

Red Stack

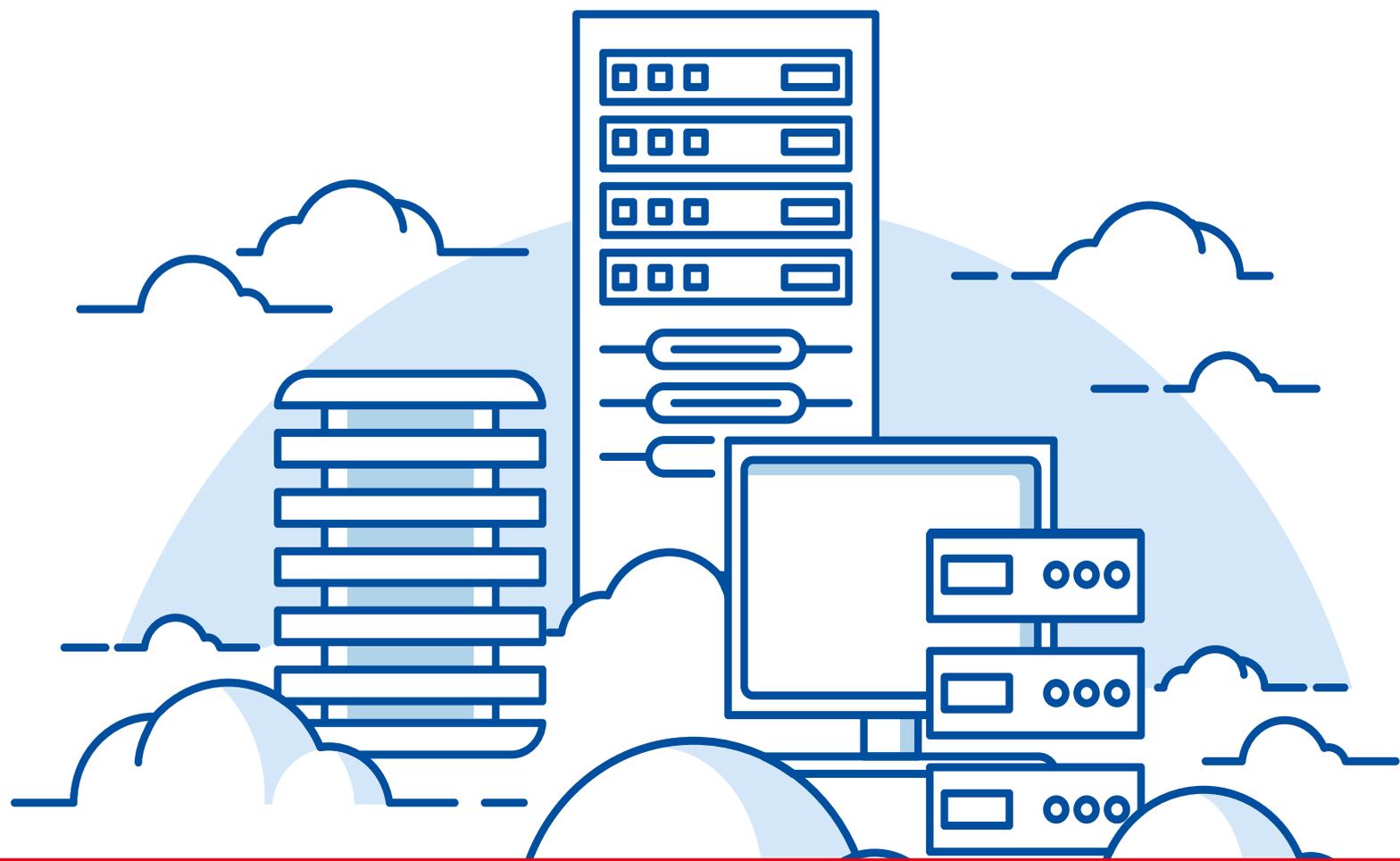
Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

Cloud-Services



Oracle NoSQL

Verarbeitung von
Streamingdaten

Im Interview

Andre Lünsmann,
Barmenia Versicherungs-
gruppe



Topaktuell

Die autonome
Oracle-Datenbank

ORACLE®

**Testen Sie die Oracle Cloud
mit Credits im Wert von**

\$300



**Registrieren Sie sich und erhalten Sie eine Gutschrift
für das Pay-as-you-go-Abonnement von Oracle
Cloud-Services in drei einfachen Schritten:**

cloud.oracle.com/de_DE/tryit



Jan-Peter Timmermann
Leiter Middleware und
Infrastruktur Community

Liebe Mitglieder, liebe Leserinnen und Leser,

mit Spannung haben wir die Worte von Paul Wehner auf der Delegierten-Versammlung am 20. April 2018 in Berlin verfolgt. Es ist schon sehr interessant, wie Oracle das Thema „Cloud“ betrachtet und was für Fortschritte sie machen. Leider sind wir immer noch nicht wirklich von den rechtlichen Bedingungen der Cloud-Thematik angetan und hoffen, dass sich dort noch einiges zum Guten ändern wird.

Im Herbst letzten Jahres wurden die Oracle-Cloud-Rechenzentren in Frankfurt eröffnet. Spätestens seitdem nimmt die Oracle-Cloud auch in Deutschland Fahrt auf. Die Architektur wird kräftig modernisiert und die neue Oracle Cloud Infrastructure (OCI) ersetzt Schritt für Schritt die Dienste der OCI-Classical. Oracle setzt damit Maßstäbe, die die anderen Anbieter zurzeit blass aussehen lassen.

Aber brauchen wir so viel Leistung aus der Cloud wirklich? Wie nutzen Sie Cloud-Dienste und wie gut fühlen Sie sich über die Angebote informiert? Das möchte die DOAG gerne wissen und hat deshalb eine Umfrage zusammengestellt. Bitte nehmen Sie sich die Zeit. Sie finden die Umfrage unter „www.doag.org“.

Wir wünschen Ihnen viel Spaß beim Lesen und viel Erfolg mit allen Cloud-Projekten.

Ihre



Robert Marz
DOAG-Themenverantwortlicher Cloud

MUNIQSOFT

Lizenzen

Durchblick mit IQ

Experten-Knowhow für Ihr Oracle Lizenzmanagement!
Setzen Sie auf Lizenzmodelle, die nicht nur wirtschaftlich sind, sondern auch einem Audit standhalten.

Nutzen Sie die Beratung von Muniqsoft

ORACLE® Gold Partner

Specialized
Oracle Database



Jetzt Beratungstermin vereinbaren:
+49 (0)89 6228 6789-39

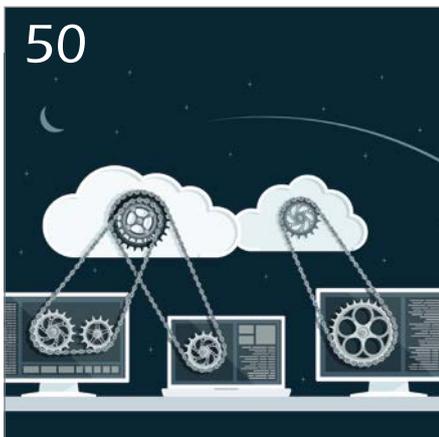
www.muniqsoft.de



Im privaten Umfeld fällt es uns gar nicht mehr auf, wie häufig wir die Cloud benutzen



Seit mehr als fünf Jahren ist das Thema „Cloud“ in aller Munde



Viele neuer IT-Lösungen kombinieren Cloud-Dienste mit vorhandenen lokalen Systemen

Einleitung

- 3 Editorial
- 5 Timeline
- 8 „Ein Test mit dem Oracle Advanced Customer Support brachte nicht den erhofften Erfolg ...“
Interview: Andre Lünsmann

Cloud

- 11 Aufgeklärt – ein Überblick über Cloud Services
Robert Marz
- 13 Oracle Database 18c und Database as a Service
Tobias Deml
- 18 Über den Wolken – grenzenlose Freiheit mit purem Metall?
Katharina Büchse und Florian Barth
- 23 Cloud Service – Himmel und Hölle
Alexander Weber
- 29 „Lift & Shift“ oder wie kommt meine (Pluggable) Datenbank in die Oracle-Cloud
Kai Uwe Fischer
- 37 Die autonome Datenbank: ADWC für Warehouse-Anwendungen
Ulrike Schwinn
- 41 Architecting 4 the Cloud
Kai Weingärtner und Stefan Kühnlein
- 47 Leute hassen es zu warten
Jan-Peter Timmermann
- 50 Cloud meets On-Premises – Herausforderungen in hybriden Infrastrukturen
Danilo Schmiedel
- 55 Serverless mit Fn Project
Dr. Frank Munz
- 63 Container Native Development Platform
Marcel Amende

Entwicklung

- 67 Objekte machen das Leben leichter
Jürgen Sieben

Datenbank

- 69 Verarbeitung von Streamingdaten – ein Erfahrungsbericht der Testphase mit Oracle NoSQL
Marcus Bender, Rainer Marekvia, Thomas Robert und Dr. Nadine Schöne

Intern

- 73 Termine
- 73 Neue Mitglieder
- 74 Impressum
- 74 Inserenten

✦ Timeline

2. März 2018

Der DOAG-Vorstand diskutiert auf seiner Sitzung in Frankfurt den Programm-Rahmen für die DOAG 2018 Konferenz + Ausstellung und bereitet den Call for Paper vor. Damit sind die Voraussetzungen geschaffen, um auf der Veranstaltung, die vom 20. bis 23. November 2018 wie gewohnt in Nürnberg stattfindet, wieder ein sehr interessantes Vortragsprogramm anbieten zu können. Darüber hinaus legt der Vorstand die letzten Details zur 30-Jahre-Feier der DOAG fest.

12. März 2018

Bereits am Vortag der JavaLand 2018 findet die kleine Schwester statt: Die JavaLand4Kids richtet ihren Fokus auf die Programmierer und Entwickler von morgen. Zum ersten Mal ist nicht nur eine Grundschulklasse der Hans-Christian-Andersen-Schule aus Sankt Augustin zu Gast, sondern auch Schüler des Max-Ernst-Gymnasiums aus Brühl. Die Grundschüler im Alter von neun bis zehn Jahren schaffen in kurzer Zeit musikalische Variationen mit Sonic Pi, einem Open-Source-Musikprogramm für den Raspberry Pi. Die Informatik-Klasse des Gymnasiums hat mit Iryna Feuerstein und Oliver Milke zwei erfahrene Mentoren an ihrer Seite. Hier steht das Thema „Neo4J“ im Raum, eine Graphen-Datenbank, die man nicht aus dem Schulalltag kennt. Nach einer etwas umfangreicheren Einführung tauchen die Jugendlichen ab in die spannende, neue Welt der Daten-Visualisierung. Last but not least beschäftigt sich der andere Teil der Klasse mit der gehobenen Kunst des Minecraft Modding, moderiert von Arun Gupta, der seit vielen Jahren die sehr aktive Devovx4Kids-Gruppe in der BayArea organisiert. Noch am selben Tag kommt sowohl von den Lehrern als auch von den Schülern die Rückmeldung, dass es sehr viel Spaß gemacht habe und sie gerne wiederkommen würden – ganz klar die Aufforderung, auch für das Jahr 2019 ein spannendes Programm auf die Beine zu stellen.



Die Kleinen sind begeistert bei der Sache

13./14. März 2018

Fried Saacke, DOAG-Geschäftsführer und Vorstandsvorsitzender des Interessenverbands der Java User Groups e.V. (iJUG), eröff-

net mit einem kräftigen „Jatumba“ die JavaLand 2018. Er kann stolz einen neuen Besucherrekord verkünden: Mehr als 1.900 Entwickler aus zwanzig Nationen sind im Phantasialand in Brühl. Nach dem Start im Jahr 2014 mit 800 Teilnehmern ist die Veranstaltung ständig stark gewachsen. Den Auftakt zur diesjährigen Konferenz gibt Keynote-Speakerin Dr. Holly Cummins. In ihrem Vortrag „Cloudy with a Chance of Meatballs: Cloud Surprises for the Java Developer“ teilt sie im bis auf den letzten Platz besetzten Vortragssaal neben einer historischen Perspektive auf Cloud Computing auch einige Gedanken darüber, wohin die Entwicklung in Zukunft gehen könnte. Neben den rund 110 Vorträgen in acht Streams zu Themen wie „Architektur und Sicherheit“, „Container und Cloud“, „Enterprise und Microservices“ sowie „Innovationen“ sorgen auch die aktuellen Entwicklungen im Java-Kosmos rund um die Zukunft von Java EE/Jakarta EE für viel Gesprächsstoff. Auch die vielen Community-Aktivitäten, die zusammen mit zahlreichen Java User Groups gestaltet wurden, bieten den Teilnehmern viele Möglichkeiten zum Ausprobieren, Entdecken und Spielen.



Die Eröffnungsveranstaltung ist bis auf den letzten Platz besetzt

19./20. März 2018

Unter dem Motto „Daten als Motor der Digitalisierung“ übertrifft die gemeinsame Data-Analytics-Konferenz von Oracle und der DOAG im Phantasialand in Brühl mit rund 250 Besuchern und 14 Ausstellern die Erwartungen. Zukünftige Innovationen und Digitalisierung ziehen sich wie ein roter Faden durch die gesamte Veranstaltung. Besonders erfreulich: Neben Oracle-Experten und Großunternehmen halten namhafte Kunden knapp die Hälfte der Vorträge. Das Programm ist vielfältig; neben klassischen, technischen Themen finden auch aktuelle Fragestellungen zu Machine Learning, Design Thinking, Geodaten und EU-Datenschutz-Grundverordnung Gehör. Auch außerhalb der Vortragsräume kommen viele aktive Gesprächsrunden auf. Eine interaktive Podiumsdiskussion zum Thema „Der Weg zu künstlicher Intelligenz und Machine Learning“, bei der jeder zu Wort kommen kann, rundet die Veranstaltung am Nachmittag ab. Die Organisatoren Rolf Scheuch, Leiter der DOAG Data Analytics Community, Christiane Toffolo-Haupt, Oracle Sales Direktorin für Analytics und Big Data, und Alfred Schlaucher, langjähriger Oracle-Experte im Bereich

„Data Warehouse“, ziehen ein positives Resümee: „Die Mischung aus Vorträgen zu BI/DWH, Cloud Computing und zukunftsgerichteten Themen sind eine gute Basis für ein intensives Networking der Teilnehmer. Wir freuen uns auf die Fortsetzung im Jahr 2019.“



Data Analytics 2018: Der Zukunft ein Stück näher

27. März 2018

In einer regelmäßig stattfindenden Telko reden die Vertreter der europäischen Anwendergruppen mit Tom Scheirsens, EMEA User Groups Relationship Manager von Oracle. Sie erfahren, dass Oracle das traditionell am Vortag der Oracle OpenWorld von den Usergroups organisierte Programm gestrichen hat und dass es trotzdem nur wenige ausgesuchte Usergroup-Vorträge in das Hauptprogramm schaffen werden. Die Oracle OpenWorld verliert damit aus Sicht der Usergroups an Bedeutung und wird zur reinen Marketing-Veranstaltung des Herstellers.

3. April 2018

Fried Saacke, DOAG-Vorstand und Geschäftsführer, begrüßt in der DOAG-Geschäftsstelle zwei neue Mitarbeiter. Christobal Jimenez und Torsten Stemmler verstärken das IT-Team, um viele anstehende Anforderungen an die IT zu erfüllen und damit den Service für die Mitglieder und die Geschäftsstelle zu verbessern.

11./12. April 2018

Im Berliner Expertenseminar vermittelt Clemens Bleile, Oracle Certified Professional und Oracle Certified Expert for Performance Management and Tuning, sein Wissen zum Datenbank- und SQL-Tuning. Die technischen Datenbank-Themen sind bei den Expertenseminaren traditionell gut besucht und kommen bei den Teilnehmern auch sehr gut an.

18. April 2018

Peter Gübeli, Präsident der Swiss Oracle User Group (SOUG), präsentiert auf der jährlichen Generalversammlung die Bilanz des Jahres 2017. Es ist trotz der leicht rückläufigen Mitgliederzahl ein weiteres erfolgreiches Jahr. Der Vorstand hat für seine zukünftige Strategie eine externe Unterstützung beauftragt; die Studie zeigt, dass die SOUG sich weiterhin auf Oracle fokussieren, aber

mehr junge Menschen an Bord bekommen und auch den Kundent Stamm professioneller pflegen muss. Dafür wird jetzt eine Key-Account-Management-Unterstützung gesucht. Peter Gübeli teilt der Generalversammlung mit, dass er die SOUG verlassen wird. Der Vorstand dankt ihm für sein tolles Engagement während der letzten vier Jahre, Interimspräsident ist jetzt Thierry Bossahrt.

18. April 2018

Nach der Generalversammlung findet ein verkürzter SOUG-Day mit zwei parallelen technischen Streams statt. Dabei steht die Oracle-Datenbank im Vordergrund, die Vorträge reichen vom I/O-Management über Security-Themen bis hin zu Performance-Fragen. Die Veranstaltung endet mit dem traditionellen Apéro, bei dem die Teilnehmer die Möglichkeit haben, ihr Netzwerk zu erweitern.



Jerôme Witt bei seinem Vortrag

19./20. April 2018

Rund 80 aktive Mitglieder der DOAG versammeln sich zu ihrem jährlichen Führungskräfte-Treffen in Berlin, um über die Zukunft des Vereins zu beraten und auf dreißig Jahre Vereinsgeschichte zurückzublicken. Es beginnt mit einer Informationsrunde zum Stand der ein Jahr zuvor definierten Aktivitäten von insgesamt sieben Arbeitsgruppen. Darüber hinaus räumt die Veranstaltung der thematischen Arbeit in den derzeit sieben Communities mehrere Stunden Zeit ein, um die kommenden Aktivitäten und Ziele zu besprechen.

20. April 2018

Das Führungskräfte-Treffen endet mit einem Gastbeitrag von Paul Wehner, Senior Director Sales Consulting bei Oracle Deutschland, der die „Cloud first“-Strategie näher beleuchtet und auf Fragen des Plenums, insbesondere zu den Themen „Lizenzierung in virtualisierten Umgebungen“ und „Datenschutz in der Cloud“, eingeht. Zum „Managed Cloud Service“ kommen zukünftig „Autonomous Database Cloud Services“ dazu. Die Unterschiede liegen vor allem darin, dass bei den „Autonomous“-Lösungen die Kosten für die Datenbank-Administration deutlich niedriger liegen. Der Verlauf der Produkt-Roadmap sei von der Kundenresonanz abhängig. Oracle selber sieht sich – so Wehner – durch die Rechenzentren in Austin (Texas) und bald in Frankfurt am Main technisch gut vorbereitet.

20. April 2018

Die DOAG feiert in Berlin mit 140 geladenen Gästen das 30-jährige Bestehen. Unter den Gästen sind viele ehemalige Wegbegleiter, die die DOAG mit aufgebaut haben, sowie wichtige Geschäftspartner. Ein Film stellt die Entwicklung des Vereins von der Gründung bis heute vor (siehe „<http://www.doag.org/go/video-30-jahre>“).



