

Laß das Orchester spielen - Docker Orchestrierung und Visualisierung

Thorsten Wussow
Slix GmbH
Au i. d. Hallertau

Schlüsselworte

Docker, Docker Swarm, Microservice, Cloud, Kubernetes, Mesos, Oracle Container Cloud Service

Einleitung

Nach den ersten Schritten mit Docker und wenn man dann über die produktive Einführung von Microservices nachdenkt, dann stellt sich die Frage, wie man komplexe Systeme die aus mehreren Services und Containern bestehen, wiederholbar aufbauen, diese betreiben und dabei über die Vielzahl der laufenden Services und den bestehenden Verbindungen untereinander die Übersicht behält. Ebenso sollen die Systeme weitestgehend automatisiert Arbeiten und auch Hochverfügbar sein. Auf dem Markt gibt es aktuell verschiedene Produkte dafür, Cloudbasierte (Oracle Container Cloud), sowie für OnPremise-Installationen die den Fokus auf Orchestrierung und Visualisierung haben. In dem Vortrag werden verschiedene System vorgestellt, kurz gezeigt und auf die Besonderheiten eingegangen.

Begriffsklärung

Orchestrierung was bedeutet dieser Begriff genau?

Das meint Wikipedia dazu:

Orchestrierung (englisch *orchestration*, Instrumentierung, Inszenierung) ist das flexible Kombinieren mehrerer Services zu einer Komposition. Diese Komposition beschreibt einen ausführbaren Geschäftsprozess. Sowohl unternehmensinterne als auch unternehmensexterne Dienste können kombiniert werden. Der Prozessfluss wird durch einen Teilnehmer kontrolliert. Jeder Dienst hat dabei einen eingeschränkten Sichtbereich (englisch *scope*) und kann für Prozesse nur innerhalb seines Sichtbereichs entscheiden. Aktivitäten hinter einem direkten Kommunikationspartner bleiben verborgen.

Für Orchestrierung gibt es noch ein weiteres, nicht so gängiges Wort: Dienstekomposition. Dieses Wort beschreibt etwas besser was sich hinter der Orchestrierung verbirgt: Eine Komposition, also Zusammenstellung, von Diensten oder eben Microservices um einen kompletten Prozess abzubilden. Auf den Vortrag bezogen bedeutet dies, ein Tool mit welchem man Microservices welche in einem Dockercontainer laufen, zu einem Prozess (oder für den Anwender nach aussen eine komplette Anwendung) zusammenstellen kann, ohne mit der dahinterliegenden Komplexität konfrontiert zu werden. Von diesen Tools wird aber noch mehr erwartet: Sie sollen auch diese Zusammenstellung von Microservices am laufen halten (Hochverfügbarkeit), dafür sorgen das sie performant läuft (Clustering) und eine grundlegende Überwachung sicherstellen.

Und die Visualisierung? Die bringen viele Orchestrierungstools auch mit. Es wird also der Verbund der Microservices dargestellt, so das man leicht die Zusammenhänge zwischen den einzelnen Services erkennen kann. Es gibt aber auch Tools die nur eine Visualisierung anbieten und die Orchestrierung nur rudimentär beherrschen.

Warum benötigt man ein Tool zur Orchestrierung?

Im Umfeld der Dockercontainer werden die Orchestrierungstools meist als ClusterManager verwendet, um einen Microservice performant und skalierbar in einem Cluster von Dockercontainern zu betreiben. Einen Dockercluster ohne ein solches Tool aufzubauen und am laufen zu halten ist sehr komplex und fehlerbehaftet. Ein solches Tool stellt ein meist einfach gehaltenes Interface zur Bereitstellung und Verwaltung der Cluster bereit und verbirgt die dahinterliegende Komplexität. Die Tools verwenden die in Docker integrierten Möglichkeiten um solche Cluster aufzubauen und zu verwalten.

Wann benötigt man ein Tool zur Orchestrierung?

