

# Bereitstellung von Microservice mit dem OCCS

**Stefan Kühnlein**  
**OPITZ CONSULTING GmbH**  
**München**

## **Schlüsselworte**

Cloud, Microservice, Docker, Container

## **Einleitung**

Microservices sind längst nicht mehr nur ein Buzzword, sondern etablieren sich immer mehr in aktuellen IT-Vorhaben. Sie bilden inzwischen die Grundlage flexibler und robuster Softwarearchitekturen und viele monolithische Anwendungen werden durch eigenständige, fachliche Microservices ersetzt.

Insbesondere durch den Einsatz von Container-Technologien wird die Verwaltung und Verteilung von Microservices wesentlich vereinfacht. Container-Technologien wie Docker und rücken immer mehr in das Interesse von Entwicklern und IT-Architekten. Mit dem Oracle Container Cloud Service stellt Oracle neben dem Oracle Application Container Cloud eine weitere Service Laufzeitumgebungen bereit, mit denen Microservices in der Oracle Cloud betrieben werden können.

## **Das Konzept der Container**

Die Bereitstellung von Containern in der Applikationsentwicklung ist an und für sich keine neue Idee. Auf Betriebssystemebene gibt es sie bereits in Form von Serverklassen. Die Container-Technologie greift im Grunde eine zentrale Aufgabe von Betriebssystemen auf, um Zugriffe auf verschiedene Ressourcen zu ermöglichen und diese auf die laufenden Prozesse zu verteilen. Die neue Technologie führt diese Aufgabe konsequent weiter. Anstelle der gemeinsamen Nutzung von Netzwerken, Dateisystem, CPU und Speicher werden diese durch die Container-Technologie isoliert. Somit wird einem Container eine eingeschränkte Menge an CPU, Speicher und Netzwerkkonfiguration exklusiv zur Verfügung gestellt.

## **Docker**

Die ursprünglich zugrundeliegende Container-Technologie in Linux erstellte Google auf Basis von LXC (Userspace Interface for Linux Kernel Containers). Im ersten Schritt entwickelte Google einen Client, der auf LCX aufsetzt und mit dem erste Container und Images erstellt und verwaltet werden konnten. Diese Images waren im Wesentlichen nichts anderes als Dateisysteme, die sowohl die auszuführenden Prozesse als auch die Konfigurationsdatei für den Start des Prozesses beinhalteten. Bereits damals ging es vor allem darum, mit einem einzigen Befehl ein bestehendes Image zu laden und dieses zu starten.

Inzwischen sind aus Docker alle LXC-Abhängigkeiten entfernt und durch eine Virtualisierung auf Betriebssystemebene ersetzt. Dieser Ansatz ist im Vergleich zu klassischen Virtualisierungstechniken deutlich leichtgewichtiger, da kein eigener Kernel instanziiert werden muss und somit deutlich weniger Speicher benötigt wird.

Ein Docker-Container enthält – wie in Abbildung 1 dargestellt – die Anwendung sowie alle benötigten Abhängigkeiten, teilt jedoch den Kernel mit anderen Containern.

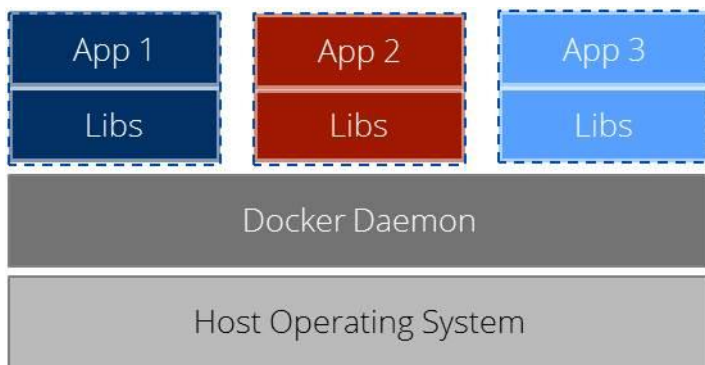


Abbildung 1: Nutzung des gemeinsamen Betriebssystems

Container werden als isolierte Prozesse im User Space des Betriebssystems ausgeführt, das auf dem Host installiert ist und sind nicht an eine bestimmte Infrastruktur gebunden. Somit können Container auf jedem Computer, in jeder Infrastruktur-Umgebung und in der Cloud ausgeführt werden.