Die ehemaligen Vorstandsvorsitzenden der DOAG (von links): Fried Saacke, Dr. Dietmar Neugebauer, Eva Kraut, Agnes Hombrecher, Jochen Frickel und Stefan Kinnen (heute)

20. April 2018

Parallel zum Führungskräfteforum findet die Gründungsveranstaltung des DOAG Legal Council statt. Die neue Runde mit spezialisierten IT-Rechtsanwältinnen und Rechtsanwälten ergänzt die fachliche Kompetenz der DOAG, soll den DOAG-Gremien beratend zur Seite stehen, Anliegen der Mitglieder gebündelt beraten und mit dem gesamten DOAG-Netzwerk fundiertes Fachwissen austauschen.

21. April 2018

Die ordentliche Delegiertenversammlung der DOAG tagt, um die strategische Ausrichtung und zukünftigen Ziele des Vereins festzulegen. Die Versammlung verabschiedet die Social-Media-Guidelines der DOAG und wählt Axel vom Stein und Anja Albrecht als Kassenprüfer des Vereins.

21. April 2018



Dr. Dietmar Neugebauer ist Ehrenmitglied der DOAG. Die ordentliche Delegiertenversammlung verleiht dem promovierten Chemiker diese Ehre, nachdem er für unterschiedliche Bereiche des Vereins in besonderem Maße Verantwortung getragen und diesen zuletzt von Juni 2008 bis Mai 2016 als Vorstandsvorsitzender erheblich geprägt hat. Neben seinem ehrenamtlichen Engagement war der Münchner rund 25 Jahre als Datenbank-Spezialist für die

BMW-Gruppe tätig und trug im Unternehmen die Verantwortung für die Produktstrategie, Architektur und Lizenzierung von Oracle-Produkten.

24./26. April 2018

In Düsseldorf findet die vierte Ausgabe der Apex Connect statt. Mit mehr als 350 Besuchern stellt die Konferenz einen neuen Rekord auf. Rund 60 Vorträge und vier Keynotes bringen die Teilnehmer in den Bereichen „Apex“, „PL/SQL“ und „JavaScript“ auf den aktuellen Stand. Mit Apex Beginners Track, World Café und Apex Lightning Talks sind spannende, neue Formate mit am Start. Außerdem immer mit dabei: der einmalige Community-Spirit. Inzwischen schon Tradition: Am Anfang einer jeden Apex Connect steht eine Runde Frühsport auf dem Programm. Die La-Ola-Welle, die begleitet von lauten „Apex“-Rufen durch die Reihen geht, spricht für die Begeisterung der versammelten Apex-Anhänger. Den weitesten Anreiseweg hat Oracle Developer Advocate Connor McDonald – er ist für seine Keynote extra aus Australien angereist. Trotz Jetlag macht er sich hoch motiviert daran, den Teilnehmern die neuesten SQL- und PL/SQL-Features zu vermitteln. Auch wenn der Erscheinungstermin der neuen Apex-Version noch offen ist, verkündet Marc Sewtz aus dem Oracle-Apex-Development-Team allerlei Wissenswertes rund um das kommende Release 18.1. Anschließend gibt er einen Überblick über Oracle JET in Apex 18.1 und stellt die neuen Diagrammtypen vor. Golo Roden, Buch-Autor und Gründer von „the native web“, hält die Fahnne von JavaScript hoch und lässt es sich nicht nehmen, mit gängigen Missverständnissen rund um die Skriptsprache aufzuräumen. 25 Studierende und Auszubildende erhalten durch Sponsoren die Chance, über das NextGen-Programm die Apex Connect kostenlos zu besuchen.



Apex Connect 2018: Die Welle der Begeisterung

23. April 2018

Die Collaborate 2018 findet in Las Vegas statt, organisiert von der Independent Oracle Users Group (IOUG), der Oracle Applications Users Group (OAUG) und der Quest International Users Group. Für die DOAG sind Fried Saacke, Vorstand und Geschäftsführer, sowie Dr. Frank Schönthaler und Kasi Färcher-Haag als Vertreter der Business Solutions Community vor Ort. Im Rahmen des internationalen Empfangs sprechen sie mit OAUG-Präsident Douglas Manning über die zukünftige Zusammenarbeit, nachdem die OAUG die Zusammenarbeit mit der DOAG überraschend gekündigt hat. Eine Lösung ist noch nicht in Sicht.

24. April 2018

Bei der Oracle-Keynote auf der Collaborate 2018 ist gerade mal etwa die Hälfte der insgesamt 4.000 Teilnehmer im Raum. Bemerkenswert für die Veranstaltung, die vor Jahren noch 8.500 Besucher verzeichnete, ist, dass die Cloud-Themen von Oracle nicht auf größtes Interesse stoßen.



Andre Lünsmann (links) im Gespräch mit Stefan Kinnen

„Ein Test mit dem Oracle Advanced Customer Support brachte nicht den erhofften Erfolg ...“

Andre Lünsmann ist verantwortlich für den Oracle-Betrieb bei der Barmenia Versicherungsgruppe. Stefan Kinnen, Vorstandsvorsitzender der DOAG, und Wolfgang Taschner, Chefredakteur des Red Stack Magazin, sprachen mit ihm über seine Erfahrungen.

In welchem Geschäftsbereich ist die Barmenia aktiv?

Andre Lünsmann: Die Barmenia Versicherung ist seit mehr als hundert Jahren als Krankenversicherung tätig. Wir bieten darüber hinaus auch Lebensversicherungen sowie Unfall- und Kfz-Versicherungen an.

Was sind dabei die besonderen Herausforderungen an die IT?

Andre Lünsmann: Gerade alteingesessene Versicherungen stehen heute vor der Herausforderung, ihre Leistungen auch online anzubieten. Im Wettbewerb mit Start-up-Unternehmen ist es wichtig, unsere Services entsprechend dem heutigen Kundenverhalten aufzubauen. Der Türöffner für Versicherungen ist, im Gegensatz zu früher, heute weniger ein persönliches Beratungsgespräch, sondern ein moderner digitaler Service, der auf Webseiten oder in Apps mit wenigen Klicks den Kunden über den Bedarf informiert und daraus ein Beratungsgespräch, einen Chat oder einen Online-Abschluss initiiert. Unser Geschäftsmodell hat sich damit verändert und weiterentwickelt.

Wie lösen Sie diese Aufgaben?

Andre Lünsmann: Die Barmenia hat, wie viele andere Versicherungen auch, in der Vergangenheit natürlich ein umfangreiches Mainframe-System betrieben. Da der Hersteller um die Jahrtausendwende herum den Rechner nicht mehr unterstützte, standen wir vor der Frage, unsere IT neu auszurichten. Ein erster Gedanke war, den Mainframe eines anderen Herstellers einzusetzen, was wir jedoch aufgrund des hohen Aufwands schnell verworfen haben. So fiel in langen Sessions die Entscheidung, ein Online-fähiges System mit entsprechender Web-Oberfläche aufzubauen. Im Zuge dessen haben wir in Kooperation mit einem Partner die Applikationen Schritt für Schritt in Java neu entwickelt. Als Übergangslösung wird der Host auf einer virtuellen Maschine betrieben.

In welchem Bereich kommen Produkte von Oracle zum Einsatz?

Andre Lünsmann: Im Zusammenhang mit der Host-Ablösung haben wir im Jahr 2010 auf der Infrastruktur-Ebene Oracle in Verbindung mit Java und WebLogic sowie auch die Oracle-Datenbank eingeführt. Alle wichtigen Anwendungen laufen heute auf der Oracle-Infrastruktur auf Linux-Basis. Die Virtualisierung erfolgt mittels VMware.

Stellt die aktuelle VMware-Regulierung durch Oracle für Sie ein Problem dar?

Andre Lünsmann: Wir haben im Jahr 2012 eine ULA-Lizenz erworben, die diese Art der Virtualisierung mit abdeckt. Ein Grund

Über die Barmenia Versicherung

Die Barmenia zählt zu den großen, unabhängigen Versicherungsgruppen in Deutschland. Das Produktangebot der Unternehmensgruppe reicht von Kranken- und Lebensversicherungen über Unfall- sowie Kfz-Versicherungen bis hin zu Haftpflicht- und Sachversicherungen. Mehr als 3.400 Innen- und Außendienstmitarbeiter und eine Vielzahl von Maklern betreuen einen Bestand von mehr als 2,2 Millionen Versicherungsverträgen.

für die ULA-Lizenz bestand darin, dass wir bei den vielen Entwicklungs-, Test- und Produktiv-Systemen nicht genau sagen konnten, wie viele CPUs bei uns im Einsatz sind. Als wir im Jahr 2015 zusätzlich die Advanced-Security-Option in Betrieb genommen haben, hat sich die ULA als günstigerer Weg erwiesen.

In welchem Umfang kommen Oracle-Datenbanken bei Ihnen zum Einsatz?

Andre Lünsmann: Wir setzen produktiv rund 180 Oracle-Datenbanken ein. Die Patches erfolgen vollautomatisiert. Der Wechsel auf die Version 12.2 ist derzeit in Vorbereitung.

Wie garantieren Sie die Sicherheit Ihrer Daten?

Andre Lünsmann: Wir haben zu Beginn sehr lange über ein passendes Berechtigungssystem nachgedacht. Im Jahr 2012 wurde dann Enterprise User Security mit eingeführt.

Wie administrieren Sie Ihre Datenbanken?

Andre Lünsmann: Ursprünglich haben wir im Kontext mit der Enterprise Edition den Enterprise Manager eingesetzt, heute nutzen wir Tools von Drittanbietern, darunter eine Workflow-Komponente, um die Abläufe weitgehend zu automatisieren. Auf diese Weise sind wir in der Lage, innerhalb einer halben Stunde eine komplette Oracle-Instanz aufzubauen.

Wie sind Ihre Erfahrungen mit Oracle Support?

Andre Lünsmann: Das ist ein schwieriges Thema. Ursprünglich hatten wir einen externen Dienstleister, der uns damals während der Einrichtungsphase bei auftretenden Problemen unterstützt hat. Nachdem das System live gegangen ist, haben wir gleich mit den ersten Tickets bei Oracle gemerkt, dass das nicht rund läuft. Ein Test mit dem Oracle Advanced Customer Support brachte nicht den erhofften Erfolg, sodass wir wieder einen externen Dienstleister eingebunden haben.

Wie sehen Sie den Umstieg der Patch Set Updates auf die Release Updates?

Andre Lünsmann: Wir haben seit Jahren ein eingespieltes Verfahren etabliert, um Patches auszurollen. Davor bewerten wir die Patch-Matrix, um zu entscheiden, welche Patches wir überhaupt einspielen möchten. Ich glaube nicht, dass uns der Umstieg auf die Release Updates Probleme bereiten wird.

Wie gehen Sie an das Thema „Big Data“ heran?

Andre Lünsmann: Wir beschäftigen uns damit, allerdings ist das Thema nicht in der IT angesiedelt, sondern im Controlling. Wir liefern jede Nacht Auszüge aus den Produktionssystemen in das BI-System, wo sie analysiert und weiterverarbeitet werden.

Welche Rolle spielen Cloud-Anwendungen in Ihrem Unternehmen?

Andre Lünsmann: Die Cloud ist für eine Versicherung in erster Linie ein zentrales Thema für die Rechtsabteilung. Keiner möchte unsere hochsensiblen Daten in einer öffentlichen Cloud sehen. Insofern denken wir eher über Private-Cloud-Lösungen nach. Für mich ist entscheidend, welche Vorteile es künftig bringt, wenn Daten in der Cloud liegen.

Oracle hat gerade in Frankfurt ein deutsches Rechenzentrum für die Cloud eröffnet. Wäre das für Sie eine Option?

Andre Lünsmann: Ein deutscher Standort hört sich zunächst gut an. Als amerikanischer Anbieter gelten allerdings für Oracle nach wie vor die Patriot-Bestimmungen. Solange Oracle keine Zertifizierung nach deutschem Recht vorlegen kann, bringt der Standort in Frankfurt nichts.

Spielen die Oracle-Unternehmenslösungen eine Rolle bei Ihnen?

Andre Lünsmann: Nein.

Wie beurteilen Sie die Zukauf-Strategie von Oracle?

Andre Lünsmann: Ich fände es spannend, wenn Oracle eher kleine, aber technologisch innovative Unternehmen zukaufen würde. Die Frage wäre allerdings, ob Oracle solchen Start-ups entsprechende Freiräume im Unternehmen gewähren würde, um den notwendigen innovativen Fortschritt auch zuzulassen.

Können Sie sich vorstellen, ein Komplettsystem von der Hardware bis zu den Applikationen von einem einzigen Hersteller wie Oracle einzusetzen?

Andre Lünsmann: Das kann ich mir eher nicht vorstellen, da wir auch stark auf Open-Source-Produkte und eigenentwickelte Komponenten setzen.

Wäre der Einsatz eines Oracle Engineered System bei Ihnen denkbar?

Andre Lünsmann: Wir haben Anfang 2017 eine Private-Cloud-Appliance evaluiert. Das hätte allerdings keine Verbesserungen im Vergleich zu unseren bestehenden Systemen gebracht.

Sehen Sie Oracle strategisch primär als Anbieter von Engineered Systems oder von Cloud-Lösungen?

Andre Lünsmann: Für mich ist das Cloud-Thema deutlich stärker. Es macht für mich aus Oracle-Sicht wesentlich mehr Sinn, dort unterwegs zu sein. Darüber hinaus finde ich es gut, dass Oracle für die Cloud-Lösungen die eigenen Produkte einsetzt, was dazu führt, deren Leistung und Qualität besser erfahren zu können.

In welche Richtung wird sich Ihre IT in den kommenden Jahren entwickeln?

Andre Lünsmann: Ich hoffe ganz stark, dass wir uns deutlich mehr in Richtung Private-Cloud- und Service-Dienstleister entwickeln. Wir arbeiten intensiv daran, unsere Prozesse weiter zu automatisieren, und evaluieren laufend neue Werkzeuge, um eine effizientere Administration zu erreichen. Gleichzeitig bieten wir auch unseren Mitarbeitern die Möglichkeit, sich entsprechend weiterzubilden und in moderne Technologien einzuarbeiten.

Welche Unterstützung erwarten Sie dabei von einem IT-Unternehmen wie Oracle?

Andre Lünsmann: Ich denke, dass sich Oracle in den nächsten Jahren technologisch weiterentwickeln wird. Vor allem muss das Unternehmen eine deutliche Verbesserung der Produktqualität schaffen, um seinen Kunden signifikante Vorteile im Vergleich zur Konkurrenz zu bieten.



Zur Person: Andre Lünsmann

Andre Lünsmann (54) ist seit seiner Ausbildung zum Versicherungskaufmann und der Weiterbildung zum Versicherungsfachwirt in den 1980-Jahren bei den Barmenia Versicherungen beschäftigt. Nach einer etwa einjährigen Ausbildung zum Organisations-Entwickler wechselte er in die Anwendungs-Entwicklung und war dort als Entwickler und später als Projektleiter in diversen Großprojekten tätig. Seit dem Jahr 2009 ist er Teamleiter und in wechselnden Aufgaben für Datenbanken, Systemmanagement und dezentrale Produktionsaufgaben verantwortlich. Nach der erfolgreichen Host-Ablösung der IBM im Jahr 2010 ist er unter anderem für die Oracle-Datenbanken verantwortlich. Seit dem Jahr 2012 ist er Vertreter der Barmenia im Kreis der DOAG und dort unter anderem als Mitglied des Anwenderbeirats aktiv.

Wie sehen Sie den Stellenwert der DOAG?

Andre Lünsmann: Die DOAG ist für mich unbestritten eine sehr professionelle und vielschichtige Organisation. Ich bin jetzt seit fünf Jahren regelmäßig auf der Jahreskonferenz, was für mich gut investierte Zeit ist, vor allem wenn ich an die vielen Vernetzungsmöglichkeiten denke.

Wie steht es um Ihre Arbeit im Anwenderbeirat der DOAG?

Andre Lünsmann: Ich halte die Einrichtung des Anwenderbeirats für sehr gut, weil wir hier die Erfahrungen vieler Kunden austauschen können, die in den laufenden Dialog mit Oracle einfließen.



Aufgeklärt – ein Überblick über Cloud Services

Robert Marz, its-people GmbH

Wir sind im Jahr 2018 und die Cloud ist allgegenwärtig. Im privaten Umfeld fällt es uns schon gar nicht auf, wie häufig wir die Cloud benutzen: E-Mails checken? Die liegen in der Cloud. Die Social-Media-Accounts sowieso. Alexa oder einen anderen digitalen Assistenten etwas fragen? Die Spracherkennung funktioniert nur durch die Cloud und die Antworten werden von unzähligen Cloud Services zusammengestellt.

Wie ist die aktuelle Verkehrslage? Die Antwort kommt natürlich aus der Cloud und ist so präzise nur möglich, weil unzählige Verkehrsteilnehmer (genauer: deren Smartphones) ihre aktuellen Bewegungsvektoren in die Cloud schicken. Bevor wir überhaupt am Arbeitsplatz angekommen sind, haben wir die Cloud schon ausgiebig benutzt. Und im Job tun

wir dies nicht weniger, sogar bei den Unternehmen, die nach eigener Auffassung noch gar nichts in die Cloud migriert haben: Die Mitarbeiter benutzen in der Regel, was ihnen hilft, Dinge erledigt zu bekommen. Einige Anwendungen, wie E-Mail, GitHub, Trello oder Dropbox, werden gar nicht mehr als Cloud-Dienste wahrgenommen.

Sicherheit und Datenschutz

Sicherheit war lange Zeit ein Grund, sich aus der Cloud fernzuhalten. Das hat sich stark verändert: Laut einer Umfrage von Coleman Parkes aus dem Jahr 2016 sind 78 Prozent aller Unternehmen der Meinung, dass sich die Sicherheit insgesamt durch den Umzug in eine Public Cloud erhöhen könnte. Die EU-

Datenschutz-Grundverordnung (DSGVO) bestärkt diesen Trend. Schließlich bieten alle Cloud-Anbieter reihenweise Zertifikate an, die DSGVO-Konformität bestätigen. Solche Zertifikate für On-Premises-Anwendungen zu erlangen, ist aufwendig.

Autonome Datenbanken

Alle spannenden Neuentwicklungen finden mittlerweile in der Cloud statt. Der Autonomous Data Warehouse Cloud Service (ADWC) ist ein gutes Beispiel dafür. Von den Experten und DBAs wurde er zunächst belächelt: „Der ADWC hat mehr Einschränkungen als Features – ein Data Warehouse ohne Partitionierung, kann das funktionieren?“ Wie autonom die Datenbank wirklich ist, werden wir erst noch sehen.

Aus Sicht von Oracle sind DBAs jedoch nicht die Zielgruppe für ADWC. Die Fachabteilungen, die mal eben ein Data Mart für ein paar Auswertungen bauen wollen, sind froh, wenn sie das mit kleinem Budget ohne übermäßiges Involvieren der IT-Abteilung bewerkstelligen können; die bestehenden Einschränkungen werden Schritt für Schritt reduziert. Die Cloud ermöglicht wesentlich kürzere Release-Zyklen.

Software-Entwicklung

Der Software-Entwicklungsprozess kann durch Cloud Services erheblich beschleunigt werden. Die Oracle Developer Cloud Services bündeln gängige Open-Source-Tools zu einer Suite zusammen. Egal ob Kanban-Board, Git-Repository, Ticketing-

System oder Build-Automatisierung: Alles ist vorkonfiguriert und betriebsbereit. Backup und Recovery erledigt die Cloud. Das ist sehr bequem und ermöglicht es, sich ganz auf die Software-Entwicklung zu konzentrieren. Alle großen Cloud-Anbieter stellen ähnliche Services zur Verfügung. Das führt dazu, dass die Cloud für verteilte Software-Entwicklungsteams immer häufiger zur ersten Wahl wird.

