

Entwicklung und Betrieb von Microservices und Cloud-Native Anwendungen mit dem ACCS und DevCS

Peter Doschkinow
Oracle Deutschland B.V. & Co KG

Schlüsselworte

Cloud Native Applications, Microservices, DevOps, Oracle Cloud, Oracle Application Container Cloud Service, Oracle Developer Cloud Service

Einleitung

Im Unternehmen finden sich heute typischerweise 3 Anwendungskategorien – ältere Back-End ERP oder HR-Systeme, die wichtige Geschäftsfunktionen implementieren, aber überholte Technologien verwenden und wegen diversen Abhängigkeiten nicht mehr angefasst werden können, große leistungsfähige, monolithische Java EE basierte transaktionale Anwendungen, die in den letzten 5-10 Jahren entstanden sind und das Geschäft am Laufen halten und neue moderne Applikationen, bei denen die eingesetzte Technologie sekundär ist und die nur danach gemessen werden, ob sie schnell genug auf ständig neuen Geschäftsanforderungen reagieren können, robust und beliebig skalierbar sind, und neue Business-Chancen sofort unterstützen können. Mit dieser letzten Kategorie werden Cloud Native Applications, Microservices und innovative, ansprechende mobile Anwendungen assoziiert, die für den großen Erfolg von Firmen wie Uber, Airbnb und Netflix entscheidend beigetragen haben.

Cloud Native Applications

Native Applications sind solche, die für eine spezielle Plattform oder Geräte-Typ entworfen und optimiert wurden. Analog, Cloud Native Applications sind Anwendungen, die für den Einsatz in der Cloud gebaut wurden, die Vorteile von Cloud-Frameworks, die aus verteilten und elastischen Cloud-Diensten bestehen, ausnutzen und ihre Einschränkungen berücksichtigen. The Twelve Factor App (<http://12factor.net>) kann als Best Practice für Cloud Native Applications angesehen werden. Letztes Jahr wurde die Hersteller-übergreifende Cloud Native Computing Foundation (<http://cncf.io>) gegründet, die dieses neue Anwendungsmuster unterstützt und vorantreibt. Als wichtigste Eigenschaften von Cloud Native Applications sieht sie eine Microservice-Architektur und dynamische Orchestrierung und Verwaltung ihrer Komponenten, die in Container verpackt und verteilt werden.

Microservices

Microservices ist eine Anwendungs-Architektur (zum Unterschied von SOA – eine Enterprise-Architektur), bei der sich die Anwendung aus vielen kleinen lose-gekoppelten und autonomen Microservices zusammensetzt. Microservices nutzen SOA Best-Practices. Sie können ihre eigene Datastores haben, externalisieren ihren Zustand und modellieren entsprechend dem Domain-Driven-Design unterschiedliche Geschäftsfunktionen. Microservices können in unterschiedlichen Programmiersprachen implementiert werden. Sie können auch unabhängig von einander deployed werden, wobei dann nur einzelne Anwendungsfunktionen nicht verfügbar sind. Microservices setzen einen automatisierten DevOps-Prozess voraus und werden über ihren gesamten Lebenslauf von kleinen gemischten Teams entwickelt und betrieben.

Application Container Cloud Service (ACCS)

Der Oracle Application Container Cloud Service ist eine sichere, hoch-verfügbare, elastische und polyglotte Cloud Plattform für Cloud Native Applications. Momentan können JavaSE und Node.js basierte Anwendungen mit Anbindungen an den Database-, Java- und Messaging Cloud Services darauf betrieben werden. Geplant ist die Unterstützung von Laufzeitumgebungen für PHP, Python, Cache (auf Coherence-Basis) und Java EE web profile (auf WebLogic-Basis) sowie die Integration mit weiteren Cloud-Diensten wie NoSQL, MySQL und API Gateway. Zusammen mit dem Developer Cloud Service bietet ACCS eine offene, integrierte und vollkommen automatisierte DevOps Pipeline.