### Oracle Container Cloud Service

Mit dem Oracle Container Cloud Service (OCCS) stellt Oracle einen weiteren Service in der Cloud bereit, mit dem Microservices innerhalb eines Docker Containers in der Cloud bereitgestellt und ausgeführt werden können. Während im OACCS die Infrastruktur komplett weggekapselt ist, stellt der OCCS eine vollständig transparente Infrastruktur zur Ausführung und Verwaltung von Docker Containern bereit. Zusätzlich zur Infrastruktur beinhaltet der OCCS umfangreiche Werkzeuge, um containerbasierte Microservices in der Oracle Cloud zur Verfügung zu stellen, zu orchestrieren und zu verwalten.

Analog zum OACCS kann der OCCS sowohl als eine Non-metered Subscription als auch als eine Metered Subscription erworben werden. Auch dieser Dienst kann mit der Trial Subscription getestet werden.

### Instanzen des Container Cloud Services

Für die Bereitstellung von Docker Containern im OCCS müssen zuerst die notwendigen Instanzen erzeugt werden. Eine Instanz des OCCS besteht immer aus einem Manager-Knoten und mindestens einem Worker-Knoten. Im OCCS können bis zu 250 Worker-Knoten erzeugt werden. Zusätzlich zur Anzahl der Worker-Knoten muss die Größe des Worker-Knotens festgelegt werden. Ein Worker-Knoten besteht mindestens aus einer OCPU mit 7,5 GB RAM – die maximale Größe beträgt 16 OCPUs mit 120 GB RAM. Auf die Konfiguration des Manager-Knotens kann im Rahmen der Erstellung der OCCS Instanz kein Einfluss genommen werden.

Sowohl der Manager-Knoten als auch die Worker-Knoten werden im OCCS als Oracle Compute Virtual Machines (VMs) abgebildet. Auf die Konfiguration der VMs kann aktuell noch kein Einfluss genommen werden. Mit einem der nächsten Releases soll es eine Möglichkeit geben, zum Beispiel die Kernel-Parameter der VMs zu modifizieren. Dies ist im Falle des Deployments einer Elasticsearch-Datenbank in OCCS auch notwendig, da für ihren Betrieb die Standardeinstellung für mmap zu niedrig ist. Abbildung 2 zeigt die OCCS Instanz mit einem Manager-Knoten und drei Worker-Knoten einer Service-Definition.