DevOps und Serverless

DevOps ist einer der aktuell großen Trends. Unternehmen leiten den nötigen kulturellen Wandel ein und durchbrechen die Mauer zwischen Dev und Ops. Das agile Vorgehen, das in kleine Schritten schnelle Ergebnisse liefern soll, bringt immer kleinere Komponenten hervor, die unabhängig voneinander entwickelt und betrieben werden, aber erst zusammen die eigentliche Applikation bilden. Diese Microservices können als Container oder „serverless“ betrieben werden. Der Serverless-Betrieb von Microservices war lange nur in der Cloud denkbar – bis Oracle das Fn Project vorgestellt hat. Die „container native serverless platform“ ist Open Source und läuft sowohl in der Cloud als auch On-Premises.

Container-Lösungen wie Docker sind ein Segen: Einfach zu verteilen, einfach zu starten oder zu beenden. Alle abhängigen Bibliotheken werden in der richtigen Version mit ausgeliefert. Die Applikation im Nachbar-Container ist nicht beeinträchtigt. Für den produktiven Betrieb von Docker-Containern hat sich der Container-Cluster-Manager Kubernetes durchgesetzt.

Eine zentrale Säule von DevOps ist die Automatisierung aller Vorgänge sowohl beim Betrieb als auch beim Ausrollen von Komponenten. Dazu gehört auch, dass die benötigten virtuellen Maschinen und Netzwerke automatisch erzeugt und wieder abgerissen werden. Das nennt sich dann „Software Defined Infrastructure“. Die Web-Oberflächen der Cloud-Anbieter stehen dem entgegen. Aus diesem Grund bieten alle Anbieter proprietäre REST- und Commandline-Interfaces an. Automatisierungstools wie Ansible oder Terraform vereinheitlichen deren Ansteuerung.

Fazit

Cloud Services sind vielfältig und aus der heutigen Zeit nicht mehr wegzudenken. Es gibt weit mehr, als in diesem Artikel beschrieben sind. Mit Sicherheit werden neue hinzukommen und bestehende mit Getöse oder unbemerkt wieder verschwinden. Niemand weiß, was die Zukunft bringt. Aber sie ist mit Sicherheit wolkig.



Robert Marz
robert.marz@its-people.de

Schwachstelle bei Kernel und Hypervisor

Entwickler fast aller PC-Betriebssysteme – inklusive Windows, macOS, Linux und FreeBSD – haben auf Grund eines Missverständnisses einen Fehler in ihren Code eingebaut, über den ein lokaler Angreifer Kernel-Speicher auslesen und unter Umständen manipulieren kann. Das könnte zum Ausführen böser Codes mit System-Rechten führen. Die genauen Auswirkungen der Schwachstelle hängen vom Betriebssystem

und dem Kernel ab. Davon betroffen sind auch die Oracle-Linux-Plattformen.

Als Lösung für verschiedene Oracle Linux 5-, 6- und 7-Versionen sowie Red Hat Enterprise Linux-Server- und Red Hat Enterprise Linux Server-Support-Versionen stehen verschiedene Sicherheitsupdates zur Verfügung (siehe „<https://adv-archiv.dfn-cert.de/adv/2018-0881>“).



Oracle Database 18c und Database as a Service

Tobias Deml, ORACLE Deutschland B.V. & Co. KG

Die neu angekündigte Datenbank-Version 18c steht bereits seit einigen Monaten in der Oracle-Cloud zur Verfügung. Das On-Premises-Release wird demnächst folgen. Inhaltlich hat sich bei dieser Version in vielerlei Bereichen etwas getan. Der Artikel stellt die Neuerungen vor und gibt eine kurze Anleitung zur Erstellung eines Oracle Database Cloud Service.

Hinsichtlich der in der Version 18c eingeführten Neuerungen existiert bereits eine Vielzahl von Präsentationen, Vorträgen und Whitepaper. Deshalb geht der Autor nicht auf jede einzelne neue Funktionalität ein, sondern zeigt etwas kleinere, aber dennoch sehr nützliche Verbesserungen auf, die die Nutzung einer Oracle-Datenbank 18c attraktiv machen.

Die neuen Features gliedern sich in die vier Bereiche „Performance“, „Multitenant“, „Security“ und „Data Warehouse & Big Data“ (siehe Abbildung 1). Wenn man die darunter gelisteten Funktionalitäten und zusätzlich Dokumentationen genauer betrachtet, fallen zwei Trends auf: einerseits der Fokus auf die Weiterentwicklung der Development-Funktionalitäten und andererseits der Ausbau von Technologien zur Abbildung agiler Entwicklungsmethoden.

Der Artikel geht auf folgende kleinere, aber sehr interessante Neuerungen in der Datenbank-Version 18c ein:

- Alter Session Cancel SQL
- Verschlüsselung von Anmelde-Informationen im Data Dictionary
- Inline External Tables
- Private Temporary Tables

Alter Session Cancel SQL

Jeder kennt diese Situation, dass ein Benutzer oder gar man selbst ein SQL auf der

Datenbank absetzt, das irrtümlicherweise ein viel zu großes Ergebnis zurückliefert. Aufgrund dessen liegt nun eine Session mit einem Langläufer-SQL auf der Datenbank und am naheliegendsten wäre es, die Ausführung dieses Statements abbrechen. Bisher war das Mittel der Wahl, die Session mit einem „Alter System Kill Session“-Befehl abbrechen, was aber diverse Seiteneffekte zur Folge hat.

Die Datenbank-Version 18c bietet eine neue Funktionalität zum besseren Handling dieser Situationen. Nun ist es möglich, lediglich das ausgeführte SQL abbrechen, ohne die darunterliegende Session zu beenden. Das entsprechende SQL-Kommando zur Durchführung dieser Aktion lautet „ALTER SYSTEM CANCEL SQL ,sid,

serial#, [instance_id], [sql_id];“ . Die beiden letzten Übergabe-Parameter sind optional beziehungsweise von der eingesetzten Datenbank-Technologie abhängig. Die von der Beendigung des SQL betroffene Verbindung wird dabei mit einer Meldung über die Aktion benachrichtigt (siehe Listing 1).

Verschlüsselung von Anmelde-Informationen im Data Dictionary

Die IT der heutigen Zeit ist stark beeinflusst von Security- und Compliance-Regularien. Ein Teil dieser Regelwerke ist der Schutz von Zugangsdaten vor unberechtigter Benutzung durch Dritte. Ab der Version 18c

Performance • Low Latency Memory Lookups • Non Volatile Memory Support • In-Memory Column Store Improvements		Multitenant • Per-PDB Switchover • Transportable Backups • Snapshot Carousel • Faster Upgrades	
Security • Integration with Active Directory • Per PDB Key storage • Password-less schema creation • No default passwords		Data Warehouse & Big Data • Automatic propagation of nologged data to standby • Polymorphic Table Functions • Alter Table Merge Partition Online • Approximate Query Improvements	

Abbildung 1: Größte Neuerungen der Oracle-Datenbank 18c

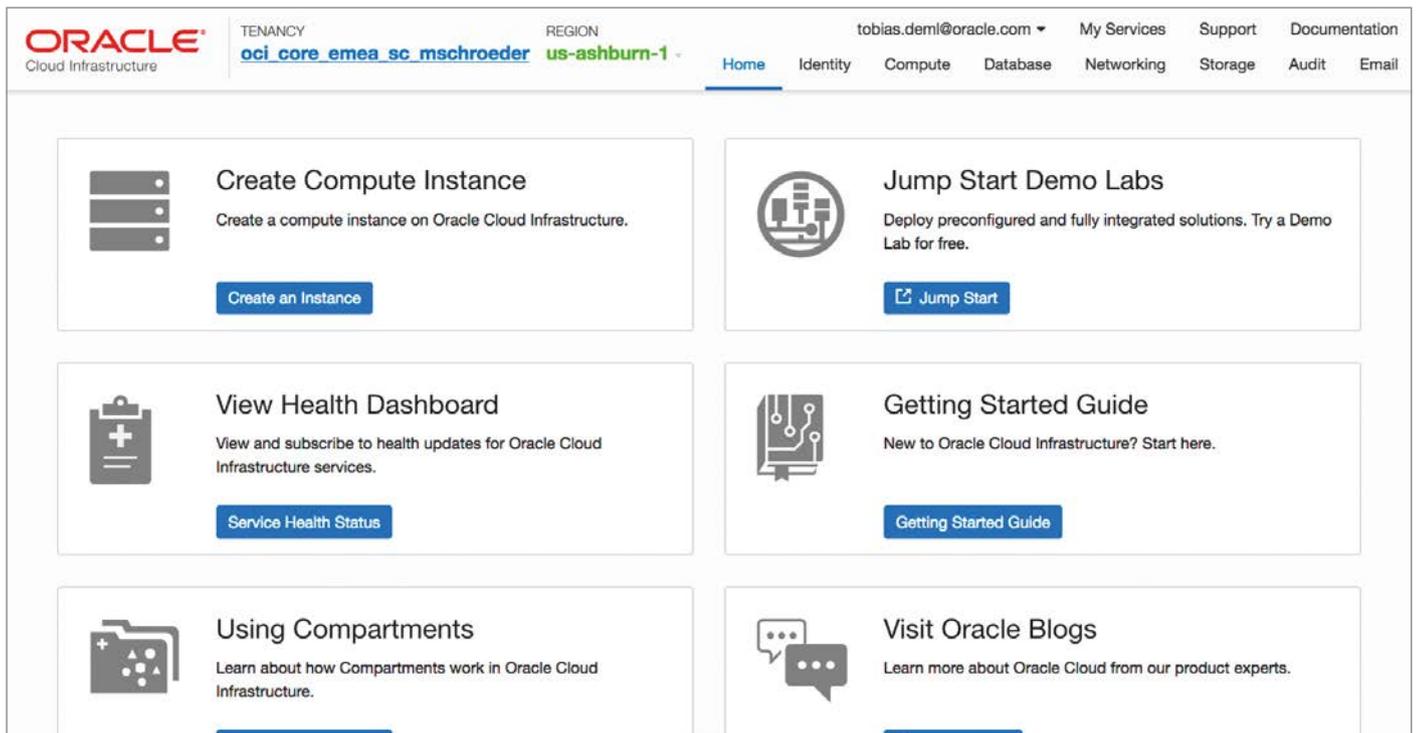


Abbildung 2: Start-Bildschirm der Oracle Cloud Infrastructure

ist es möglich, das gesamte Data Dictionary mithilfe der Transparent Data Encryption (TDE) zu verschlüsseln. Diese Funktionalität verschlüsselt unter anderem auch die in Database-Links und im „DBMS_Scheduler“ hinterlegten Passwörter.

Wer bereits in seiner Datenbank TDE einsetzt, kann dieses Feature mit dem SQL-Statement „ALTER DATABASE DICTIONARY ENCRYPT CREDENTIALS;“ aktivieren. Anschließend werden per Abfrage der Tabellen „SYS.LINK\$“ und „SYS.SCHEDULER\$_CREDENTIAL“ die verschlüsselten Zugangsdaten eingesehen und deren Verschlüsselung geprüft. Da für die Entschlüsselung dieser Daten der Data Encryption Key notwendig ist, muss bei Abfrage eines Datenbank-Links oder Start eines Scheduler-Jobs das Wallet der Datenbank den Status „OPEN“ aufweisen.

Inline External Tables

Die Nutzung der „External Table“-Funktionalität ist für Entwickler und Daten-Integratoren kaum mehr wegzudenken. Besonders die Aggregation strukturierter Daten aus der Datenbank und weniger strukturierter Daten von außerhalb sind beliebte Anwendungsfälle. Oft ist es allerdings schwierig, die Definition der External-Table auf dem aktuellen Stand

zu halten, da sich der Aufbau der externen Daten schnell ändern kann. Dafür bietet die neue Datenbank-Version eine

Funktionalität namens „Inline External Tables“, die genau auf diese Thematik abzielt.

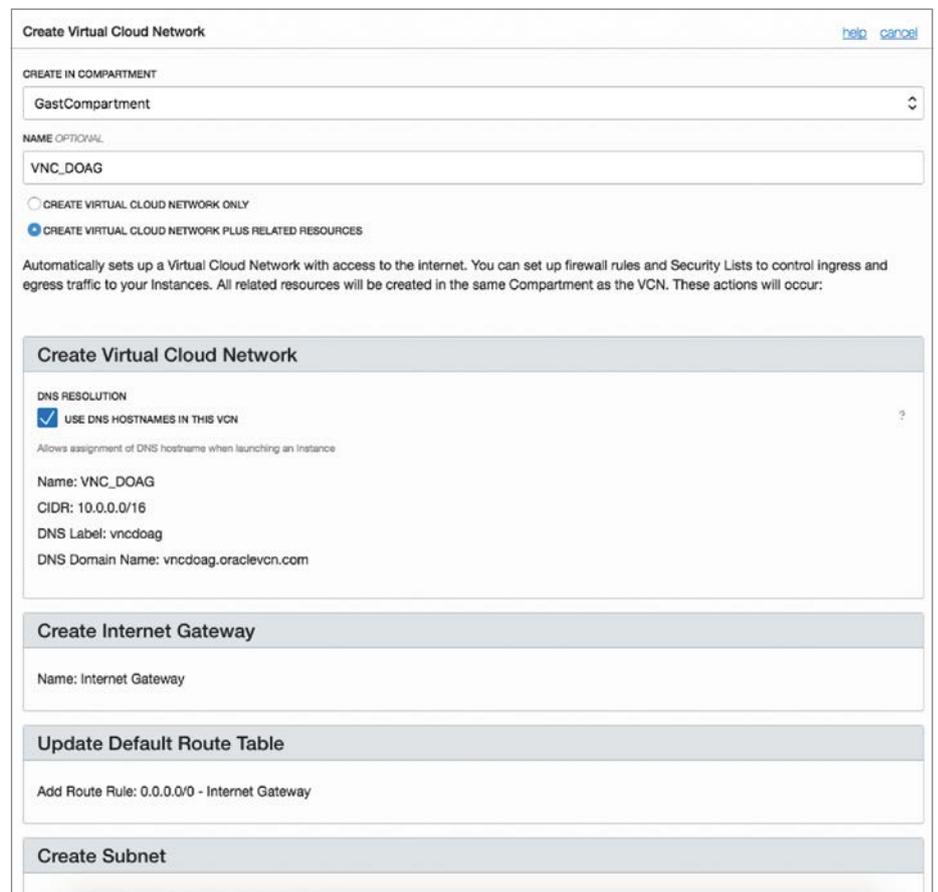


Abbildung 3: VCN Creation Wizard

Bei diesem Feature lässt sich beim Absetzen des SQL-Statements bestimmen, welche Form die externen Daten besitzen. Dazu muss keine „External Table“-Defini-

tion mehr angelegt sein, lediglich ein Directory-Eintrag mit dem Pfad zur externen Datenquelle ist notwendig (siehe Listing 2). Zwei Beispiele zeigen, wie dieses Feature

anhand von Beispieldaten funktioniert (XSL-Daten außerhalb der Datenbank siehe Listing 3 und „Inline“-Definition der externen Tabelle siehe Listing 4).

DB System Information

DISPLAY NAME: RedStackDOAGMagazin

AVAILABILITY DOMAIN: TBOz:US-ASHBURN-AD-1

SHAPE TYPE: VIRTUAL MACHINE BARE METAL MACHINE

SHAPE: VM.Standard2.2

TOTAL NODE COUNT: 1

ORACLE DATABASE SOFTWARE EDITION: Enterprise Edition Extreme Performance

AVAILABLE STORAGE SIZE (GB): 256

Scale up the available storage size for DB System up to 40960 GB.

TOTAL STORAGE SIZE (GB): 712

The total storage consumed by the DB system. Oracle charges for the storage consumed.

LICENSE TYPE: LICENSE INCLUDED
The cost of the cloud services includes the Oracle licensing.

BRING YOUR OWN LICENSE (BYOL)
You have bought the Oracle licenses directly from Oracle. The cloud provider is not responsible for charging or validating your licenses.

SSH PUBLIC KEY: CHOOSE SSH KEY FILES PASTE SSH KEYS

Choose SSH Key files (.pub) from your computer:

ssh_key.pub

Browse

Abbildung 4: Erstellung des Datenbank-Service I

Network Information

VIRTUAL CLOUD NETWORK: VNC_DOAG

CLIENT SUBNET: Public Subnet TBOz:US-ASHBURN-AD-1

HOSTNAME PREFIX: DOAG

HOST DOMAIN NAME: sub04170911350.vncdoag.oraclevcn.com

Each part must contain only letters and numbers, starting with a letter. 63 characters max.

HOST AND DOMAIN URL: DOAG.sub04170911350.vncdoag.oraclevcn.com

Abbildung 5: Erstellung des Datenbank-Service II

Private Temporary Tables

Bei der Datenverarbeitung oder datenbankseitigen Entwicklung werden gerne „Global Temporary Tables“ zur Pufferung voraggregierter Daten verwendet. Diese temporären Datenstrukturen stehen aber, wie der Name es vermuten lässt, global jedem Benutzer der Datenbank zur Verfügung. Es gibt Anwendungsfälle, bei denen dies aus Sicht von Security-Anforderungen ein Problem darstellt. Die neue Datenbank-Version 18c stellt dafür mit dem Feature „Private Temporary Tables“ (PTT) eine Lösung bereit. Diese PTT besitzen unter anderem folgende Eigenschaften:

- Die Struktur und der Inhalt dieser Tabelle sind nur für eine Session sichtbar
- Bei Erstellen dieser Tabelle wird keine Definition im Data Dictionary abgelegt
- Das Create-Statement dieser Tabelle löst kein implizites Commit aus

Weil aus der Erstellung von PPT kein Eintrag in das Data Dictionary resultiert, lässt sich diese Funktionalität auch in einer Active-Data-Guard-Umgebung nutzen. Das Beispiel in Listing 5 zeigt die Erstellung und das Handling von Private Temporary Tables inklusive der Verwendung des notwendigen Präfixes. Genauere Informationen zu den vorgestellten Features stehen im deutschsprachigen Oracle-Blog [1].

Oracle Database as a Service (DBaaS)

Als Nächstes folgen die Schritte, um sich eine eigene Oracle-Datenbank-Umgebung in der Cloud zu erstellen. Nach dem Einloggen seines Accounts in die Oracle-Cloud-Tenancy erscheint zum Start eine Übersicht (siehe Abbildung 2). Am linken oberen Rand stehen Informationen zum Namen der Tenancy und der entsprechend zugeordneten Oracle-Cloud-Region. Rechts oben sind unter anderem verschiedene Auswahlmöglichkeiten zu den angebotenen Services und der Verwaltung der Cloud-Accounts zu finden.

Database Information

DATABASE NAME

DATABASE VERSION

PDB NAME (Optional)

DATABASE ADMIN PASSWORD

Password must be 9 to 30 characters and contain at least 2 uppercase, 2 lowercase, 2 special, and 2 numeric characters. The special characters must be _ , #, or -.

CONFIRM DATABASE ADMIN PASSWORD

Confirmation must match password above.