Der ACCS nutzt hinter den Kulissen eine leicht-gewichtige, Docker-Container basierte Infrastruktur. Der Entwickler bereitet seine Java SE oder Node.js Anwendung als ein File-Archiv vor, in dem sämtliche Abhängigkeiten und ggf. 2 gesonderte Konfigurationsdateien verpackt sind. Wenn er sie dann über das ACCS Browser UI oder die ACCS REST API installiert, wird sie in Docker Container entsprechend den konfigurierten Ressourcen provisioniert und ein Load Balancer wird automatisch vorangestellt:

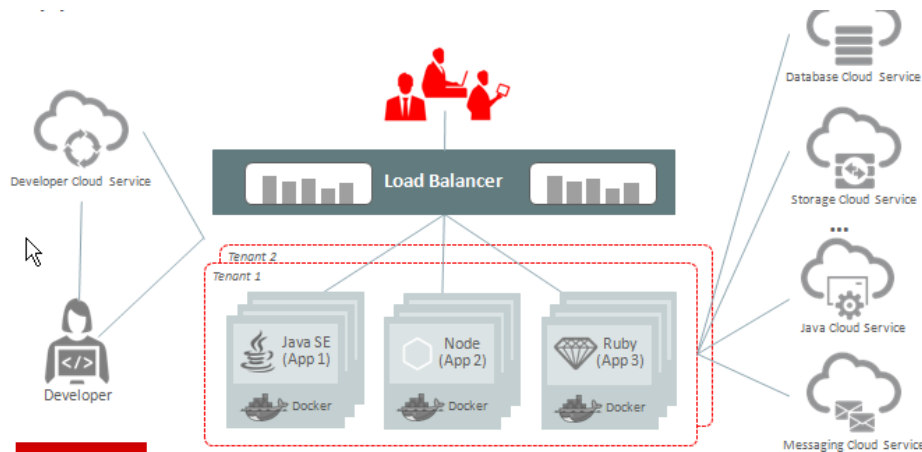


Abb. 1: ACCS Architektur

Der ACCS vereinfacht die Entwicklung und den Betrieb von Cloud Native Applications und Microservices, da er folgende Infrastruktur-Anforderungen entweder selbst oder durch die Integration von weiteren Cloud-Diensten abdeckt:

Polyglot	Configuration	Data Storage
State	Scaling	Load Balancer
Monitoring	Logging	<u>Eventing</u>
Messaging	Security	Continuous Delivery

Abb. 2: Infrastruktur-Anforderungen von Microservices und Cloud Native Applications

Im Vortrag werden diese ACCS-Eigenschaften detaillierter erläutert.

Ein Alleinstellungsmerkmal von ACCS ist die Integration von Java Flight Recorder und Java Mission Control(JMC). Somit ist es möglich Java SE Anwendungen in Produktion ohne Performance- und sonstige Nebenwirkungen zu profilieren und die Ergebnisse mit JMC auszuwerten.

Oracle Developer Cloud Service (DevCS)

Mit ACCS kommt der Oracle Developer Cloud Service umsonst mit. Dieser integriert verschiedene Werkzeuge, die für die Erstellung von Anwendungen und die Team-Kollaboration von Interesse sind, in einem Entwicklerportal. Anwendungen werden dabei durch Projekte verwaltet.

Abbildung 1 zeigt eine Sicht auf das Projekt namens „DemoProject“ und man erkennt hier die wesentlichen Bestandteile des Oracle Developer Cloud Service:

- Git zur Verwaltung von Quellcode: ein oder mehrere Git Repositories können einem Projekt zugeordnet werden
- Graphische Behandlung von Merge Requests durch einen Workflow
- Issue Tracking
- Hudson Build/CI Server
- Maven Unterstützung zur Automatisierung von Paketierung/Test/Bereitstellung der Anwendung
- Deployment u.a. auf dem zugehörigen ACCS oder Java Cloud Service durch Deployment Konfigurationen
- Agile-Board und Wiki für Kollaboration, Kommunikation und gemeinsame Benutzung von Informationen im DevOps Prozess.