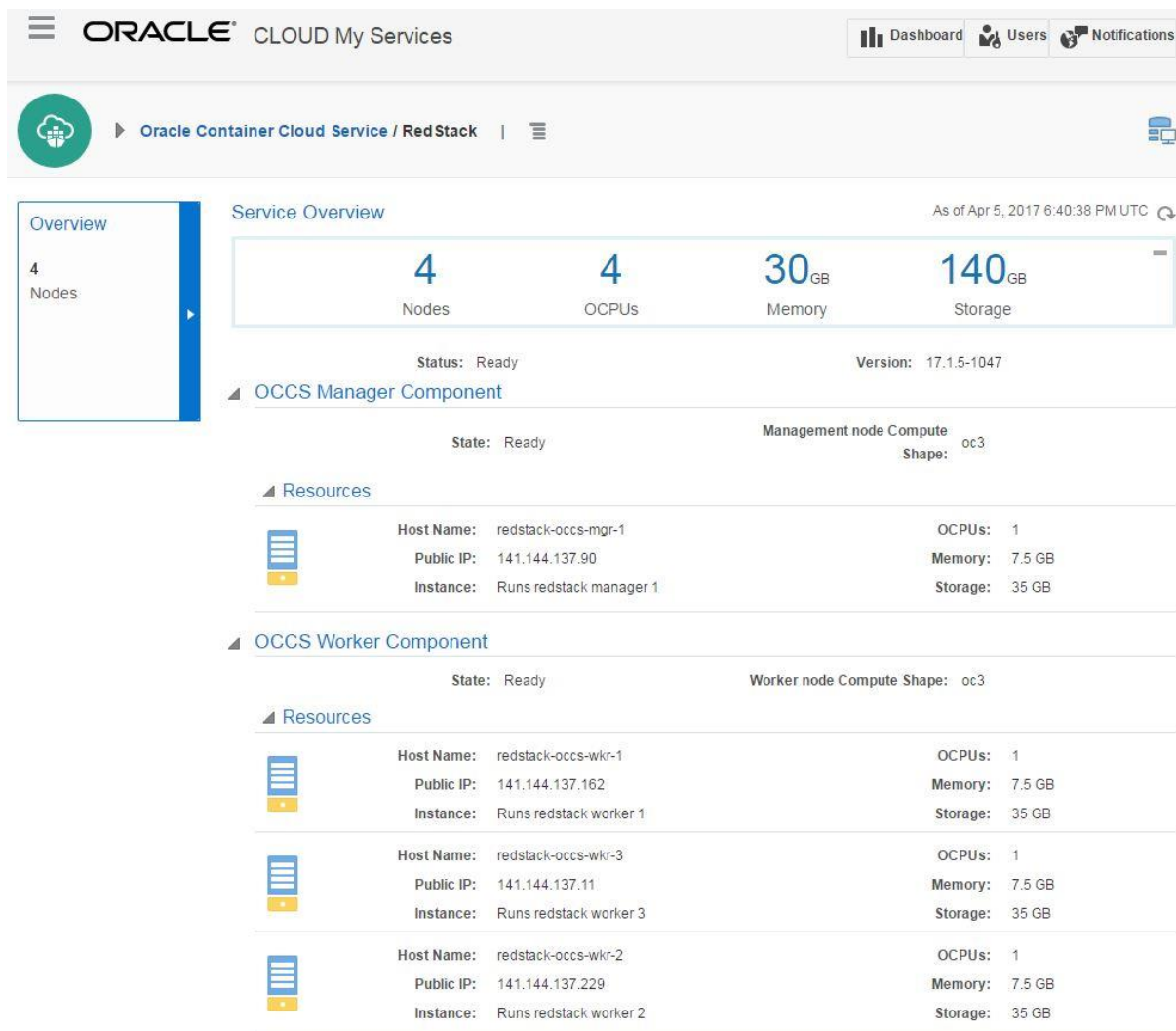


Abbildung 2: Übersicht der Manager- und Worker-Knoten

## Oracle Container Service Dashboard

Nach der erfolgreichen Provisionierung der OCCS Instanz kann nun die Service Console bzw. das Dashboard über das Hamburger-Menü gestartet werden. Für den Start der Service Console ist eine Autorisierung mit Benutzernamen und Passwort des Administrators erforderlich. Beides wurde bei der Initialisierung des Services vergeben. Abbildung 3 zeigt das Dashboard des OCCS.