Will man in einem Unternehmen Microservices auf Basis von Dockercontainern produktiv einführen und diese möglichst automatisiert betreiben, dann sollte man ein solches Tool einsetzen. Dadurch besteht die Möglichkeit den Betrieb zu automatisieren, die Beziehungen zwischen den einzelnen Microservices zu verwalten und man behält den Überblick über Masse der Container welche dann in einem produktiven System laufen.

Marktübersicht

Diese Übersicht stellt nur einen Teil der auf dem Markt befindlichen Umgebungen dar.

Docker Swarm

Docker Swarm ist das in Docker enthaltene Orchestrierungstool. Es beinhaltet einen ClusterManager und entsprechende Möglichkeiten ein Netzwerk und andere Voraussetzungen für den Betrieb eines Serviceclusters einzurichten. Ein Docker-Swarm-Cluster läuft nach dem Master-Slave-Prinzip und besteht aus einem oder mehreren Mastern und einem oder mehreren Workernodes. Diese kommunizieren über ein internes Netzwerk, welches sich auch über mehrere Hosts spannen kann. Der Master steuert die Verteilung der Services auf die Workernodes und sorgt für den Start und Stop der Instanzen. Eine grafische Oberfläche dazu gibt es nicht von Haus aus. Zur Visualisierung und einfacheren Bedienung kann Shipyard oder Portainer verwendet werden (siehe unten)

DEMO

Kubernetes

Kubernetes ist eine von Google betreute open source Orchestrierungsplattform für Docker und rkt Container. Es kann deklarativ konfiguriert werden. Es erfreut sich einer weiten Verbreitung auch bei einigen Cloud-Anbietern, z.B. Openshift, Google Containerengine und Oracle Cloud (siehe unten)

Kubernetes arbeitet im Master-Slave-Prinzip. Ein Cluster besteht also immer aus einem Master und mehreren Nodes. Über den Master kann über eine REST API der Cluster verwaltet und gesteuert werden, ebenso ist der Master für das Starten und die Verteilung der laufenden Container auf die Nodes zuständig (scheduling). Das Tool kubectl dient zum einfachen Zugriff auf die API des Clusters.

Eine grafische Oberfläche ist auch mittels Kubernetes Dashboard enthalten.

DEMO

Marathon (DC/OS)

Marathon ist das Orchestrierungstool für DC/OS oder Apache Mesos. Für Marathon wird ein laufender DC/OS oder Mesos-Cluster benötigt. Mit Marathon können über die Integrierte UI Anwendungscluster erstellt, verwaltet und überwacht werden. Marathon ist Hochverfügbar ausgelegt,

kann mit Docker und Mesos-Containern umgehen. Ebenso stellt das Tool Loadbalancing und Service-Discovery zur Verfügung.

DEMO

Oracle Container Services

Die Container Services von Oracle wurden im September 2017 freigegeben und basieren weitestgehend auf Kubernetes. Aktuell sind noch nicht alle Funktionen in die Oracle API übernommen. Die Container Services arbeiten auch mit der Oracle Container Cloud zusammen und können dort Cluster oder Teile davon verwalten.

DEMO (auch mit Nodes in Cloud)

Portainer

Portainer ist eine grafische Oberfläche zum Verwalten von Docker-Containern oder eines Docker-Swarm. Portainer läuft selbst als Docker-Container. Es ist einfach zu installieren und leichtgewichtig. Es deckt weitestgehend alle Orchestrierungsfunktionen von Docker-Swarm ab.

Shipyards

Shipyards ist eine weitere grafische Oberfläche zum Orchestrieren von Docker-Containern basierend auf Docker-Swarm. Shipyards kommt selbst in einem Docker-Container und benötigt eine Docker-Engine zum Laufen.

DEMO

Kontaktadresse:

Thorsten Wussow
Slix GmbH
Nandlstädter Weg, 6
D-84072 Au i. d. Hallertau

Telefon: +49 (0) 12-345 6789
Fax: +49 (0) 12-345 6788
E-Mail: thorsten.wussow@slix.de
Internet: www.slix.de