AUTOMATIC BACKUP
 Configure the service to automatically back up this database to Oracle Cloud Infrastructure Object Storage.

If you previously used RMAN or dbcli to configure backups and then you switch to using the Console or the API for backups, a new backup configuration is created and associated with your database. This means that you can no longer rely on your previously configured unmanaged backups to work.

DATABASE WORKLOAD

ON-LINE TRANSACTION PROCESSING (OLTP)
 Configure the database for a transactional workload, with bias towards high volumes of random data access.

DECISION SUPPORT SYSTEM (DSS)
 Configure the database for a decision support or data warehouse workload, with bias towards large data scanning operations.

[Show Advanced Options](#)

Abbildung 6: Erstellung des Datenbank-Service III

Launch DB System

<p>RedStackDOAGMagazip Availability Domain: TBOz:US-ASHBURN-AD-1 OCID: ...n5n32q Show Copy</p>	<p>Oracle Database Software Edition: Enterprise Edition Extreme Performance Shape: VM.Standard1.2</p>	<p>Virtual Cloud Network: VNC_DOAG Client Subnet: Public Subnet TBOz:US-ASHBURN-AD-1 Private IP: Loading... Public IP: Loading... Available Data Storage: 256 GB Total Storage Size: 712 GB</p>	<p>Launched: Tue, 17 Apr 2018 09:16:54 GMT</p>
--	--	--	--

Abbildung 7: Übersicht über die Datenbank-Systeme

```

dhcp-10-175-220-54:~ tobiasdeml$ ssh -i /Users/tobiasdeml/ssh_key opc@129.213.35.124
Last login: Tue Apr 17 11:11:32 2018 from 156.151.8.10
[opc@doag ~]$
[opc@doag ~]$
[opc@doag ~]$ sudo su - oracle
[oracle@doag ~]$
[oracle@doag ~]$
[oracle@doag ~]$ . oraenv
ORACLE_SID = [oracle] ? ORCL
The Oracle base has been set to /u01/app/oracle
[oracle@doag ~]$
[oracle@doag ~]$
[oracle@doag ~]$ sqlplus / as sysdba

SQL*Plus: Release 18.0.0.0.0 Production on Tue Apr 17 11:15:50 2018
Version 18.1.0.0.0

Copyright (c) 1982, 2017, Oracle. All rights reserved.

Connected to:
Oracle Database 18c EE Extreme Perf Release 18.0.0.0.0 - Production
Version 18.1.0.0.0

SQL>
SQL>
    
```

Abbildung 8: Den Test starten

Zur Erstellung eines Datenbank-Service muss vorher ein sogenanntes „Virtual Cloud Network“ (VCN) existieren, dass im folgenden Schritt erstellt wird. Über die Menüpunkte „Networking“ und „Virtual Cloud Networks“ erreicht man die notwendige Oberfläche, um über die blaue Schaltfläche „Create Virtual Cloud Network“ die zuvor beschriebene Netzwerk-Komponente zu erstellen. Im nachfolgenden Creation-Wizard ist dann, nach Auswahl der Option „Create Virtual Cloud Network Plus Related Resources“, lediglich noch der Name zu ergänzen und schon lässt sich mit der Betätigung der blauen Schaltflächen die VCN-Erstellung starten (siehe Abbildung 3).

Über die Menüpunkte „Database“ und „Systems“ erreicht man die Übersicht des

Sämtliche hier erwähnten Listings finden Sie unter folgendem Link:

<http://www.doag.org/go/RS318/TobiasDeml>

Datenbank-Service. Dort lässt sich über die blaue Schaltfläche „Launch DB System“ der Erstellungsassistent für die Datenbank aufrufen. In diesem Wizard gibt es drei Teilbereiche, in denen die Informationen zu Datenbank-Umgebungen hinsichtlich Benennung, Platzierung und Größe angegeben werden (siehe Abbildungen 4 bis 6).

Für zusätzliche Informationen zur Erstellung des Datenbank-Service kann man über die Schaltfläche „help“ direkt an die richtige Stelle in der Oracle-Cloud-Dokumentation springen. Sobald alle

notwendigen Felder der Maske ausgefüllt sind, startet der Button „Launch DB System“ die Erstellung der Services. Auf der DBaaS-Übersichtsseite erscheint dazu ein Eintrag (siehe Abbildung 7).

Dieser soeben erstellte Dienst hat nun den Status „PROVISIONING...“ (oranges Symbol). Sobald er zur Verfügung steht, wechselt der Status auf „AVAILABLE“ (grünes Symbol). Anschließend kann man sich mit dem Service verbinden und die Tests der neuen Datenbank-Version 18c starten (siehe Abbildung 8).

Weitere Informationen und Links

- [1] Deutschsprachiger Oracle-Blog zur Database 18c: <https://blogs.oracle.com/coretec/datenbank-18c>
- [2] Deutschsprachiger Oracle-Blog – Thema Oracle Cloud: <https://blogs.oracle.com/coretec/cloud-16>
- [3] Dokumentation Oracle Database 18c: <https://docs.oracle.com/en/database/oracle/oracle-database/18/index.html>

- [4] Dokumentation Oracle Database Cloud Service: <https://docs.us-phoenix-1.oraclecloud.com/Content/Database/Concepts/databaseoverview.htm>



Tobias Deml
tobias.deml@oracle.com

Ihre Open Source Infrastruktur mit einem Klick!



Infrastructure at your Service.



Consulting · Service Management (SLAs) · Lizenzmanagement · Workshops
Phone +41 32 422 96 00 · Basel · Nyon · Zürich · dbi-services.com

Über den Wolken – grenzenlose Freiheit mit purem Metall?

Katharina BÜchse und Florian Barth, Robotron Datenbank-Software GmbH

Auch Metall kann fliegen, es benötigt nur hinreichend viel Antrieb, um die Masse in die Lüfte zu heben. Solch simplen physikalischen Gesetzen ist sich Oracle durchaus bewusst und macht entsprechend Wind um das neue Konzept, das sie im Herbst 2016 in die Cloud gebracht haben: Bare Metal.

Bare Metal verspricht die eigenen vier Wände (quasi den Privatjet) für Daten, ganz ohne nervige Nachbarn. Allerdings muss man sich als Kunde erst einmal durch die Check-ins der geänderten Kostenmodelle, stetigen Umgestaltungen oder einfach nur Umbenennungen der Services und zu guter Letzt sogar der Lizenzbedingungen kämpfen, bis man endlich Business Class fliegen kann. Dieser Artikel bietet dafür einen kleinen Flughafen-Guide, da das neue Cloud-Konzept von Oracle Aufmerksamkeit verdient hat. Selbst wenn oder gerade weil der Wind um die Cloud auf Oracle-Kunden möglicherweise wie Gegenwind wirkt.

Die lieben Nachbarn

Zwanzig Jahre, nachdem Reinhard May zum ersten Mal von der Freiheit über den Wolken geträumt hatte, sang er: „Ob im größten Saal, ob im kleinsten Zimmer – irgendein Depp bohrt irgendwo immer!“. Es ist eben nicht so einfach mit den lieben Nachbarn, sei es im Mietshaus oder in den Wolken. Während man sich als Mieter über die fehlende Sonntagsruhe oder scheinbar nicht funktionierende Heizungen (weil das Warmwasser ja gerade von allen anderen zum Duschen genutzt wird) aufregt, streiten sich die Daten in der Cloud um so wertvolle Ressourcen wie Rechenleistung, Hauptspeicher und Netzwerk-Kapazität. Wie der Streit ausgeht, zeigt sich dann spätestens in der Performance der Anwendung.

Das muss jedoch gar nicht sein, denn Oracle bietet einem nun die Möglichkeit,

sich quasi nicht nur eine Wohnung, sondern ein ganzes Haus in den roten Wolken zu mieten. Sämtliche dort abgelegten Daten wohnen dann auch wirklich komfortabel, denn zum Einsatz kommen Oracle Engineered Systems in Form von ODA und Exadata. Zudem hat Oracle passenderweise auch gleich noch die entsprechenden Rechenzentren in Frankfurt am Main eröffnet, weil auch Daten sicherlich nur sehr ungern aus Deutschland wegziehen wollen, ist ja schließlich schön hier.

Namen sind wie Schall und Rauch

Die erste Herausforderung, der man sich stellen muss, um zu einer eigenen Bare-Metal-Maschine in der Oracle-Cloud zu kommen, ist der Weg durch das Labyrinth der Bezeichnungen für die von Oracle angebotenen Cloud Services (diese Formulierung ist übrigens extra ein bisschen umständlich, nur so als kleinen Vorgeschmack auf das, was jetzt kommt). Als Oracle das neue Konzept einführte, kam damit eine ganze Oracle-Bare-Metal-Cloud auf den Markt. Diese Bezeichnung gibt es jetzt nicht mehr. Oracle benennt nicht mehr ganze Wolken danach, wie die Services darin aufgebaut sind. Das liegt wohl unter anderen daran, dass schon vor der Umbenennung auch in der Bare-Metal-Cloud virtuelle Maschinen angelegt werden konnten. Die Ironie dahinter lässt sich sehr schnell erkennen. Nun lautet der neue Name „Oracle Cloud Infrastructure“ (OCI). Er suggeriert bereits, dass man es

hier nicht nur mit einzelnen, in sich abgeschlossenen Services zu tun hat, sondern sich stattdessen in der Cloud eine ganze Infrastruktur aufbaut.

Was aber ist aus der alten Oracle-Cloud geworden? Diese hieß zuerst „Elastic Cloud“, dann war die Rede von „Classic Cloud“, aber „Classic“ klang anscheinend zunächst zu altbacken, weswegen zeitweise nur noch von Cloud die Rede war. Zur Unterscheidung der einzelnen Services war aber „Classic“ noch in Gebrauch, allerdings auch nicht durchgängig, teilweise musste man aufgrund des fehlenden „Bare Metal“ im Namen auf Classic schließen. Jetzt ist die Bezeichnung „Classic“ für die gesamte alte Cloud wieder Mode, aber diesmal in Form von „OCI Classic“.

Dabei wird sichtbar, dass sich beide Wolkenarten immer stärker aneinander annähern, das Komplizierte daran ist aber, dass sie doch noch sowohl physisch als auch logisch voneinander getrennt sind. Physisch, weil in den von Oracle aufgebauten Rechenzentren bisher nur entweder Classic oder Bare Metal Services zur Verfügung stehen, und logisch, weil für die Verwaltung unterschiedliche Tools angeboten werden. Selbst die Web-GUI sieht vollkommen anders aus. Da sich aber Cloud-Web-GUIs sowieso ständig ändern, fällt das wahrscheinlich noch am wenigsten auf.

Der Kostenpunkt

Auch beim Thema „Kosten“ ändert Oracle in letzter Zeit gern mal etwas. Hat man im Sommer noch gegrübelt, was denn nun

der Unterschied zwischen „metered“ und „non metered“ ist, lautet nun das neue Schlagwort „Universal Credits“. Das Tolle an denen ist, dass sie nicht mehr an bestimmte Services gebunden sind, sondern universell eingesetzt werden können. Man bezahlt einfach für ein Jahr einen festen Betrag und hat dann jeden Monat entsprechend ein Zwölftel davon zur freien Verfügung. Zum Vergleich: Bei „non metered“ musste man sich genau festlegen, welche Services man im nächsten Jahr brauchen wird. Allerdings gibt es das eigentliche „metered“-Prinzip nicht mehr, dass man nämlich einfach für ein Jahr Credits kauft und diese dann das ganze Jahr über zur Verfügung hat. Stattdessen verfallen ungenutzte Universal Credits jeweils am Ende des Monats, und wenn man mal in einem Monat mehr Credits benötigt, als man Universal Credits gebucht hat, wird das als „pay as you go“ abgerechnet und man bekommt eine Rechnung von Oracle.

Wer sich für Bare Metal entscheidet, muss zudem beachten, dass die ersten beiden OCPUs einer Bare-Metal-Maschine anders abgerechnet werden als deren restliche OCPUs. Die Begründung dafür ist relativ einfach: Die ersten beiden OCPUs müssen den Betrieb der gesamten Maschine und sämtliches Cloud-Tooling, das man verwendet, finanzieren. Man mietet damit schließlich das gesamte Haus und muss dafür auch sämtliche Nebenkosten tragen, selbst wenn man die Hälfte der Räume leer stehen lässt. Aber was tut man nicht alles, um nachts in Ruhe durchschlafen zu können.

Um zu vermeiden, dass jetzt die Oracle-Sales-Mitarbeiter diese Zeitschrift unzufrieden zur Seite legen, weil hier beim Thema „Kosten“ das überhaupt Wichtigste und Beste fehlt, sei noch erwähnt, dass Oracle jetzt auch „Bring Your Own Licence“ (BYOL) in sein Programm aufgenommen hat. Kommen Sie also in die Cloud, Ihre Lizenz ist (ja) schon da.

Oracle Cloud Infrastructure – der Name ist Programm

Neben der Abwesenheit störender Nachbarn ist das Besondere an der neuen Oracle Cloud, dass man sich hier eine ganz persönliche Netzwerk-Struktur aufbauen kann (und in einem gewissen Grad

auch muss). *Abbildung 1* zeigt ein Beispiel für so ein wolkiges Netzwerk.

Den Rahmen bildet hier ein sogenanntes „Virtual Cloud Network“ (VCN), in dem wiederum Subnetze angelegt werden. In den Subnetzen werden dann die eigentlichen Instanzen platziert. Hierbei steht der Begriff „Instanz“ allerdings nicht für Datenbank-Instanz, sondern für einen Host. Das kann eine ganze Bare-Metal-Maschine, das kann aber auch nur eine virtuelle Maschine sein. Auch Datenbank-Systeme, sei es auf Bare Metal oder in einer VM, werden in so ein Subnetz platziert.

Bei den Subnetzen kann man sich übrigens noch aussuchen, ob sie „private“ oder „public“ sein sollen. Damit gibt man vor, ob es in diesem Subnetz möglich sein soll, einer Instanz bei ihrer Erstellung eine öffentliche IP-Adresse zuzuweisen. Um das vollkommen auszuschließen, macht man das Subnetz einfach „private“.

Geringfügig feingranularer lässt sich der Zugriff noch über die sogenannten „Security Lists“ steuern. Selbige fungieren als Firewall, bei der man einstellt, wer auf welchen Port innerhalb des Subnetzes Zugriff haben soll. Damit kann man das Subnetz übrigens auch vollkommen abschotten, und trotzdem bietet Oracle die Möglichkeit, auf einzelne Maschinen dieses Subnetzes auch noch aus einem anderen Subnetz zuzugreifen. Diese Funktionalität wird durch sogenannte „sekundäre Virtual Network Interface Cards“ (VNIC) bereitgestellt. Die Instanzen bekommen also einfach virtuelle Netzwerkkarten, von denen sich mehrere anlegen lassen, und das eben auch in andere Subnetze. Nur in ein anderes Subnetz umziehen kann man damit seine Instanzen leider noch nicht, weil die primäre VNIC bisher fest ist.

Ist das virtuelle, wolkige Netzwerk angelegt, muss man jetzt noch dafür sorgen, dass von dort auch Netzwerk-Verkehr nach außen – sei es ins Internet oder über VPN in sein Rechenzentrum – möglich ist. Den Wegweiser dafür bilden „Route Tables“, die an die Subnetze angeknüpft sind. Sie sind durchaus strapazierfähig und können auch mit mehreren Subnetzen eine Beziehung eingehen. Ihre Arbeit besteht darin, die Bits und Bytes der Subnetze entweder zum Internet-Gateway, also der Schnittstelle für die große, weite Welt, oder zum Dynamic-Routing-Gateway zu schicken, das wiederum mittels



Exzellente Baupläne für die Digitale Ökonomie!

Dafür steht PROMATIS als Geschäftsprozess-Spezialist mit mehr als 20 Jahren Erfahrung im Markt. Gepaart mit profundem Oracle Know-how schaffen wir für unsere Kunden die Digitale Transformation:

- Oracle SaaS für ERP, SCM, EPM, CX, HCM
- Oracle E-Business Suite und Hyperion
- Oracle Fusion Middleware (PaaS)
- Internet of Things und Industrie 4.0

Vertrauen Sie unserer Expertise als einer der erfahrensten Oracle Platinum Partner – ausgezeichnet als Top 25 Supply Chain Solution Provider 2017.

PROMATIS



PROMATIS Gruppe
Tel. +49 7243 2179-0
www.promatis.de
Ettlingen/Baden · Hamburg · Berlin
Wien (A) · Zürich (CH) · Denver (USA)

CustomerPremises-Equipment die Daten über VPN beispielsweise in sein Rechenzentrum überträgt.

DB System – Datenbank mit System

Die Frage nach einer Datenbank in der Oracle Cloud ist vergleichbar mit der Frage, ob ein Smartphone eigentlich noch telefonieren kann. Klar kann es das, aber es zwingt einen ja keiner dazu. Man kann sich auch einen Messenger installieren, wenn man mehr auf rein schriftliche Kommunikation steht. Es sei jedoch gesagt, dass einem dabei so einiges entgeht. Oracle bietet in seiner Cloud nicht einfach nur einen Service, der reine Datenbank-Funktionalität bereitstellt; vielmehr bekommt man genau das, was man sowieso die ganze Zeit haben wollte: einen Host, auf dem Oracle-Software installiert ist, so, wie man das von zu Hause gewohnt ist.

Auf dem Host besteht voller Root-Zugriff und man kann nach Belieben schalten und walten. Sollte man also mal nichts zu tun und in diesem Monat noch ein paar Universal Credits übrig haben, lässt sich innerhalb kürzester Zeit ein hochperformantes System aufbauen, das dann nach Lust und Laune zerschossen werden kann. Nur hinterher nicht vergessen, das Ganze wieder zu terminieren.

Das ist aber noch nicht alles. Oracle setzt für sein Steckenpferd OCI auf Engineered Systems, also ODA oder Exadata. Die dazugehörigen Database Bare Metal Services

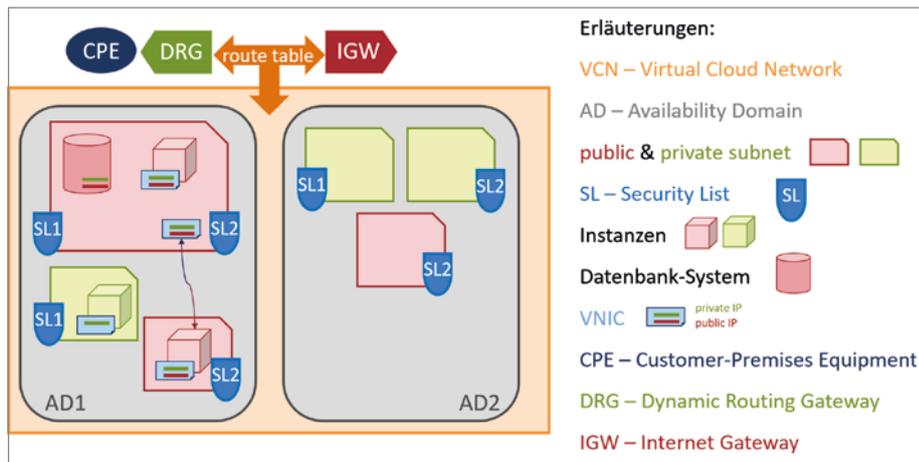


Abbildung 1: Netzwerk-Struktur in der OCI

bieten neben der bereits angedeuteten, hervorragenden Performance auch ein CLI, das einem das Leben als Administrator einfacher und schöner macht. In der Cloud nennt Oracle das gute Stück „dbcli“, die Ähnlichkeit zum „oakcli“ ist jedoch unverkennbar.