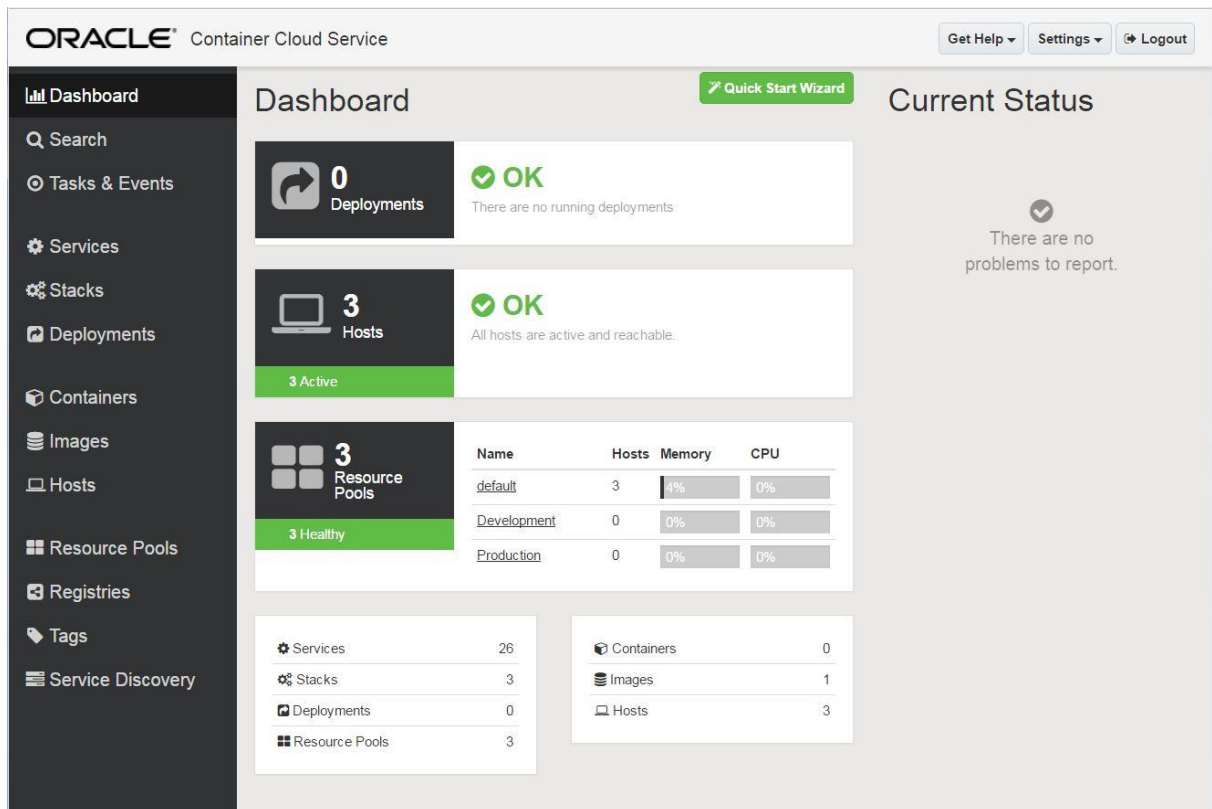


Abbildung 3: Dashboard des Container Cloud Services

Im Folgenden werden einige wichtige Bereiche des Service Dashboards des OCCS vorgestellt.

## Hosts

Wie bereits erwähnt, wird im Rahmen der Definition einer neuen Service-Instanz die Anzahl der Hosts bzw. Work Nodes angegeben. Jeder Host repräsentiert eine eigenständige Oracle Compute Virtual Machine, auf der im Rahmen des Deployments entweder einzelne Container (Services) oder eine Orchestration von Containern (Stacks) bereitgestellt werden.

Zusätzlich wird bei der Initialisierung eines neuen Services zu den Worker-Knoten noch ein Manager-Knoten hinzugefügt. Dieser übernimmt beim Deployment die Verteilung der Services bzw. des Stacks auf die entsprechenden Worker Nodes und überwacht die Ausführung der Container. Zu diesem Zweck wird auf den Worker-Knoten ein entsprechender Agent installiert. Sollte die Kommunikation zwischen Worker-Knoten und Manager-Knoten aufgrund einer Störung für einen Zeitraum von mehr als einer Minute nicht möglich sein, so wird der Worker-Knoten automatisch auf inaktiv gesetzt.

## Resource Pools

In vielen Fällen ist es notwendig, Hosts in logische Einheiten zu gruppieren. Hierzu stellt der OCCS sogenannte Resource Pools bereit. Mithilfe dieser Pools werden Hosts in isolierten Gruppen zusammengefasst. Sie vereinfachen erheblich die Verwaltung der Hosts sowie das Deployment der Services und Stacks auf die entsprechenden Hosts.

Initial werden bei der Erstellung der Service-Instanz die folgenden drei Resource Pools angelegt:

- Default
- Development
- Production

Die Namen der Pools sind lediglich Vorschläge und können bei Bedarf beliebig umbenannt, erzeugt oder gelöscht werden. Abbildung 4 zeigt die Aufteilung der Hosts auf die entsprechenden Resource Pools.