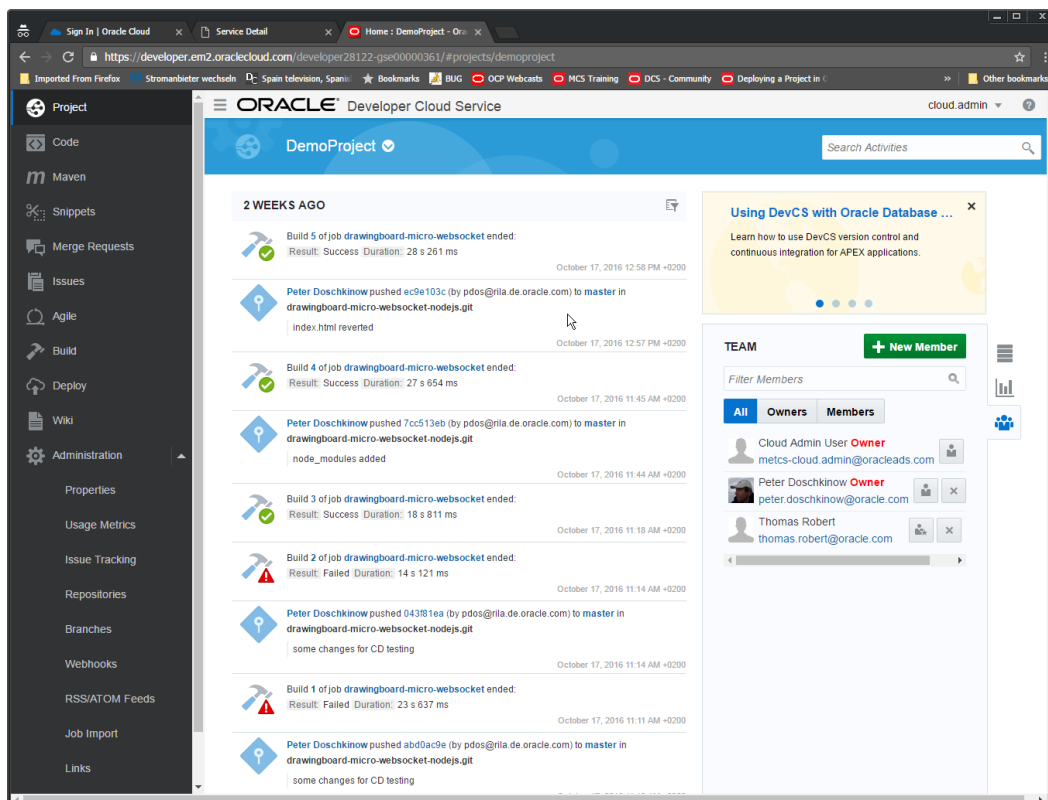


Abb. 2: Projektsicht im Oracle Developer Cloud Service

Der Developer Cloud Service kann als PaaS Developer Plattform angesehen werden, wobei die Entwicklung geographisch und über Organisationsgrenzen hinweg verteilt erfolgen kann. Eine dezentrale Entwicklung wird dank der Einbindung der Cloud Dienste in IDEs wie Oracle JDeveloper, Eclipse (Oracle Enterprise Pack for Eclipse) und NetBeans sehr gut unterstützt. Über konfigurierbare WebHooks lassen sich auch weitere Tools aus der Kunden-Entwicklungsumgebung wie Jenkins und Jira anbinden.

Als zentrale Buildumgebung kann der Hudson CI Server in Kombination mit dem ACCS und dem Oracle Java Cloud Service genutzt werden, um nach erfolgreichem Build und Test die Anwendung auf der gewünschten Zielumgebung zu deployen.

Zusammenfassung

Der ACCS wurde entwickelt um eine einfache Ablaufumgebung für Cloud Native Applications bereitzustellen. Zusammen mit dem DevCS und anderen Oracle Cloud Diensten bietet er dem Entwickler ein vorgefertigtes und integriertes Enterprise-Ecosystem, das den kompletten DevOps-Prozeß automatisiert.

Links

ACCS: https://cloud.oracle.com/de_DE/acc

DevCS: https://cloud.oracle.com/de_DE/developer_service

Kontaktadresse:

Peter Doschkinow
Riesstrasse 15
D- 80992 München

Telefon: +49 (0) 89-14302514

E-Mail: peter.doschkinow@oracle.com

Internet: www.oracle.com