Man kann damit sogar etwas tun, was in der alten Oracle-Cloud so nicht möglich war: Mit dem einfachen Befehl „dbcli create-database --dbname mydbt --admin-password“ kann man sich eine Non-CDB-Datenbank erstellen. Wirft man einen genaueren Blick auf den Befehl, sollte zudem auch klar werden, dass gar kein Parameter bezüglich Container-Datenbank angegeben wurde. Das ist das Erstaunliche am „dbcli“ – der Default-Wert hinsichtlich Container-Datenbank lautet „Non-CDB“. Beim reinen Database Service, also ohne Bare Metal, macht man für eine Non-CDB Kopfstand und verliert dabei auch noch sein gesamtes Cloud-Tooling: Man muss

dort die CDB, die beim Erstellen des Service angelegt wurde, löschen, dann manuell selbst eine Non-CDB anlegen und diese dann wie On-Premises administrieren. Das geht, ist aber aufwendig.

Die I/O-Performance

Selbst musikalische Muffel werden bei diesem Thema feststellen, dass ihnen die Lobhymnen ganz von allein über die Lippen kommen. Für den Test kam ein (beziehungsweise das) Silly-Little-Oracle-Benchmark (SLOB) von Kevin Closson zum Einsatz, welches mit SQL-Abfragen I/O-Workload generiert und zum Schluss einen AWR-Report ausgibt. *Abbildung 2* zeigt einen Ausschnitt eines besonders schönen Exemplars.

Auf Cloud-Seite wurde ein Database Bare Metal Service mit Dense IO und 36 OCPUs auf Herz und Niere geprüft, wo-

IO Profile			
	Read+Write Per Second	Read per Second	Write Per Second
Total Requests:	1,537,180.2	1,537,095.4	84.8
Database Requests:	1,537,094.0	1,537,091.0	3.0
Optimized Requests:	0.0	0.0	0.0
Redo Requests:	80.6	0.1	80.4
Total (MB):	12,064.2	12,008.8	55.4
Database (MB):	12,008.8	12,008.7	0.1
Optimized Total (MB):	0.0	0.0	0.0
Redo (MB):	55.3	0.0	55.3
Database (blocks):	1,537,124.5	1,537,117.9	6.7
Via Buffer Cache (blocks):	1,537,121.3	1,537,117.4	3.9
Direct (blocks):	3.3	0.5	2.8

Abbildung 2: AWR-Report

bei man bei der Anzahl der OCPUs sicherlich noch wesentlich knausriger hätte sein können, schließlich sollte I/O und nicht Rechenkapazität getestet werden, aber wie sagt man so schön: „Wer hat, der kann“.

Für die Tests wesentlich entscheidender war dagegen, was sich hinter „Dense IO“ verbirgt; das steht derzeit für 28,8 TB NVMe SSD Raw Storage, der sich auf neun Platten verteilt. Die Daten können also gleichzeitig von neun SSD-Platten gelesen werden, die wiederum per NVMe angebunden sind. Diese technischen Voraussetzungen und ein offensichtlich sehr gutes Zusammenspiel der Hardware mit der darauf installierten Oracle-Software sorgen dafür, dass Oracle sich beim Thema „I/O-Performance“ ganz sicher nicht verstecken muss.

Das SLOB testet dabei übrigens wirklich nur die Hardware-Plattform, inwieweit diese für ein Oracle-Datenbank-Deployment mit hohen I/O-Performance-Anforderungen geeignet ist. Anwendungslogik wird dabei außen vor gelassen. Was auch bedeutet, dass die Datenbank-Software selbst nicht getestet wird, aber von der ist ja schließlich weithin bekannt, wie gut sie ist.

Security – sicher ist sicher

„Bei der Sicherheit hört der Spaß auf“, sagen alle. Weil Oracle das weiß und zudem

ein gewisses Interesse daran hat, dass man sich in der Cloud sicher fühlt, kommt die OCI mit einem eigenen Identity and Access Management Service (IAM) sowie einem Auditing Service daher. Wenn man also herausfinden will, wer die letzten Universal Credits aufgebraucht hat, die man eigentlich selbst für wichtige Tests nutzen wollte – irgendwo im Sammelsurium der ganzen Audit-Einträge wird man fündig. Nur die Web-Oberfläche bietet sich nicht unbedingt für die Suche an, da wäre man für eine Weile beschäftigt.

Für die Rechteverwaltung innerhalb des IAM ist die Web-Oberfläche schon eher geeignet. Damit ein Nutzer das Recht bekommen kann, auf eine Ressource zuzugreifen, reicht es allerdings nicht, nur den Nutzer selbst und die Ressourcen anzulegen. Der Nutzer muss einer Gruppe zugeordnet sein. Außerdem werden die Ressourcen in sogenannte „Compartments“ eingeteilt. Diese dienen zur inhaltlichen Gruppierung der Ressourcen, etwa nach Projekten oder Abteilungen in der Firma. Die Rechtevergabe erfolgt nun durch Policies, welche einer Gruppe bestimmten Ressourcen innerhalb eines Compartment Zugriffsrechte gewährt. *Abbildung 3* zeigt dazu ein Beispiel zur Veranschaulichung.

Bei genauerer Betrachtung dürfte auffallen, dass ein und derselbe Nutzer un-

terschiedliche Rollen innehaben und somit gleichzeitig mehreren Gruppen zugeordnet sein kann. Zudem kann es für ein und dieselbe Gruppe natürlich auch mehrere Regeln geben. Genau genommen kann sogar eine einzelne Policy mehrere Regeln enthalten, sodass Regeln, die inhaltlich zusammengehören, auch schön gruppiert werden können.

Mit Energie in die Cloud

Wenn ein Software-Hersteller wie Robotron die Oracle-Cloud auf Herz und Nieren prüft, liegt ja nichts näher, als dass da auch mal die hauseigene Software raus in die Wolken gelassen wird. So wurde also „robotron*~~e~~count“ in der Cloud installiert und sich hinterher daran erfreut, dass OCI beim Erstellen der Netzwerk-Struktur ausreichend Freiheit liefert, um eine mehrschichtige Anwendungsumgebung aufzubauen. Allerdings gab es während der Tests die Rechenzentren in Frankfurt noch nicht und so waren die Antwortzeiten der Anwendung zwischen Dresden und Phoenix, Arizona, doch zu langsam; sie hat sich jedoch mit einem zusätzlichen Terminalserver besänftigen lassen.

Auch wenn es nun mit den Rechenzentren in Frankfurt das Problem der Entfer-

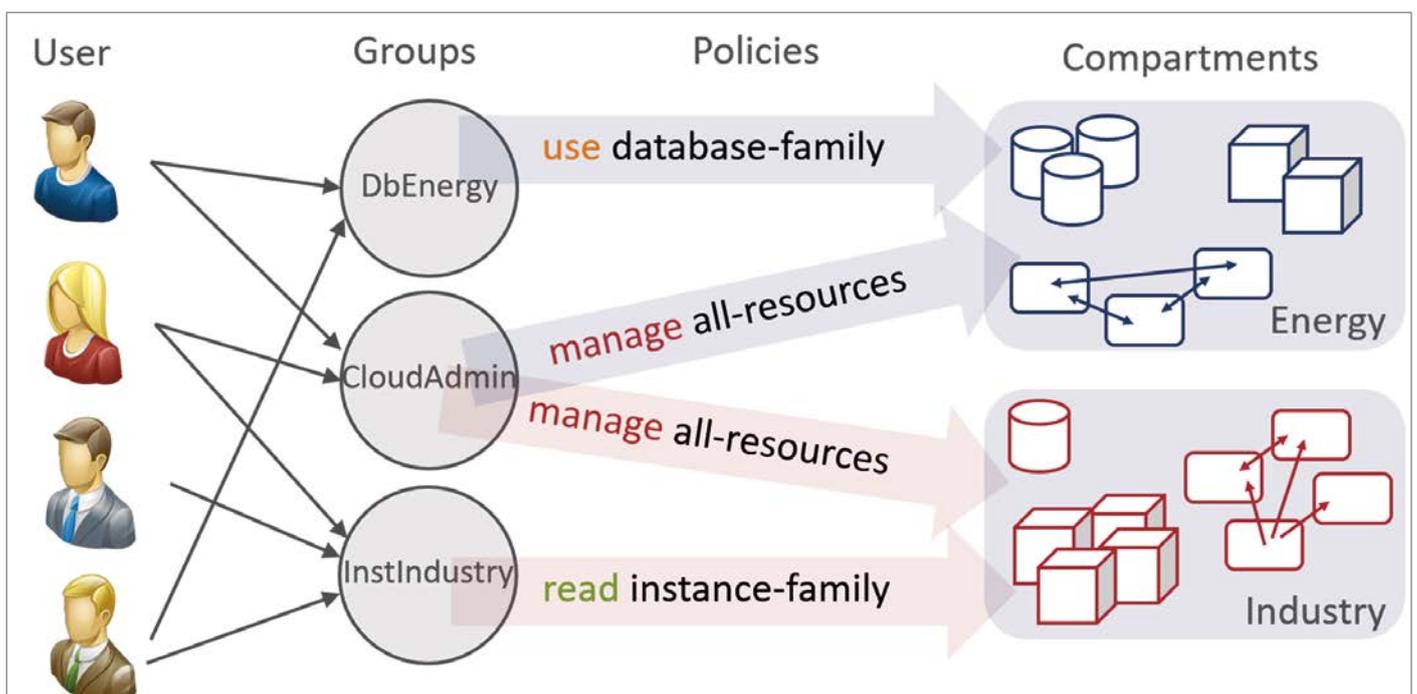


Abbildung 3: Rechtevergabe mit Identity and Access Management Service

nung so nicht mehr gibt, muss sich doch jeder Umzugswillige darüber Gedanken machen, wie die Anbindung von der Cloud ins eigene Rechenzentrum gestaltet sein soll. Je nach Verfügbarkeit bietet Oracle hierfür auch 1- oder 10-Gbps-FastConnect-Anbindungen, natürlich gegen Einwurf hinreichend vieler Scheine und Münzen. Letztere sind zudem auch für eine eventuelle MPLS-Leitung und deren Anbindung an das eigene Rechenzentrum erforderlich.

Fazit

Wenn man in die Cloud geht, dann nicht, weil es gerade Mode ist, weil das eben alle machen, oder gar, weil Oracle das

gern so hätte, sondern weil es einem Vorteile bringt. Diese Vorteile müssen zudem auch alle Nachteile aufwiegen, die die Abgabe von Kompetenzen und der Umzug von „haben“ zu „mieten“ so mit sich bringen kann. Mit den Bare Metal Services möchte Oracle wieder einen dieser Nachteile, nämlich den der störenden Nachbarn, erschlagen. Das geschieht auf durchaus eindrucksvolle Art und Weise, mit schicker Performance und einem eigenen Cloud-Netzwerk. Es ist einen Test wert, seinen Daten mal ein bisschen Ruhe zu gönnen. Übrigens ist Reinhard Mey zu Zeiten des Eigenheims zu „Irgendein Depp mäht irgendwo immer“ übergegangen – aber Wiesen gibt es in den Wolken keine.



Katharina Buechse
katharina.buechse@robotron.de



Florian Barth
florian.barth@robotron.de



Günther Stürner
dbms publishing

**Aus der Ferne betrachtet:
Software (und Daten) sind der Treibstoff der
Zukunft!**

Cloud-Computing ist da und wird nicht mehr verschwinden. Amazon, Google & Co. rüsten seit Jahren massiv auf, um uns alle mit Rechenleistung zu beglücken. Sie werden nicht müde zu betonen, dass die klassischen Firmen-Rechenzentren bald der Vergangenheit angehören werden und dass in der neuen Cloud-Welt alles besser, viel einfacher und vor allem viel billiger ist als das, was aus dem eigenen Maschinenraum kommt.

Wie das Spiel ausgeht, ist noch offen. Dass die Firmen-Rechenzentren eine gefährdete Spezies sein sollen, mag ich nicht glauben. Es wird sich allerdings in diesem Umfeld vieles verändern. Jedes Unternehmen muss seine eigene Balance zwischen Eigenbetrieb und Fremdbezug der IT-Services finden.

Oracle hat lange Zeit das Cloud-Thema ignoriert, um in der letzten Minute doch noch auf den Zug aufzuspringen. Ob der Einstieg in das Cloud-Geschäft klug war oder ob es smarter gewesen wäre, mit bestehenden Cloud-Anbietern clevere Partnerschaften einzugehen und sich voll auf das Software-

Geschäft zu konzentrieren, lässt sich heute noch nicht abschließend beurteilen.

Es ist, wie es ist: Oracle spielt mit und zieht alle Register, um den Abstand zu den heutigen „dicken Wolken“ möglichst schnell zu egalisieren. Die Zukunftsfrage von Oracle mit dem Erfolg der Cloud-Bemühungen zu verknüpfen, wie es von manchen Oracle-Granden gerne gemacht wird, halte ich jedoch schlicht für falsch. Es sei denn, es ist ein Umbau des Oracle-Hauses von einem führenden Software-Anbieter mit angegliedertem Hardware- und Cloud-Angebot in ein Cloud-Haus mit angegliedertem Software-Angebot geplant. Dann allerdings gute Nacht!

Oracle hat in den letzten fünf Jahren – ja, so lange ist das Thema dort schon akut – Großartiges geleistet. Respekt. Aber Oracle darf nicht ohne Not seine Perlen vor die Säue schmeißen. Software ist die Kernkompetenz, die das Unternehmen groß gemacht hat. Software und Daten sind der Treibstoff der Zukunft. Hier gibt es noch viel zu tun und noch viel zu gewinnen.

Oracle wird nachhaltigen Erfolg haben, wenn es gelingt, eine Balance zwischen On-Premises- und Cloud-Angeboten zu finden. Hier muss das Kundenbedürfnis im Vordergrund stehen und nicht die aktuell von Oracle favorisierte Lösung. Oracle wird dann Erfolg haben, wenn es die eigene Cloud-Umgebung gegenüber anderen Cloud-Umgebungen nicht bevorzugt. Das Ziel muss es doch sein, seine Software auf allen möglichen Hardware- und Cloud-Plattformen zu betreiben. Oracle wird dann Erfolg haben, wenn man sich auf seine Wurzeln besinnt und ein faires und nachvollziehbares Preismodell findet (gerne auch schwarz auf weiß zum Nachlesen). Und nicht zuletzt, wenn es gelingt, verloren gegangenes Vertrauen zurückzugewinnen, denn „Nur wo man sich vertraut, wird man dick!“



Cloud Service – Himmel und Hölle

Alexander Weber, Firma e:ndlich

Dieser Erfahrungsbericht stellt zunächst den Oracle Cloud Schema Service vor und zeigt dann die Aufgaben während der Migration. Es folgt ein Einblick in eine Phase der Nutzung, die zu einer massiven Störung des Geschäftsbetriebes führte und somit die möglichen Schwachstellen eines Cloud Service offenbart. Abschließend sind die Vor- und Nachteile dieses Cloud Service gegenübergestellt.

Seit mehr als fünf Jahren ist das Thema „Cloud“ in aller Munde und die Firma „endlich“ hat sich deshalb frühzeitig entschieden, hier selbst Know-how aufzubauen. Es gab zwei wichtige Gründe, sich mit den Cloud Services zu beschäftigen. Erstens, um den Aufwand der selbst betriebenen Infrastruktur zu minimieren. Zusätzlich sollten die Verfügbarkeit gesteigert und der Betrieb stabiler werden. Die zweite Motiva-

tion war ein großes Interesse daran, sehr früh eigene Erfahrungen in der Anwendung von Cloud Services zu sammeln.

Welchen Cloud Service?

Prinzipiell gibt es mehrere Ansätze, einen Cloud Service zu wählen. Eine Möglichkeit ist, alles so zu nehmen, wie es ist, entspre-

chende Maschinen in der Cloud zu abonnieren und alles „1:1“ zu übertragen. Dieser Ansatz ist jedoch nicht wirklich eine Verbesserung, da man eigentlich nur den Spielplatz wechselt, die Spielzeuge aber gleich bleiben. Dies kann sinnvoll sein, wenn man ohnehin mit seinem bisherigen Betreiber unzufrieden ist und wechseln will.

Eine zweite Option ist es, die Hardware zu virtualisieren und einige vorhandene

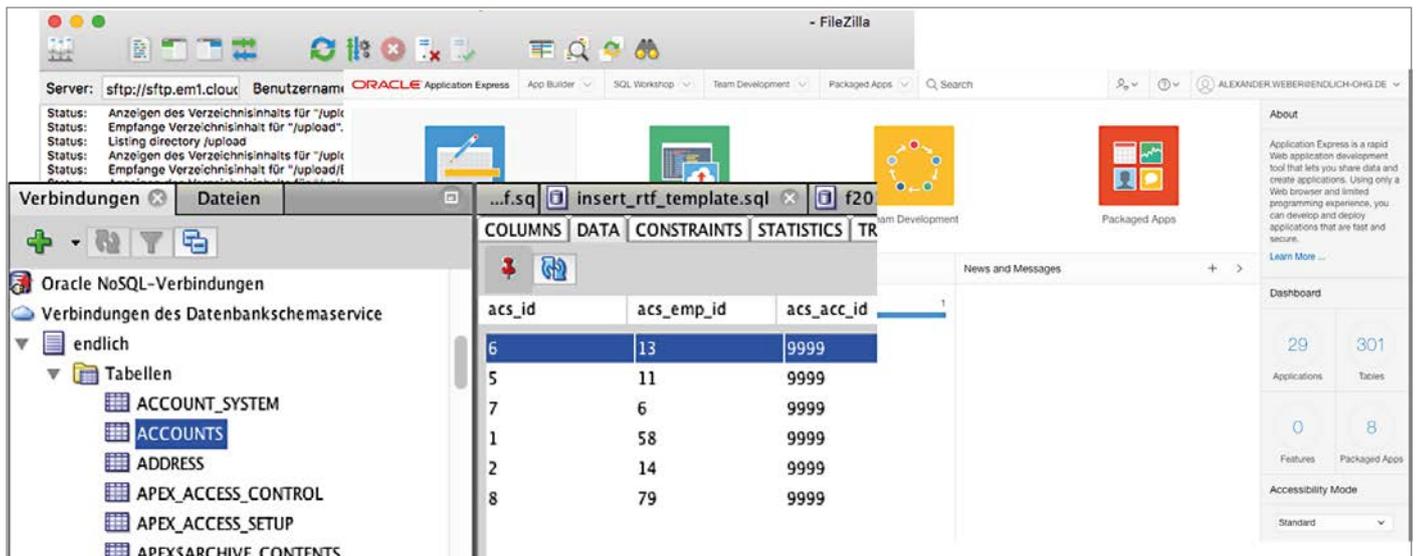


Abbildung 1: Ausschnitte aus den Cloud-Service-Zugängen mit Apex, FileZilla und SQL Developer

Dienste in die Cloud zu verlagern (etwa eine Datenbank oder ein Active Directory), damit geht man schon in Richtung Konsolidierung und Standardisierung der eigenen Anwendung.