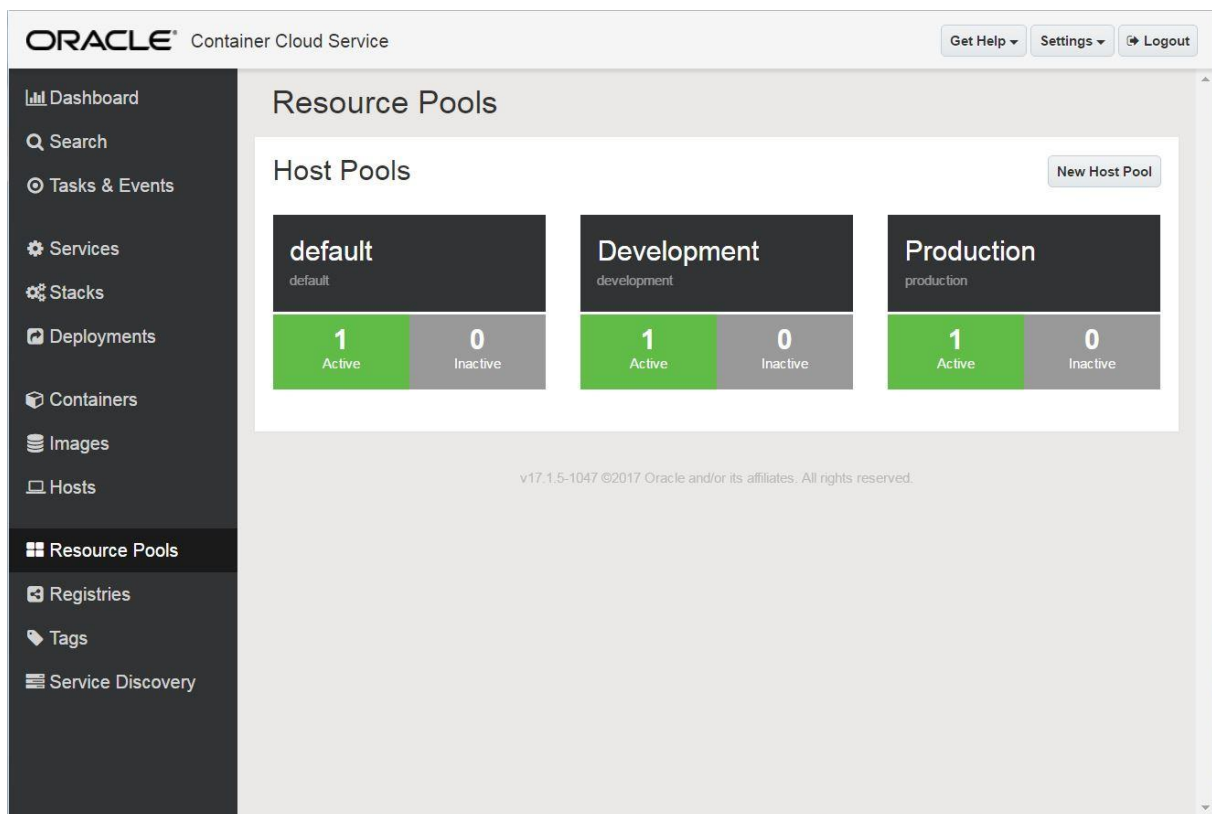


Abbildung 4: Aufteilung der Hosts in Resource Pools

Auch die Zuordnung der Hosts zu den Resource Pools kann jederzeit geändert werden. Dabei sollte jedoch beachtet werden, dass das Verschieben eines Hosts in einen anderen Resource Pool alle laufenden Services beendet. Stehen im ursprünglichen Resource Pool noch freie Kapazitäten zur Verfügung, so werden die beendeten Services automatisch auf einem neuen Host innerhalb des ursprünglichen Resource Pool gestartet.

## Services

Im OCCS umfasst ein Service alle notwendigen Konfigurationen für die Bereitstellung eines Docker Images auf einem Host im entsprechenden Resource Pool. Services stellen im OCCS noch nicht den lauffähigen Container dar, sondern sie entsprechen vielmehr einer High-Level-Konfiguration oder einem Template, in dem alle Konfigurationsparameter für die spätere Ausführung gespeichert sind.

Nach der Initialisierung des OCCS stehen eine Reihe von vorkonfigurierten Services bereit, die entweder out-of-the-box verwendet werden können, oder gemäß spezifischer Anforderungen angepasst werden. Die Liste der Services kann durch die Angabe eines Images und der notwendigen Konfiguration beliebig erweitert werden. Für das Anlegen bzw. das Erweitern einer Konfiguration der Services stehen dem Benutzer die folgenden Möglichkeiten zur Verfügung:

- Auswahl von Optionen aus einer Liste
- Einfügen oder Kopieren eines Docker Befehls
- Einfügen oder Kopieren einer Definition in YAML

Abbildung 5 zeigt die Konfiguration von NGINX im Service Editor.

