoft auch Windows Container unter Windows Server 2016, Windows 10 und Microsoft Azure möglich.

Eines der Hauptargumente, das für Docker spricht, ist das Docker Hub. Hier können Container öffentlich oder privat hochgeladen und geteilt werden. So sind für viele Anwendungsfälle bereits offizielle Container oder von der Community erstellte Container online verfügbar und können mittels eines einfachen Kommandos heruntergeladen werden.

Docker und APEX – Wie passt das?

Aktuell erhalten Docker Container von Oracle keinen Support. Somit ist zu beachten, dass jegliche Lösungen auf eigene Gefahr genutzt werden. Da sich Oracle jedoch zunehmend auch mit Docker beschäftigt, ist nicht auszuschließen, dass Docker Container in Zukunft ebenfalls den Oracle Support erhalten.

Pull & Run - Getting started

Docker ist besonders geeignet für das schnelle Aufsetzen einer Test- oder Entwicklungsumgebung. In dem Docker Hub kann beispielsweise ein vorkonfiguriertes Image mit Oracle Datenbank, Oracle Application Express 5.0.3 und ORDS heruntergeladen werden. Dies bedarf lediglich eines Accounts im Docker Hub und des Pull Befehls in der Kommandozeile.

```
docker pull araczkowski/oracle-apex-ords
```

Der o.g. Befehl lässt Docker ein öffentliches Image, das in der Community entwickelt wurde, herunterladen. Mittels des Run Befehls wird der Container anschließend gestartet und die Ports werden entsprechend zugewiesen. Nach einigen Momenten ist der Container „Up&Running“ und die Datenbank sowie der ORDS fahren hoch.

```
docker run -d --name ora -p 49160:22 -p 8080:8080 -p 1521:1521  
araczkowski/oracle-apex-ords
```

Build - Offizielle Images von Oracle

Neben Community-Images werden außerdem von Software Herstellern diverse vorkonfigurierte Images zur Verfügung gestellt, die direkt eine grundlegende Systemumgebung ermöglichen. Oracle bietet seit Kurzem ebenfalls vorkonfigurierte Images an. Oracle bietet dies jedoch mithilfe der Build Funktion von Docker an, da Oracle nicht die gesamte Datenbank im Docker Hub veröffentlicht hat. Die Docker Buildfile kann über GitHub heruntergeladen werden und in Kombination mit den offiziellen Binärdateien der Oracle Datenbank genutzt werden. In einem Tutorial erklärt Gerald Venzl von Oracle, wie der Build Prozess durchgeführt werden kann und so ein virtuelles System aufgesetzt wird. Auch hier bietet Oracle zwar keinen Support an, jedoch wird so lediglich offizielle Software verwendet.

Die genannten Images von Oracle lassen sich problemlos anpassen und um zusätzliche Software erweitern. Ebenfalls können selbst angepasste Container in private Repositories hochgeladen werden, um fertige Entwicklungsumgebungen innerhalb eines Teams bzw. eines Unternehmens schnell zur Verfügung zu stellen. So kann jeder Entwickler eine eigene virtuelle Entwicklungsumgebung nutzen, die er innerhalb kurzer Zeit durch einen einfachen Pull Befehl aus dem privaten Repository seines Teams herunterladen kann.

Compose – Die eigene Systemlandschaft zusammenstellen

Das Zusatztool Docker Compose ist bereits bei der aktuellsten Version von Docker vorinstalliert. Während Docker Build für das Erstellen einer Systemumgebung mit nur einem Container genutzt werden kann, ist Docker Compose für die Zusammenstellung von Multicontainer-Umgebungen konzipiert. Docker sieht vor, dass Container so klein wie möglich gehalten werden sollen und nach Möglichkeit nur einen Dienst bzw. eine Anwendung enthalten, da erst bei einer derartigen Architektur auch die Vorteile von Docker in Kraft treten können.

Innerhalb des Skripts docker-compose.yml kann hinterlegt werden, welche Anwendungen bzw. welche Container benötigt werden und wie diese konfiguriert sein sollen. Hier können beispielsweise Datenbankports oder Verknüpfungen zwischen den Containern definiert werden, sodass die Anwendungen nicht einzeln von Hand installiert werden müssen. Denkbar wäre z.B. die Konfiguration eines Containers, der die Oracle Datenbank enthält und eines Containers, der den Apache Tomcat sowie den ORDS und die statischen Dateien enthält. Diese Architektur bietet den Vorteil, dass die Container sich zwar als Paket deployen lassen, dennoch aber die Datenbank getrennt vom Webserver ist. Änderungen am Tomcat, dem ORDS bzw. den statischen Dateien können als Gesamtpaket unabhängig von der Datenbank deployt werden.