Die dritte Möglichkeit besteht darin, einen Service zu wählen, der es ermöglicht, dass die eigene Anwendung ausschließlich auf Basis von Cloud Services läuft und man sich wirklich nur noch um den Betrieb und die Weiterentwicklung der Anwendung kümmern muss.

Als letzte Alternative bliebe noch die vollständige Ablösung der eigenen Anwendung durch eine Cloud-Lösung, die annähernd dasselbe macht wie die selbstentwickelte Lösung und bei der man sich danach nicht mehr um die Weiterentwicklung kümmern muss.

Das Unternehmen des Autors hatte zum Zeitpunkt der anstehenden Entscheidung für einen Wechsel in die Cloud viele Anwendungen im Haus, die auf Oracle-Basis und im Frontend auf Apex liefen. Der maximal unterstützte Technologie-Stack wurde von Oracle selbst angeboten, denn hier erhielt man einen Service, der sich sogar um den Betrieb und die Pflege von Apex kümmert. Die Wahl fiel deshalb auf den Oracle Cloud Schema Service.

Der Oracle Cloud Schema Service

Dieser Service läuft unter der Kategorie „Platform as a Service“ (PaaS). Man bezeichnet so eine Dienstleistung, die in der

Cloud eine Computer-Plattform für Entwickler von Webanwendungen zur Verfügung stellt. Dabei handelt es sich um

schnell einsetzbare Laufzeit-Umgebungen.

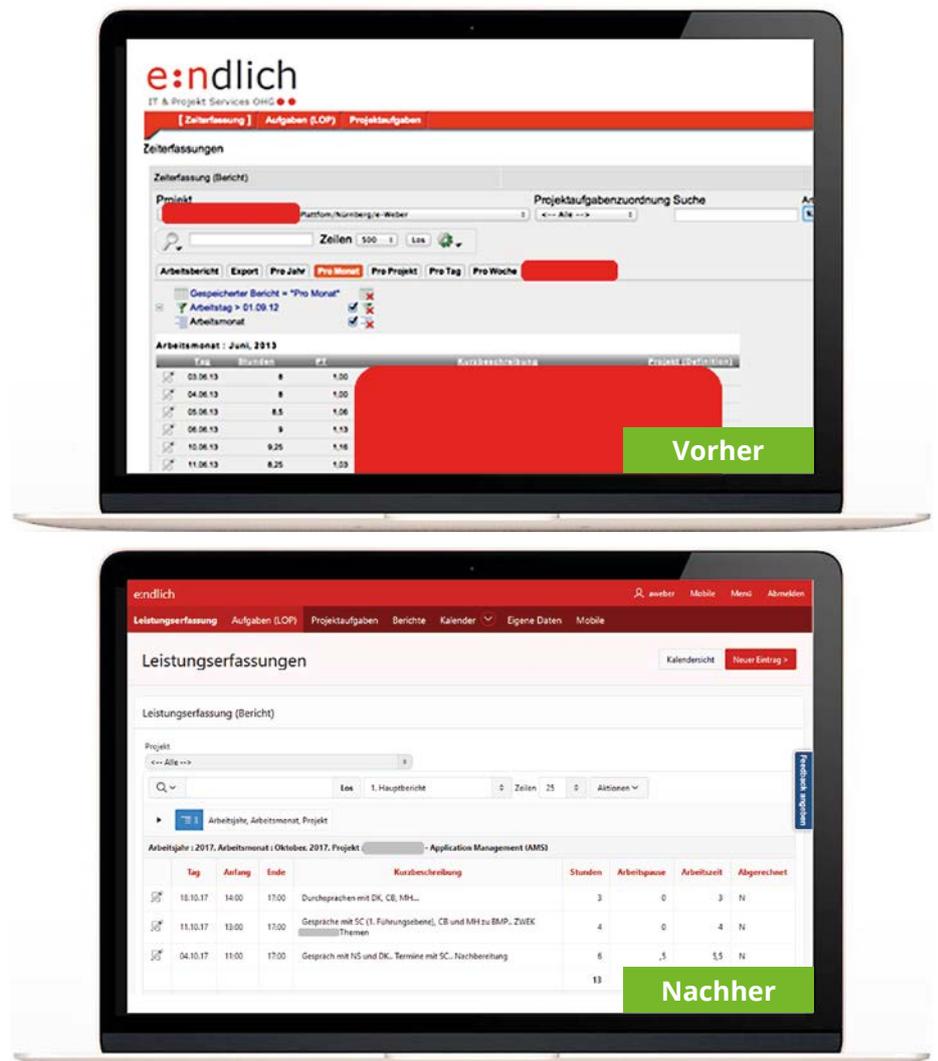


Abbildung 2: Teilmodul „Leistungserfassung“ vor und nach der Migration in die Cloud

gen, aber auch um Entwicklungs-Umgebungen, die mit geringem administrativen Aufwand und ohne Anschaffung der darunterliegenden Hard- und Software genutzt werden können (siehe Abbildung 1).

Der Database Schema Service besteht derzeit aus:

- Apex 5 (5.1.3), zum Zeitpunkt der Migration war es Apex 4 (4.2.6)
- Ein Datenbank-Schema auf einer 11g R2 Datenbank (11.2.4)
- SFTP-Zugang für Export/Import
- Zugang zu den Daten über REST-Services (zum Beispiel auch über SQL Developer)

Entwickler nutzen PaaS, um neuartige und bewusst für die Cloud gestaltete Applikationen zu entwickeln. Dabei stehen einem prinzipiell alle Micro-, REST-, Web- und Cloud-Services zur Verfügung, um die Anwendung zu erweitern. In sogenannten „Marketplaces“ kann man diese auch gut vermarkten.

Die Umsetzung/Migration

Im ersten Schritt ging es darum, eine geeignete Applikation für die Migration in die Cloud zu finden. Das Team des Autors hat folgende Kriterien zugrunde gelegt:

- Möglichkeit der Konsolidierung der Anwendung auf die Zwänge der zukünftigen Ziel-Umgebung ist mit hoher Wahrscheinlichkeit vorhanden
- Bereitstellung eines Entwicklers aus bestehenden Projekten für diese Anwendung ist möglich. Es sollte jemand sein, der die Anwendung (mit-) entwickelt hat und sie sehr gut kennt
- Möglichst keine Verzahnung mit anderer Software und wenige Schnittstellen

Die Wahl fiel auf die Apex-Anwendung zur Unternehmenssteuerung, also quasi ein ERP-System (siehe Abbildung 2).

Folgende Aufgaben standen auf der To-do-Liste:

- Konsolidierung und Re-Engineering der Anwendung
- Migration Apex 3 nach Apex 4 (damals)
- Zusammenführung des Daten- und des Applikations-Schemas
- Obfuscation-Toolkit durch „DBMS_CRYPTO“ ersetzen



Abbildung 3: Datenschutz

- Deployment-Vorgehen anpassen
- LDAP-Bezüge eliminieren
- Java-Reports-Engine durch eine PL/SQL-Eigenentwicklung ersetzen
- Neue Application-IDs

Datenschutz

Der Datenschutz spielt im Cloud-Umfeld eine große Rolle. Kunden in Deutschland und Europa legen viel Wert darauf, dass mit ihren Daten sorgsam umgegangen wird, was gleichzeitig ein gewisser Innovations-Hemmschuh ist. Das Unternehmen des Autors hat sich deshalb mit den Gesetzen und Bestimmungen auseinandergesetzt und frühzeitig einen externen Datenschutzbeauftragten ins Boot geholt, um Fragestellungen bezüglich IT-Sicherheit und Datenschutz zu begleiten (siehe Abbildung 3).

Mit Datenschutz ist in der Regel auch gleichzeitig IT-Sicherheit gemeint, was jedoch nicht dasselbe ist. Es gibt im Zusammenspiel von Datenschutz und IT-Sicherheit jedoch eine Überschneidung durch die technisch-organisatorischen Maßnahmen (TOM). Bei einer Migration von Services in die Cloud sind sowohl die eigenen TOMs als auch die des Cloud-Anbieters unter die Lupe zu nehmen. Dies betrifft insbesondere die Auftragsdatenverarbeitung.

Beim Datenschutzgesetz (BDSG beziehungsweise EU DSGVO) geht es dar-

um, die Daten von natürlichen Personen zu schützen. Dazu zählen Attribute wie Name, Vorname, Geburtstag, Geburtsort etc., aber auch die näheren personenbezogenen Daten wie Straße und Ort oder BIC und IBAN. Es geht darüber hinaus auch immer darum, ob man mithilfe der Daten in einem Datenpool einen Bezug zu einer Person herstellen kann oder nicht.

Eine Datenmenge gilt rechtlich als datentechnisch verschmutzt, wenn es einen Datensatz gibt, mit dessen Hilfe eine Person eindeutig identifizierbar ist. Als Beispiel: In einem Datenpool werden lediglich Ortsteil und Alter gespeichert und ausgewertet. Dies klingt erst mal unkritisch, allerdings könnte es sein, dass im fiktiven Ortsteil „Einsamer Hof in kleiner Stadt“ nur ein einziger Mensch mit 85 Jahren wohnt, weshalb man direkt Rückschlüsse auf die Person ziehen könnte – was nicht erlaubt ist.

Im IT-Sicherheitsgesetz steht der Schutz kritischer Infrastrukturen im Vordergrund, die beispielsweise für die Wasser- und Energieversorgung oder die Aufrechterhaltung des Gesundheitswesens verantwortlich sind. Wenn man Systeme für solche Infrastrukturen zur Verfügung stellt, muss man auch dafür Sorge tragen, dass die IT-Sicherheit gewährleistet ist. Hier liegt ein klarer Vorteil in der Nutzung von Cloud-Diensten, da diese in vielen Fällen den Anforderungen der IT-Sicherheit von vornherein genügen.

Gemäß § 9 Bundesdatenschutzgesetz (BDSG) sind alle Stellen, die personenbezogene Daten verarbeiten, erheben oder benutzen, verpflichtet, technische und/oder organisatorische Maßnahmen zu treffen, um zu gewährleisten, dass die Sicherheits- und Schutzanforderungen des BDSG erfüllt sind, etwa die Zugangs- oder Auftragskontrolle. Sie müssen die Einhaltung der TOMs selbst laufend kontrollieren, aber auch beim Cloud-Anbieter nachfragen.

Was ist mit den US-Anbietern?

Jene US-Anbieter, die einen Großteil der Cloud-Service-Angebote stellen oder die Hersteller der zugrunde liegenden Software sind, unterliegen per se nicht den Datenschutz- und IT-Sicherheitsbestimmungen der EU. Allerdings gab es dafür in der Vergangenheit das Safe-Harbour-Abkommen und nun das „EU-U.S. Privacy Shield“, eine Art Freibrief, der es erlaubt, die US-Dienste zu nutzen.

Grundsätzlich ist es hilfreich, wenn man einen Cloud-Dienst-Anbieter mit europäischer Niederlassung wählt, da dieser somit auch auf europäischem Boden betrieben wird. Bei dieser Konstellation gelten die europäischen Gesetze und Richtlinien auch für den US-Anbieter.

Oracle hat weltweit lokale Data-Center in Betrieb. Das Unternehmen des Autors hat sich damals für das Datacenter „EMEA1“ auf schottischem Boden in Lintlithgow entschieden. Viele Informationen über das Data-Center werden jedoch nicht bereitgestellt. Auch online etwas darüber herauszufinden, ist nicht leicht, was jedoch potenzielle Angriffe wegen fehlender Informationen erschwert. Man bewertet das deshalb nicht als negativ.

Datenschutz-Fahrplan für den Weg in die Cloud

Folgende Punkte sollten auf jeden Fall implementiert sein, bevor man eigene Services in die Cloud migriert:

- Es sind externer Datenschutzgeber und interner Datenschutzbeauftragter zu benennen
- Cloud-Anbieter-TOMs sind durch externe Datenschutzgeber abgefragt und existieren

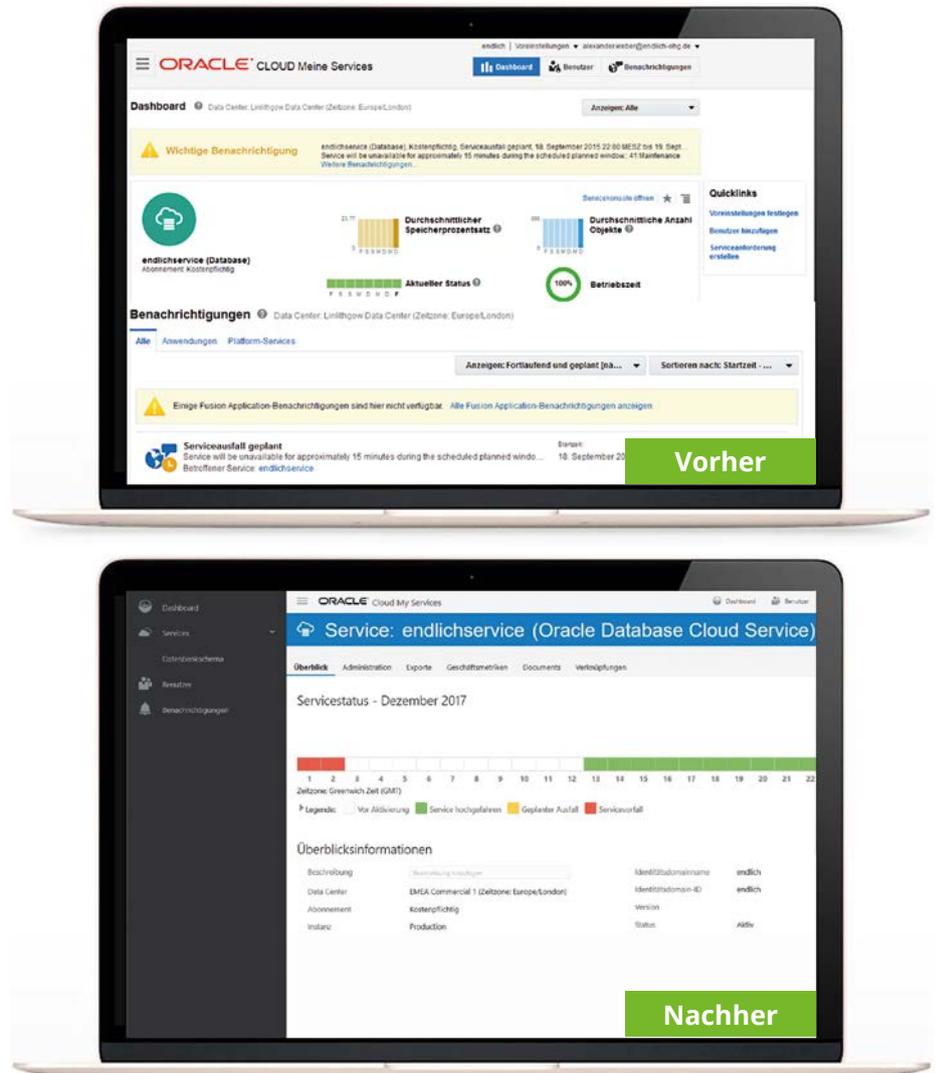


Abbildung 4: Service-Ansicht vor zwei Jahren und heute

- Cloud-Dienst erfüllt BDSG beziehungsweise die EU-DSGVO
- Cloud-Anbieter ist nach ISO 27001 zertifiziert
- Data-Center befindet sich innerhalb der EU oder/und es existiert ein Angemessenheitsbeschluss wie das „EU-U.S. Privacy Shield“

Wie sich der Service überwachen lässt

Sobald die Anwendung in der Cloud ist, hat man auch ein Interesse daran zu wissen, wie der Dienst funktioniert. Es gibt dazu ein „Cloud Service Dashboard“, in dem alle Cloud-Dienste aufgelistet sind. Dort lassen sich für jeden Service die wichtigsten Parameter wie Speicherplatz-Nutzung, Verfügbarkeits-Status in der aktuellen Woche und die

generelle Betriebszeit ablesen. Es sind zudem einige grundsätzliche Verwaltungsaktionen (wie die Benutzerverwaltung und die Initiierung von Datenexporten) möglich. Ein Datenexport wird manuell beauftragt und kann kurz darauf vom SFTP-Server abgeholt werden. Inzwischen bekommt man auch proaktive Informationen über geplante Ausfallzeiten in der Service-Ansicht und separat als E-Mail (siehe Abbildung 4). Das Look & Feel des Dashboards und auch die Service-Ansicht haben sich in den letzten Jahren mehrmals verändert, was es immer etwas anstrengend macht, sich zurechtzufinden.

Im Himmel

Nachdem das Unternehmen migriert hatte, ist man zu der Überzeugung gelangt,

dass der Aufwand von einem halben Jahr, um die Anwendung zu migrieren, auf jeden Fall gerechtfertigt war. Verglichen mit der Energie, die man vorher täglich in die Betreuung der Applikation investieren musste, ist der jetzige Zustand der Himmel des Anwendungsbetriebs. Vor allem folgende Vorteile standen im Vordergrund:

- Keine Betriebsaufgaben mehr
- Ab der Migration deutlich weniger ungeplante Ausfallzeiten
- Apex-Backend-Migrationen werden für das Unternehmen durchgeführt
- CPUs und/oder PSUs werden für das Unternehmen eingespielt
- O/S-Patches werden für das Unternehmen eingespielt
- Angriffe von außen werden abgewehrt
- Das Unternehmen kümmert sich nur noch um die Entwicklung

- Statt einer Datenbank-Lizenz und jährlichen Support-Kosten nun ein monatlich kündbares Abonnement
- Von Anfang an skalierbar

Hölle – oder was dann geschah

Bei der Cloud-Anwendung handelt es sich um die firmeninterne ERP-Anwendung zur Verwaltung der Angebote, Aufträge, Rechnungen, Assets, Mitarbeiterdaten und Leistungserfassung. Sie ist Dreh- und Angelpunkt des Geschäfts. Die Mitarbeiter nutzen die Anwendung beim Kunden, um ihre verrechenbaren Leistungen zu erfassen, Mitarbeiter im Backoffice erfassen mithilfe der Anwendungen neue Aufträge, erstellen Rechnungen, verwalten die Assets und es werden Dokumente zu

Mitarbeitergesprächen verschlüsselt hinterlegt.

Im Februar 2017 stand jedoch die Anwendung von der einen auf die andere Minute nicht mehr zur Verfügung (siehe Abbildung 5). Nach einer kurzen Wartezeit wurde ein PRIO-1-Service-Request bei Oracle eröffnet und man hoffte auf eine zügige Bearbeitung.

Zur Erinnerung: Ein PRIO-1-Service-Request ist die höchstmögliche Priorität, mit der man einen Service-Request absetzen kann. Bei der Eröffnung eines solchen Tickets muss nicht nur der Fehler beschreiben, sondern es müssen auch die Ansprechpartner, die Manager und jeweils deren Kontaktdaten benannt sein. Diese sollen dann für Oracle rund um die Uhr zur Verfügung stehen beziehungsweise erreichbar sein. Somit wurde ein klares Signal gesetzt, dass es sich um ein wichtiges Anliegen handelt und zeitnahe Unterstützung erforderlich ist. Die erste Reaktion seitens Oracle war deshalb sehr enttäuschend: „Hi Alexander, we don't support Apex Cloud Service issues. Please reach out to the right team. Regards, ... Oracle Cloud Services ...“

Man wurde jedoch nicht informiert, welches das richtige Team ist, und der Ball lag wieder beim Unternehmen; man wurde quasi abgewimmelt, was sich in dem Augenblick nicht gut anfühlte. Das Unternehmen führte zeitgleich Telefonate mit Oracle-Support, Oracle Potsdam und Oracle München. Es stellte sich heraus, dass es für den Support gar nicht so leicht zu ermitteln ist, um welchen Cloud-Dienst es eigentlich geht. Die Abo-ID und die CSI-Nummer reichten dazu jedenfalls nicht aus. Man wurde gebeten, Schema- und Tablespace-Name zu benennen – das ist jedoch nicht so einfach, wenn der Service nicht mehr zur Verfügung steht.