The screenshot shows the 'Service Editor' window for configuring a service named 'NGINX' with ID 'nginx'. The 'Image' field is set to 'nginx:1.10.1-alpine' and the 'Command' is 'sh -c "echo -e \$NGINX\_CONF > /etc/nginx/conf.d/default.conf; nginx -g 'daemon off;''". The 'Environment Variables' section contains two entries: 'NGINX\_CONF' with a server configuration snippet and 'occs:description' with a note about the educational nature of the example. On the right, the 'Available Options' sidebar lists various Docker options with checkboxes, such as 'Environment Variables' (checked), 'Ports', and 'DNS'. At the bottom, there are buttons for 'Cancel', 'Save As...', and 'Save'.

Abbildung 5: Konfiguration von NGINX im Service Editor

Neben dem Erstellen und Anlegen von Services können in diesem Bereich auch nicht mehr benötigte Services gelöscht werden. Allerdings kann ein Service nur dann gelöscht werden, wenn zu ihm kein entsprechender Container läuft.

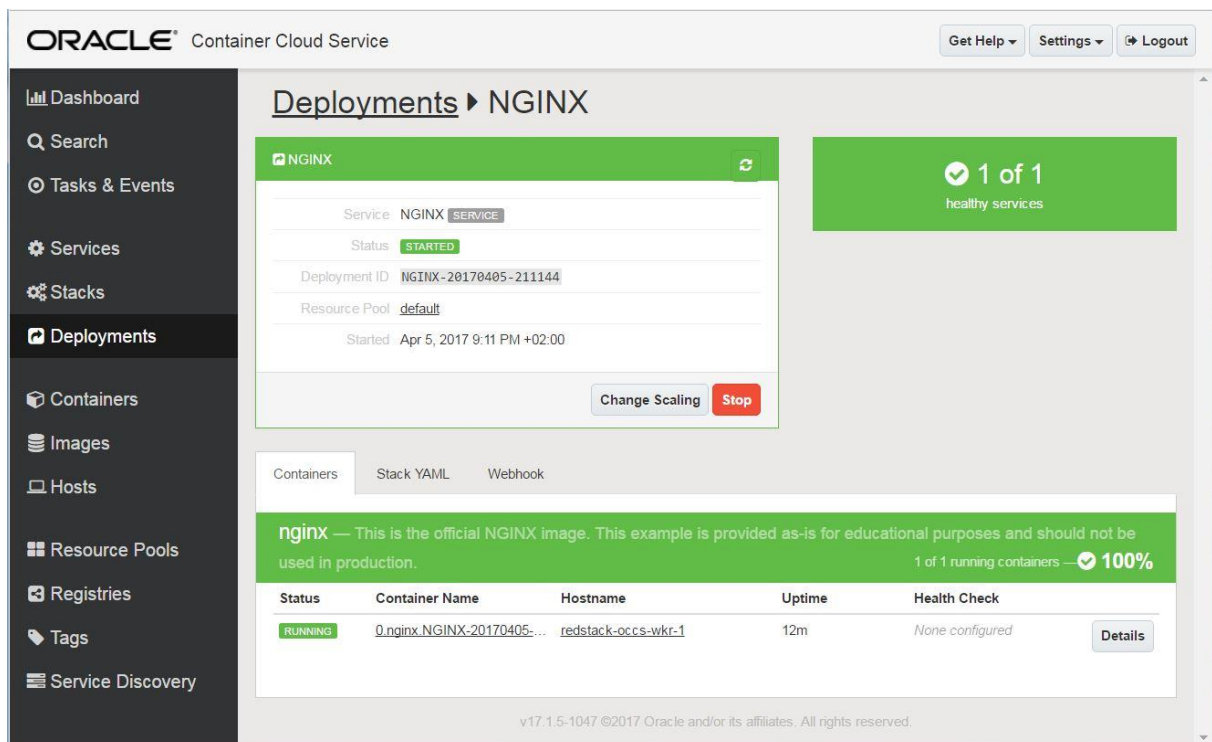
## Images

Wird im OCCS ein Docker Image zum ersten Mal deployed, wird das zugehörige Image aus der entsprechenden Registry geladen und der Liste der geladenen Images hinzugefügt. Im Bereich „Images“ des Dashboards des OCCS erfolgt die Verwaltung aller geladenen Images. Docker Images können hier direkt gestartet werden. Allerdings müssen die für die Ausführung notwendigen Parameter in der Konsole erfasst werden. Jedoch wird hierbei kein Service angelegt, sodass die bereits erfasste Konfiguration für die Ausführung des Images nicht überschrieben wird.