Besonders relevant wird Docker Compose, wenn APEX mit weiteren Technologien integriert wird: Nutzt APEX beispielsweise einen Node.js Webserver, so kann Node.js in einem separaten Container gehostet werden. Diese Modularisierung bietet vorallem Vorteile für ein unkompliziertes Deployment mit einer möglichen Nutzung von Continuous Integration und die gezielte Nutzung von APIs. Der Node.js Container kann inklusive aller zugehörigen Dateien und Abhängigkeiten deployt werden, ohne dass andere Container beeinflusst werden.

Weitere Szenarien

Erweiterte Docker Compose Szenarien

Mittels Docker Compose kann beispielsweise auch eine Systemarchitektur erstellt werden, bei der die Anwendungslogik in einer getrennten Datenbank gehostet wird, als die Daten. So werden drei Container erstellt:

1. Oracle Datenbank mit APEX und der Anwendungslogik
2. Oracle Datenbank mit den Schemata zu den Daten
3. Webserver, ORDS und statische Dateien

Zwar unterstützt die Oracle Datenbank mittlerweile die Erstellung mehrerer Datenbanken als Pluggable Databases in einer Datenbankumgebung, jedoch wird im Hinblick auf eine Service-orientierte Architektur in Verbindung mit Microservices eine erweiterte Architektur erzielt. Diese Architektur ermöglicht es, einen gesamten Container zu deployen und so die Anwendungsdatenbank bei einer neuen Version zu ersetzen, ohne die Produktivdaten zu beeinflussen.

Docker Swarm für dynamische Skalierung

Docker Swarm kann als ergänzender Ansatz für den Oracle Real Application Cluster genutzt werden. Docker Swarm ist eine Zusatzkomponente von Docker, die für die Orchestrierung von Docker Containern genutzt wird. Load Balancing kann in Docker Swarm implementiert werden, um die Zugriffe auf eine Anwendung bzw. einen Webserver gezielt zu steuern. So ergibt sich hier die Wahl, das integrierte Load Balancing des RAC zu nutzen oder auf einer höheren Ebene direkt durch Docker Swarm die Last gezielt an die Server zu senden. In komplexen Architekturen ist es zudem denkbar, die automatische Skalierung durch Docker Swarm zu implementieren und so dynamisch Nodes in dem RAC hinzuzufügen. Mittels entsprechender Skripte könnte dieser Prozess des Hinzufügens möglicherweise automatisiert werden. Jedoch ist weiterhin zu bedenken, dass bei einem einzelnen Datenbankausfall eine kurze Downtime entsteht, währenddessen der gesamte Cluster nicht erreichbar ist.

Fazit und Ausblick

Aktuell erhält Docker keinen Support von Oracle. Da jedoch Docker zunehmend an Beliebtheit gewinnt, ist es nicht ausgeschlossen, dass in Zukunft auch Docker-Umgebungen den Oracle Support erhalten. Docker bietet jedoch auch heute schon einfache und schnelle Möglichkeiten, eine Entwicklungs- oder Testumgebung schnell aufzusetzen oder z.B. eine lizenzfreie Umgebung mit der Oracle XE und mehreren Datenbanken zu betreiben. Besonders hinsichtlich Portabilität und Skalierbarkeit hat Docker großes Potenzial und wird wohl auch mit der ausstehenden Kooperation mit Microsoft an Marktrelevanz gewinnen können. Umfangreiche und komplexe Einsatzszenarien wie Load Balancing oder Clustering werden durch Zusatzkomponenten wie Docker Swarm möglich und das Deployment kann durch die gezielte Modularisierung in Microservices bereichert werden.

Für APEX bedeutet dies zum einen, dass zusätzliche Software leichter integriert und deployt werden kann, APEX getrennt von den Daten und von dem Webserver in Microservices verpackt werden kann, und somit getrennt deployt oder auch skaliert werden kann. Die Verwendung dieser Microservices ist besonders relevant für Continuous Integration z.B. mit Jenkins bzw. zur Erstellung einer Deployment Pipeline im CD-Umfeld.

Kontaktadresse

Steven Grzbielok
MT AG
Solmsstraße 10
D-60486 Frankfurt am Main

Telefon: +49 (0) 2102 30961-0
Fax: +49 (0) 2102 30961-101
E-Mail: steven.grzbielok@mt-ag.com
Internet: www.mt-ag.com