Mit etwas Glück und durch Recherche in Skripten konnte man die gewünschten Informationen zusammentragen. Zwischenzeitlich wurde das zuständige Support-Team identifiziert („CLOUDOPS“). Es hat für das Unternehmen jedoch ziemlich intransparent im Hintergrund gearbeitet. Der Kontakt im Support konnte Fragen nicht direkt beantworten, sondern musste diese erst an das CLOUDOPS-Team weiterleiten.

Das Unternehmen wurde parallel durch den vertrieblischen Ansprechpartner gebeten, einen neuen Database Schema Service

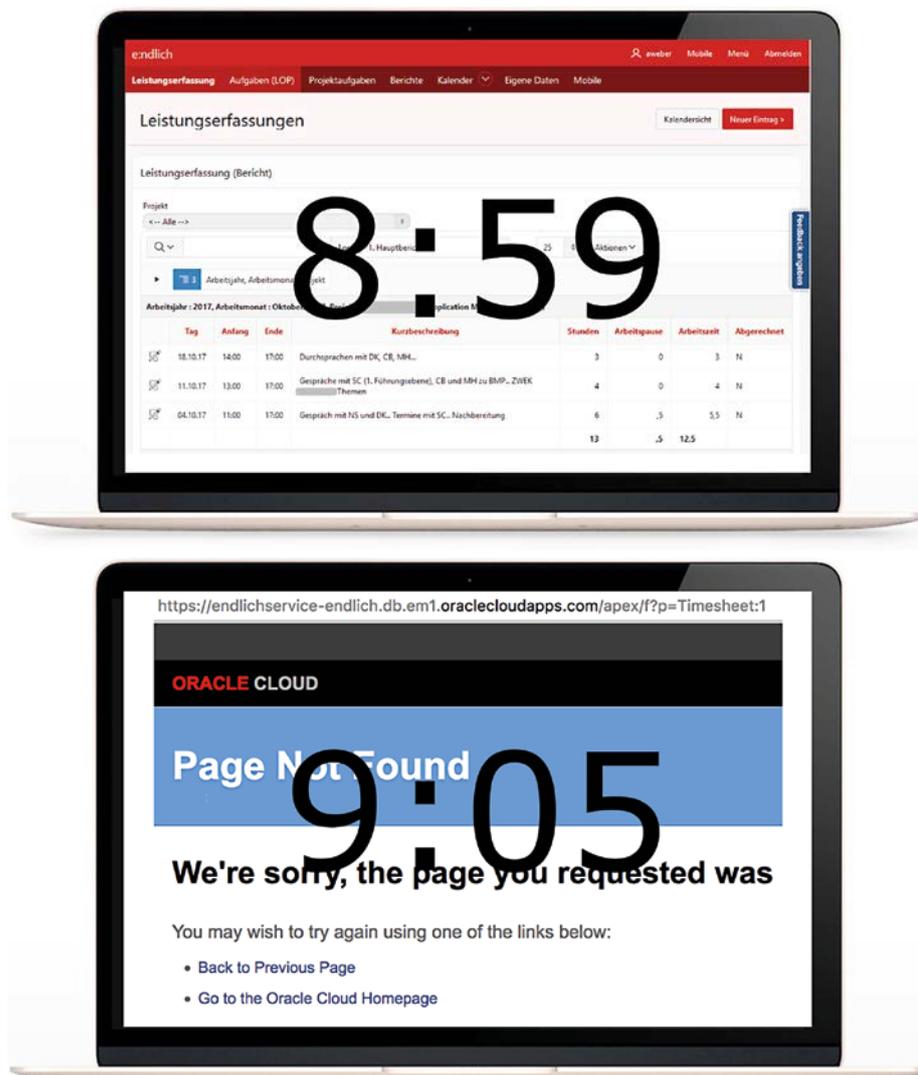


Abbildung 5: Anwendung in der Cloud nicht mehr verfügbar

zu bestellen. Wozu dies nötig war, konnte man zu dem Zeitpunkt nicht begreifen, hat es aber aufgrund der Notlage getan. Dies ging letztlich relativ schnell, allerdings musste man als kleinere Firma auch nicht über einen Einkauf gehen und zeichnungs-berechtigte Personen finden, sondern konnte alles durch die Geschäftsführung erledigen lassen. Zwei Tage nach Eröffnung des Service Request stand der neue, völlig nackte Cloud Schema Service zur Verfügung. Nackt deshalb, weil dieser Service weder unsere Daten noch unsere Anwendung enthielt.

Nun traf ein weiterer Rückschlag seitens des Supports ein. Nachdem man bestätigt hatte, dass der neue Cloud-Dienst grundsätzlich zur Verfügung steht, kam folgende Antwort: „Hi, Thanks for your confirmation. I will be closing this SR since the service is active now. For any issues, kindly log a separate SR. Kind Regards,...“

Man fühlt sich in diesem Augenblick sehr einsam und bereut zutiefst, jemals die Entscheidung pro Cloud getroffen zu haben. Dieses Beispiel unterstreicht den Beitrag von Herrn Dr. Neugebauer „Oracle Support – Quo vadis“ Anfang des Jahres im Red Stack Magazin, in dem er die sinkende Zufriedenheit mit My Oracle Support anspricht.

Das Unternehmen hat im Anschluss direkt mit dem Produktmanagement Kontakt aufgenommen und ab diesem Zeitpunkt den Eindruck, dass tatsächlich etwas passiert. Nach und nach standen immer mehr Daten aus der gewohnten Umgebung zur Verfügung. Zunächst erstmal nur der Zugriff auf die Laufzeit-Umgebung, aber es waren immerhin wieder die eigenen Daten zu sehen und die Anwendung hat grundsätzlich funktioniert. Es wurden jedoch unter anderem noch Umlaute in der Applikation falsch dargestellt und Public beziehungsweise Private Interactive Reports waren nicht verfügbar.

Auch der Zugriff auf die Apex-Entwicklungsumgebung war nicht möglich. Aber: Das Schlimmste war zunächst überstanden und das Unternehmen hat sich bei den hilfreichen Unbekannten über das Produktmanagement bedankt. Es kam folgende Antwort: „,... Kollegen in den USA haben in einer konzentrierten Aktion mit diversen anderen Cloud-Teams das Ganze wiederherstellen/ mit alten Namen aufsetzen können ...“

Fünf Tage später konnte man also wieder grundsätzlich mit der Anwendung arbeiten und der Geschäftsbetrieb war

nicht mehr eingeschränkt. In der Folgezeit wurden auch alle weiteren Schwierigkeiten gelöst, es gibt inzwischen wieder ein völlig sauberes System und man kann wie zuvor damit arbeiten.

Eines zeigt diese Antwort des Cloud-Service-Produktmanagements jedoch auch: Die Komplexität, einen versehentlich gelöschten Service wiederherzustellen – und bei diesem Beispiel handelt es sich nicht um eine komplexe Umgebung – ist offenbar sehr hoch und bedarf der Zusammenarbeit mehrerer Spezialisten.

Eine wirklich offizielle Ursache für diesen Ausfall hat das Unternehmen nie erfahren, konnte aber den Gesprächen Folgendes entnehmen: Der Cloudvertrag basierte auf einer Kreditkarte, deren Ablaufdatum im August des Vorjahres erreicht war. Es gab jedoch keine Kommunikation seitens Oracle darüber, dass dies der Fall ist. Der Service lief trotzdem von August des Vorjahres bis Februar weiter und wurde dann irgendwann auf „zu archivieren“ gestellt; die technische Ausführung dazu fand im Februar statt.

Fazit

Letztendlich lässt sich sagen, dass die Ursache für den Ausfall kein technischer Fehler des Cloud Service, ungewollter Datenverlust oder Hackerangriff war. Es handelte sich um ein Problem in den Verwaltungsprozessen, die den Cloud Services zugrunde liegen. Bei komplexen, multinationalen Strukturen in großen Konzernen, bei denen die Prozesse nicht mit den technischen Innovationen Schritt halten können, kann dies zu Problemen führen. Es ist deshalb notwendig, einen funktionierenden Kommunikationskanal zwischen den Nutzern von Cloud-Diensten und dem Anbieter herzustellen, der während des gesamten Application Lifecycle einer Cloud-basierten Anwendung funktioniert. Dies könnte der Support sein, ist es aber derzeit nicht.

Trotzdem überwiegen für uns insgesamt die Vorteile, die die Cloud Services mit sich bringen. Das skizzierte Ausfallszenario hätte auch unabhängig davon, dass es einen Cloud Service betraf, auftreten können. Der Betreiber einer Datenbank in einem On-Premises-Data-Center hätte auch versehentlich den Abbau einer produktiven Datenbank veranlassen können.

Fehler passieren und werden auch zukünftig nicht zu vermeiden sein. Jedenfalls

war der Cloud Service trotzdem so robust konzipiert, dass nach einiger Zeit wieder alles ohne Daten- und Funktionalitätsverlust hergestellt werden konnte. Abschließend die Nach- und die Vorteile im Überblick:



Nachteile

- Einige gute Ideen ließen sich wegen der eingeschränkten Funktionalität nicht umsetzen
- Gefühlte unsichere Handlungssituation in der „Lieferantenkette“ aufgrund von fragiler Datenschutzlage (EU-U.S. Privacy Shield)
- Wenn etwas schiefgeht, hat man keinen Ansprechpartner im Haus und ist der Effektivität des Supports beim Cloud-Anbieter ausgeliefert



Vorteile

- 100-prozentige Entlastung im Betrieb
- Viel weniger ungeplante Ausfallzeiten
- Gewöhnung an die abgegebene Kontrolle und Genuss der Vorteile
- keine Lizenz- und Supportgebühren, sondern einfach ein monatlich kündbares Abonnement
- Skalierbarkeit lässt sich einfach online hinzukaufen
- Mehrere Oracle Cloud Services können von einem Ort aus verwaltet werden
- Die Erfüllung der immer höher und komplexer werdenden Datenschutzaufgaben übernimmt der Cloud-Anbieter
- Know-how-Gewinn und Erhalt der eigenen Wettbewerbsfähigkeit
- Einschränkung in der Funktionalität des Schema Service hält die Anwendung technisch schlank und man konzentriert sich auf die fachliche Logik



Alexander Weber
alexander.weber@endlich.it



„Lift & Shift“ oder wie kommt meine (Pluggable) Datenbank in die Oracle-Cloud

Kai Uwe Fischer, Logicalis GmbH

„Push a Button to move your Database to the Oracle-Cloud“ ist einer der Slogans, mit denen Oracle für die Public Cloud wirbt. Ist es wirklich nur ein Knopfdruck? Dieser Artikel – basierend auf eigenen Test- und PoC-Erfahrungen – soll dies klären.

Bevor wir unsere Reise in die Oracle-Datenbank-Cloud beginnen, weist der Autor auf drei wichtige Dinge hin, die es auf dem Weg zu beachten gibt. Nach Erstellung einer Database-as-a-Service-Instanz über das Oracle-Database-Cloud-Service-Dashboard sind per Default alle Ports in der Firewall, außer dem SSH-Port 22, geschlossen (siehe Ab-

bildung 1). Damit die Reise nicht beendet ist, bevor sie überhaupt richtig begonnen hat, muss der SSH-Port in der Firewall des Unternehmens geöffnet sein. Ist dies der Fall, kann beispielsweise mit Putty oder dem SQL Developer mithilfe des erstellten Private-/Public-SSH-Schlüsselpaares auf den Datenbank-Server zugegriffen werden.

Folgende Software-Releases stehen als Oracle Database Cloud Service (DBCS) zur Verfügung:

- 11g Release 2
- 12c Release 1
- 12c Release 2
- 18c Release 1

Status	Rule Name	Source	Destination	Ports	Protocol	Description	Rule Type	Actions
🔵	ora_p2_ssh	PUBLIC-INTERNET	DB	22	TCP		DEFAULT	☰
🔵	ora_p2_http	PUBLIC-INTERNET	DB	80	TCP		DEFAULT	☰
🔵	ora_p2_https	PUBLIC-INTERNET	DB	443	TCP		DEFAULT	☰
🔵	ora_p2_httpadmin	PUBLIC-INTERNET	DB	4848	TCP		DEFAULT	☰
🔵	ora_p2_dbconsole	PUBLIC-INTERNET	DB	1158	TCP		DEFAULT	☰
🔵	ora_p2_dbexpress	PUBLIC-INTERNET	DB	5500	TCP		DEFAULT	☰
🔵	ora_p2_dblistener	PUBLIC-INTERNET	DB	1521	TCP		DEFAULT	☰
🔵	sys_infra2db_ssh	PAAS-INFRA	DB	22	TCP	DO NOT MODIFY: Permit P...	SYSTEM	☰
🔵	ora_trusted_hosts_dbli...	127.0.0.1/32	DB	1521	TCP	DO NOT MODIFY: A securle ...	SYSTEM	☰

Abbildung 1: Die Access-Rules

Zu beachten ist, dass das Software-Release 12c, im Gegensatz zur On-Premise-Welt, nur in der Single-/Multitenant-Architektur verfügbar ist. Ein weiterer Unterschied zwischen der Oracle-On-Premise-Welt und dem Database Cloud Service (DBCS) in der Oracle-Public-Cloud ist, dass alle User-Daten per Default in der Oracle-Public-Cloud per Transparent Data Encryption (TDE) verschlüsselt sind, dies gilt auch für die Standard Edition. Der Anwender muss sich um die TDE-Konfiguration im DBCS-Umfeld nicht kümmern, dies geschieht automatisch beim Anlegen des Datenbank-Cloud-Service. Listing 1 zeigt den entsprechenden Eintrag in der Datei „sqlnet.ora“.

Jede Pluggable Datenbank (PDB) benötigt dabei einen eigenen Master-Key. Mehr dazu später. Aus den genannten Punkten ist zu erkennen, dass neue Aufgaben auf einen Datenbank-Administrator zukommen werden, denn mit den Themen „Verschlüsselung“ (TDE) und „Tenant-Architektur“ wird er sich in Zukunft beschäftigen müssen.

Wege in die Oracle-Database-Cloud

Es gibt eine Vielzahl von Möglichkeiten, wie eine On-Premise-Oracle-Datenbank in die

Cloud migriert werden kann. Die eingesetzten Tools und Technologien hängen dabei von dem eingesetzten Release und der Edition ab. Im Using-Oracle-Database-Cloud-Service-Handbuch (siehe „https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/mig-migrating-premises-oracle-db-cloud.html“) sind im Kapitel 9 die einzelnen Migrationsmethoden beschrieben:

- Data Pump Conventional Export/Import
- Data Pump Full Transportable
- Data Pump Transportable Tablespace
- Remote Cloning a PDB
- Remote Cloning Non-CDB
- RMAN Cross-Platform Transportable PDB
- RMAN Cross-Platform Transportable Tablespace Backup Sets
- RMAN Transportable Tablespace with Data Pump
- RMAN CONVERT Transportable Tablespace with Data Pump
- SQL Developer and INSERT Statements to Migrate Selected Objects

- SQL Developer and SQL*Loader to Migrate Selected Objects
- Unplugging/Plugging a PDB
- Unplugging/Plugging Non-CDB

Dieser Artikel geht nicht auf die klassischen Migrationsmethoden mittels Data Pump und RMAN ein, sondern stellt folgende neue Migrationswege und deren Besonderheiten vor:

- Unplug/Plug-PDB (Kommandozeile)
- PDB-Clone (Oracle Enterprise Manager, SQL Developer)
- PDB-Hot-Clone (Kommandozeile)
- NONCDB to PDB (Kommandozeile)
- PDB-Klon von Oracle 12.1 nach Oracle 12.2

PDB-Unplug/Plug mittels Kommandozeile

In der Datenbank-Version 12 Release 1 konnte eine Pluggable Datenbank (PDB)

```
ENCRYPTION_WALLET_LOCATION = (SOURCE=(METHOD=FILE) (METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/DBHP2/tde_wallet))
```

Listing 1

```
SQL> alter pluggable database KUF3 close;
SQL> alter pluggable database KUF3 unplug into '/u01/stage/kuf3.pdb';
```

Listing 2

```
SET SERVEROUTPUT ON
DECLARE
  l_result BOOLEAN;
BEGIN
  l_result := DBMS_PDB.check_plug_compatibility(
    pdb_descr_file => '/u02/stage/kuf3.pdb',
    pdb_name       => 'kuf3');
  IF l_result THEN
    DBMS_OUTPUT.PUT_LINE('compatible');
  ELSE
    DBMS_OUTPUT.PUT_LINE('incompatible');
  END IF;
END;
/
SQL> SET SERVEROUTPUT ON
DECLARE
  l_result BOOLEAN;
BEGIN
  l_result := DBMS_PDB.check_plug_compatibility(
    pdb_descr_file => '/u02/stage/kuf3.pdb',
    pdb_name       => 'kuf3');
  IF l_result THEN
    DBMS_OUTPUT.PUT_LINE('compatible');
  ELSE
    DBMS_OUTPUT.PUT_LINE('incompatible');
  END IF;
END;
/
SQL> 2 3 4 5 6 7 8 9 10 11 12 13 compatible

PL/SQL procedure successfully completed.
```