## Deployments

Im Bereich Deployment erfolgt die Bereitstellung, Skalierung und Verwaltung von Services oder Stacks gemäß den definierten Orchestrierungsregeln. Ein Deployment wird im OCCS jedes Mal automatisch beim Start eines Service beziehungsweise eines Stacks erstellt.

Im Rahmen des Deployments eines Services oder Stacks greift die Orchestrierungsfunktionalität, die im OCCS enthalten ist. So wird bei der Bereitstellung ein neues Deployment-Objekt erzeugt und, falls notwendig, die benötigten Images aus der Registry geladen. Im nächsten Schritt erfolgt die Erzeugung der Container sowie, falls nötig, die Einrichtung der Kommunikation der Services untereinander. Anhand der Orchestrierungsregeln erfolgt die Verteilung der benötigten Instanzen eines Containers auf die definierten Hosts in den entsprechenden Resource Pools. Abbildung 6 zeigt das Deployment eines NGINX-Images im Resource Pool default.



ORACLE® Container Cloud Service

Get Help Settings Logout

Dashboard Search Tasks & Events Services Stacks **Deployments** Containers Images Hosts Resource Pools Registries Tags Service Discovery

### Deployments > NGINX

NGINX

Service: NGINX SERVICE

Status: STARTED

Deployment ID: NGINX-20170405-211144

Resource Pool: default

Started: Apr 5, 2017 9:11 PM +02:00

Change Scaling Stop

Containers Stack YAML Webhook

nginx — This is the official NGINX image. This example is provided as-is for educational purposes and should not be used in production. 1 of 1 running containers — 100%

Status	Container Name	Hostname	Uptime	Health Check
RUNNING	0.nginx.NGINX-20170405-...	redstack-occs-wkr-1	12m	None configured

Details

v17.1.5-1047 ©2017 Oracle and/or its affiliates. All rights reserved.

Abbildung 6: Deployment des NGINX-Images

## Docker Registry

Die Speicherung von Docker Images erfolgt in einer sogenannten Docker Registry. Eine Docker Registry verwaltet die verschiedenen Versionen eines Docker Images. Um einen Service in einem

Resource Pool deployen zu können, muss der OCCS das gewünschte Image aus einer Service Registry herunterladen und starten. Für das Laden von Images ist bereits die öffentliche Docker Hub Registry vorkonfiguriert. Somit kann der OCCS ohne zusätzlichen Konfigurationsaufwand auf öffentliche Images zugreifen. Für den Zugriff auf Images aus einer anderen öffentlichen oder privaten Docker Registry, kann im OCCS die Definition weiterer Registrys aufgenommen werden.

Im OCCS ist aktuell leider keine Docker Service Registry enthalten, sodass eigene Images nur über die Docker Hub Registry deployed werden können. Alternativ muss eine eigene Service Registry aufgebaut und im OCCS registriert werden.

### **Fazit**

Mit dem Oracle Application Container Cloud Service und dem Oracle Container Cloud Service stellt Oracle zwei unterschiedliche Services für den Betrieb von Microservices in der Cloud zur Verfügung. Der OACCS eignet sich insbesondere für die Bereitstellung von leichtgewichtigen Microservices in einem Docker Container. Wird hingegen eine vollständige Integration zu weiteren Services oder Standard-Containern benötigt, so ist an dieser Stelle der OCCS zu empfehlen.

### **Kontaktadresse:**

Stefan Kühnlein  
OPITZ CONSULTING Deutschland GmbH  
Weltenbuger Str. 4  
D- 81677 München

Telefon: +49 (0) 173-7279307  
Fax: +49 (0) 89-680098-440  
E-Mail: stefan.kuehnlein@opitz-consulting.com  
Internet: www.opitz-consulting.com