Listing 3

```
SQL> create pluggable database KUF3 using '/u02/stage/kuf3.pdb';
SQL> alter pluggable database KUF3 open;
SQL> show pdbs;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ WRITE	NO
4	MIG	READ WRITE	NO
5	MIG2	READ WRITE	NO
6	KUF3	READ WRITE	NO

Listing 4

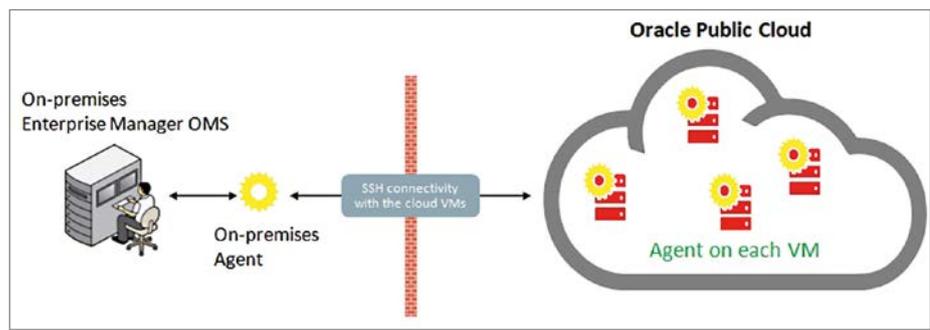


Abbildung 2: Der Hybrid-Cloud-Agent

in eine „*.xml“-Datei entladen werden. In dieser war nur die Beschreibung der PDB enthalten. Um diese PDB nun auf eine andere Datenbank umzuziehen, war es notwendig, daneben alle relevanten Datenbank-Dateien händisch zu kopieren. Dies ist in der Version 12.2 auch weiterhin möglich, allerdings gibt es nun auch die Möglichkeit, anstatt einer „*.xml“- eine komprimierte „*.pdb“-Datei zu erstellen, mit dem Vorteil, dass in diesem Dateiformat die komplette PDB enthalten ist.

Welche Schritte sind nun notwendig, um eine PDB aus einer On-Premise in eine DBCS-Umgebung zu verschieben? Im ersten Schritt muss die PDB auf der Quell-Datenbank gestoppt werden. Im Anschluss wird die gestoppte PDB ausgehängt (siehe Listing 2).

Anschließend kann die Archive-Datei, in diesem Fall „kuf3.pdb“, auf den Zielhost in der Oracle-Public-Cloud kopiert werden. Danach lässt sich mit einer Prozedur überprüfen, ob der Inhalt der Archive-Datei, in diesem Fall die PDB „KUF3“, mit dem neuen Container kompatibel ist (siehe Listing 3).

Sollte die PDB nicht kompatibel sein, lässt sich der Grund für die Inkompatibilität mit der View „PDB_PLUG_IN_VIOLATIONS“ abfragen. Da in diesem Fall kein Problem vorliegt, kann die PDB importiert werden (siehe Listing 4).

PDB-Clone mittels Oracle Enterprise Manager 13.2

Bevor eine PDB mithilfe von Oracle Enterprise Manager (OEM) in die Cloud geklont werden kann, muss die DBCS-Instanz mit dem Hybrid-Cloud-Gateway zum OEM hinzugefügt werden. *Abbildung 2* zeigt die SSH-Verbindung zwischen den Agenten in der Oracle-Public-Cloud und dem Oracle Enterprise Manager On-Premise.

Nachdem die Hybrid-Cloud-Agent-Installation abgeschlossen ist, erfolgt die Auswahl der PDB, in diesem Fall „KUF1“, die geklont werden soll. Dazu im OEM im Menü „Oracle Database“ die Optionen „Cloning“ und „Clone to Oracle Cloud“ auswählen (siehe *Abbildung 3*).

Im nächsten Fenster wird Folgendes definiert:

- Name der PDB auf dem Zielsystem
- PDB Administrator Credentials

- Name des Containers, in den die PDB hinzugefügt werden soll

Nun kann das Klonen beginnen, die einzelnen Schritte des Clone-Jobs können über das Fenster „Procedure Activity“ überwacht werden. Nach dem Klonen steht die PDB in der neuen CDB zur Verfügung. Leider erfolgte dieser Vorgang nicht fehlerfrei. Der Schritt „Post clone PDB creation operations“ bricht mit der Fehlermeldung „ORA-24964“ ab.

Die Recherche in My Oracle Support ergab, dass diese Fehlermeldung nicht bekannt ist. Der eröffnete Service Request führte zu einem Bug (siehe Abbildung 4). Auch nach dem Einspielen der neusten Oracle-Enterprise-Manager-Updates und Plug-ins vom Oktober 2017 bleibt der Fehler bestehen.

PDB-Klon mittels SQL Developer

Um eine PDB in die Oracle-Public-Cloud mithilfe des SQL Developer zu klonen, muss dieser auf dem Quell-Server installiert sein. Ist dies nicht der Fall, erhält man wie der Autor beim Klonen in die Oracle-Cloud die folgende Fehlermeldung: „Quelle ist keine lokale Datenbank; führen Sie auf dem Quellrechner sqldeveloper aus.“ Ein Hinweis dazu ist im SQL-Developer-

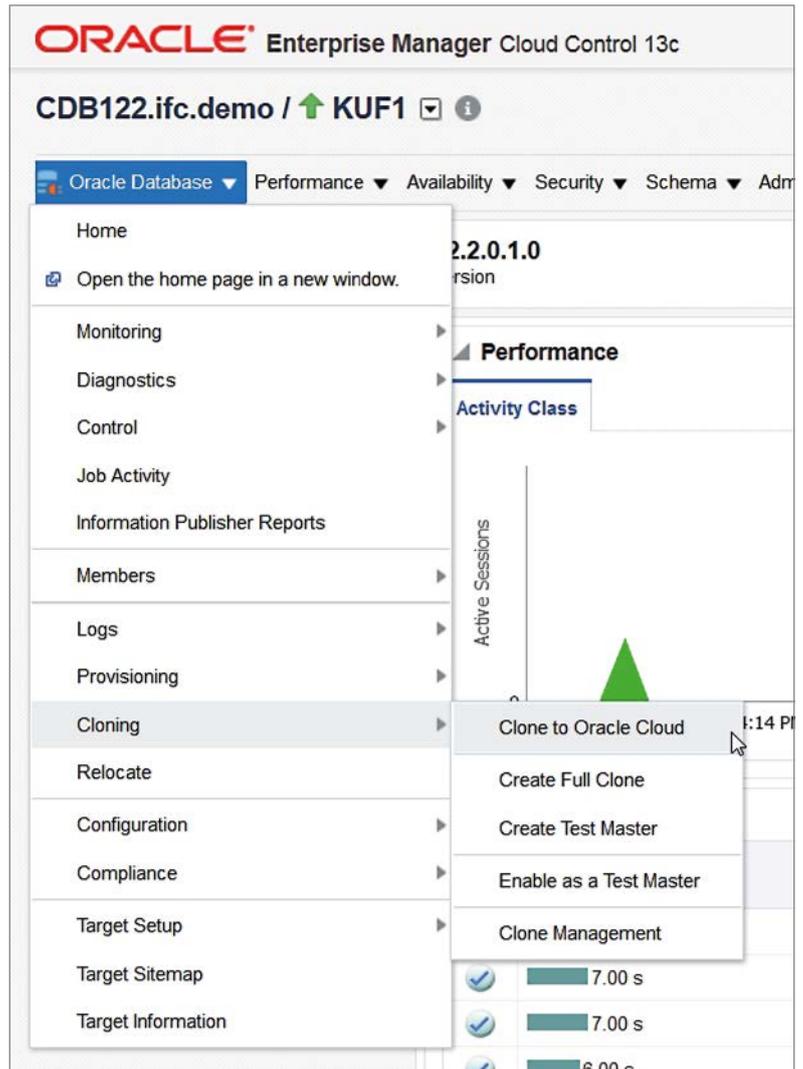


Abbildung 3: Auswahl „Clone to Oracle Cloud“

Bug 26117242 : Post clone PDB creation operation failes with ORA-24964

☑ **Bug Attributes**

Type	B - Defect	Fixed in Product Version	
Severity	2 - Severe Loss of Service	Product Version	13.2.2.0.0
Status	11 - Code/Hardware Bug (Response/Resolution)	Platform	226 - Linux x86-64
Created	May 22, 2017	Platform Version	ORACLE LINUX 6
Updated	Oct 20, 2017	Base Bug	N/A
Database Version	12.1.0.2	Affects Platforms	Generic
Product Source	Oracle	Knowledge, Patches and Bugs related to this bug	

☑ **Related Products**

Line	Enterprise Management	Family	Enterprise Manager Products
Area	Managing Databases using Enterprise Manager	Product	1366 - Enterprise Manager for Oracle Database

Hdr: 26117242 12.1.0.2 CLONEMGMT 13.2.2.0.0 DB_CLOUD PRODID-1366 PORTID-226
 Abstract: Post clone PDB creation operation failes with ORA-24964

Abbildung 4: Fehler beim Klonen mit OEM

```
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database open upgrade;
SQL> alter database local undo on;
SQL> shutdown immediate;
SQL> startup
```

Listing 5

```
SQL> create user c##ifl identified by welcome1 container=all;
SQL> grant sysoper to c##ifl container=all;
SQL> grant create session to c##ifl container=all;
```

Listing 6

```
SQL> create public database link cdb122_link connect to c##ifl identified
by welcome1 using 'CDB122';
SQL> create pluggable database KUF2_CL2 from KUF2@cdb122_link;
SQL> alter pluggable database KUF2_CL2 open;
```

Listing 7

```
# export ORACLE_SID=NONCDB;
SQL> create user MIG identified by mig;
SQL> grant create session, create pluggable database to MIG;
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database open read only;
```

Listing 8

```
SQL> create tablespace KUF datafile size 5M;
ORA-28374: typed master key not found in wallet
```

Listing 9

Handbuch, Kapitel 6.11 „Clone PDB to Oracle Cloud“, leider nicht zu finden.

Um eine Verbindung zu der DBCS-Instanz zu erstellen, muss zunächst ein neuer SSH-Host im SQL Developer hinzugefügt und konfiguriert sein. Anschließend kann über den Menüpunkt „PDB in Oracle Cloud klonen ...“ das Klonen gestartet werden. Anders als beim Oracle Enterprise Manager lässt sich hier der Name der zu klonenden PDB im Zielsystem nicht ändern. Mit der Meldung „Your PDB has been successfully cloned to the Oracle Cloud“ ist der Vorgang abgeschlossen.

PDB-Hot-Clone

Der Nachteil von PDB-Clone ist, dass während des Klonvorgangs die Quell-PDB den Anwendern nur lesend zur Verfügung steht („open read only“). Mit Einführung von PDB-Hot-Clone hat sich dies ab dem

Release 12.2 geändert. Nun steht die zu klonende PDB während des Klonvorgangs den Benutzern uneingeschränkt zur Verfügung. Voraussetzung dabei ist allerdings, dass sich die Quell-Datenbank im ArchiveLog-Modus befindet und das neue Feature „LOCAL UNDO“ verwendet wird. Mit dem Kommando „select property_value from database_properties where property_name='LOCAL_UNDO_ENABLED“ lässt sich prüfen, ob „LOCAL UNDO“ verwendet wird. Falls dieses nicht konfiguriert ist, sind einige Schritte notwendig, um diesen Modus zu aktivieren (siehe Listing 5).

Im nächsten Schritt wird auf der Quell-CDB noch ein Common User mit den folgenden Rechten angelegt (siehe Listing 6). Auf der Zielseite kann direkt nach dem Anlegen eines Datenbank-Links auf die Quell-Datenbank mit dem Klonen begonnen werden (siehe Listing 7). Im Moment ist PDB-Hot-Clone in die Oracle-Public-Cloud noch nicht über die Tools SQL De-

veloper und Oracle Enterprise Manager unterstützt.

NONCDB TO PDB

Viele Kunden-Datenbanken, die in On-Premise-Umgebungen betrieben werden, verwenden allerdings noch die klassische Architektur. Ab der Version 12 können nun auch Nicht-Container-Datenbanken mithilfe des PDB-Klonmechanismus in die Oracle-Cloud geklont werden. Dazu muss auf der Quell-Datenbank ein Benutzer existieren, der das Recht „create pluggable database“ besitzt. Dies ist auch der Grund, aus dem dieser Mechanismus erst bei Datenbanken ab Version 12 funktioniert. Anschließend müssen die Quell-Datenbanken noch im „read only“-Modus geöffnet sein (siehe Listing 8).

Auf der Ziel-Datenbank wird nun mit „create database link noncdb_link connect to mig identified by mig using ‚NONCDB“ ein Datenbank-Link zur Quelldatenbank erzeugt. Das Kommando „CREATE PLUGGABLE DATABASE PDBX FROM NONCDB@noncdb_link;“ wandelt eine NONCDB in eine PDB. Die Besonderheit an diesem Befehl ist, dass anstatt eines PDB-Namens, der ja bekanntlich nicht vorhanden ist, das Pseudonym „NONCDB“ zum Einsatz kommt. Nach dem Ende des Vorgangs muss auf der erzeugten PDB, in diesem Falle „PDBX“, noch das Skript „@\$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql“ ausgeführt werden.

Vorsicht, Falle!

Egal, ob man nun die PDB mit dem „unplug/plug clone“- oder „hot clone“-Kommando erstellt hat, die PDBs wurden erfolgreich in die Oracle-Cloud geklont und stehen den Anwendern zur Verfügung. Was passiert aber, wenn man bei einer der geklonten PDBs einen neuen Tablespace hinzufügen möchte (siehe Listing 9)?

Der Tablespace kann nicht erstellt werden, da der Master-Key der PDB nicht im Wallet gefunden wird. Wie schon erwähnt, werden alle User-Daten in der Oracle-Cloud verschlüsselt, dazu wurde der Datenbank-Parameter „ENCRYPT_NEW_TABLESPACES“ eingeführt. In der Oracle-Public-Cloud steht dieser auf „CLOUD_ONLY“. Neue Tablespaces werden automatisch mit

AES128 verschlüsselt. Die Abfrage der View „V\$ENCRYPTION_WALLET“ auf der geklonten PDB (Version 12.2) zeigt, dass ein „AUTOLOGIN“-Wallet verwendet wird und der Status auf „OPEN_NO_MASTER_KEY“ steht (siehe Listing 10).

Das Problem ist also, dass für die geklonte PDB noch kein Master-Key im zentralen Wallet („ewallet.p12“) hinterlegt ist. Doch wie kann ein solcher Key neu erzeugt werden und gibt es dabei Unterschiede je nach verwendeter Datenbank-Version? Werfen wir zunächst ein Blick auf 12.1: Der Versuch, in der geklonten PDB einen neuen Master-Key mit dem SQL-Kommando „administer key management set key identified by „xxxxx“ with backup;“ zu erzeugen, schlägt mit der Meldung „ORA-46658: keystore not open in the container“ fehl. Diese Meldung macht den Anschein, dass der Keystore nicht geöffnet ist, obwohl die vorherige Abfrage „OPEN_NO_MASTER_KEY“ lieferte. In Wirklichkeit bedeutet diese Fehlermeldung, dass die Keys nicht modifiziert werden können, solange ein „AUTOLOGIN“-Wallet benutzt wird.

Wie lässt sich „AUTOLOGIN“ deaktivieren? Dazu wechselt man in das Verzeichnis „\$ORACLE_BASE/admin/<SID>/tde_wallet“ und löscht beziehungsweise nennt die Datei „cwallet.sso“ um. Als Nächstes erfolgt ein Restart der Datenbank, um das Wallet zu schließen. Nun ist es möglich, mit dem SQL-Befehl „administer key management set keystore open identified by „xxxxx“ container=all;“ das Wallet manuell zu öffnen. Nach dem Wechseln in die geklonte PDB wird ein neues Wallet („ewallet.p12“) erzeugt. Das „AUTOLOGIN“-Wallet kann ab diesem Zeitpunkt nicht mehr verwendet und muss ebenfalls, auf Ebene der CDB, neu erzeugt werden (siehe Listing 11). Nach einem weiteren Restart der Datenbank wird diese mit dem neuen „AUTOLOGIN“-Wallet geöffnet und auf der geklonten PDB kann nun ein neuer Tablespace erstellt werden.

Man sieht, dass der Aufwand, der pro geklonte PDB betrieben werden muss, doch erheblich ist. Wenn eine Migration von mehreren PDBs am Stück erfolgt, kann die Downtime minimiert werden, indem man die Wallet-Migration erst am Ende der PDB-Migration durchführt. Sollte sich diese allerdings über einen längeren Zeitraum erstrecken, wäre eine Downtime nach jeder PDB-Migration unter Umständen nicht mehr tragbar.

```
SQL> select status, wallet_type from v$encryption_wallet;
STATUS          WALLET_TYPE
OPEN_NO_MASTER_KEY  AUTOLOGIN
```

Listing 10

```
SQL> administer key management create auto_login keystore from
keystore '/u01/app/oracle/admin/ORCL/tde_wallet/' identified by "xxxxx";
SQL> administer key management set keystore close identified by "xxxxx"
container=all;
```

Listing 11

```
SQL> administer key management set keystore open force keystore
identified by "xxx";
SQL> administer key management set key force keystore identified by
"xxx" with backup;
```

Listing 12

TABLESPACE_NAME	ENC
-----	---
SYSTEM	NO
SYSAUX	NO
UNDOTBS1	NO
TEMP	NO
USERS	NO

Listing 13

TABLESPACE_NAME	ENC
-----	---
SYSTEM	NO
SYSAUX	NO
UNDOTBS1	NO
TEMP	NO
USERS	YES

Listing 14

TABLESPACE_NAME	ENC
-----	---
SYSTEM	NO
SYSAUX	NO
UNDOTBS1	NO
TEMP	NO
USERS	YES

Listing 15

Oracle hat diese Problematik erkannt und in der Version 12.2 die Klausel „FORCE KEYSTORE“ eingeführt, die nun ein Key-Management bei gleichzeitiger Benutzung des AUTOLOGIN-Wallet ermöglicht. Nach dem Anmelden an der geklonten PDB können mit den beiden folgenden Kommandos das Wallet aktualisiert und anschließend die gewünschten Tablespaces erzeugt werden (siehe Listing 12).

Doch wie sieht es bei den bestehenden Tablespaces der geklonten PDBs aus? Sind dort die User Tablespaces verschlüsselt? Die SQL-Abfrage „select tablespace_name, encrypted from dba_tablespaces;“ liefert das folgende Ergebnis (siehe Listing 13). Lis-

ting 14 zeigt zum Vergleich die Abfrage auf einer bestehenden PDB.

Wie aus den beiden Abfrage-Ergebnissen zu erkennen ist, ist der Tablespace „USERS“ der geklonten PDB nicht verschlüsselt. Dies sollte dringend nachgeholt werden. Wurde das Wallet, wie oben beschrieben, um die neue PDB ergänzt, lässt sich der Tablespace mit „SQL> alter tablespace USERS encryption online encrypt;“ verschlüsseln. Da in DBCS Oracle-Managed-Files verwendet werden, kann bei der Verschlüsselung des Datafiles auf die „FILE_NAME_CONVERT“-Option verzichtet werden (siehe Listing 15).

```
SQL> create user mig identified by welcome1;
SQL> grant create session, create pluggable database to mig;
SQL> alter pluggable database close;
SQL> alter pluggable database open read only;
```

Listing 16

```
SQL> create database link pdb121_link connect to mig identified by mig
using 'PDB121';
SQL> create pluggable database PDB2 from pdb11@pdb121_link;
```

Listing 17

```
create pluggable database PDB2 from pdb11@pdb121_link
ERROR at line 1:
ORA-17628: Oracle error 17630 returned by remote Oracle server
ORA-17630: Mismatch in the remote file protocol version client server
```

Listing 18

```
SQL> alter session set container=PDB2;
Session altered.
SQL> alter pluggable database open;
Warning: PDB altered with errors.
SQL> select name, message from PDB_PLUG_IN_VIOLATIONS
where status<>'RESOLVED' and CON_ID=8 order by time;
NAME          MESSAGE
-----
PDB2          PDB's version does not match CDB's version: PDB's
              version 12.1.0.2.0. CDB's version 12.2.0.1.0.
PDB2          CDB is using local undo, but no undo tablespace
              found in the PDB.
```

Listing 19

```
SQL> alter pluggable database PDB2 close;
Pluggable database altered.
SQL> alter pluggable database PDB2 open upgrade;
Pluggable database altered.
SQL> exit
# cd $ORACLE_HOME/bin
# ./dbupgrade -c 'PDB2'
# sqlplus / as sysdba
SQL> alter pluggable database PDB2 open;
Pluggable database altered.
SQL> exit
# /u01/app/oracle/product/12.2.0/dbhome_1/perl/bin/perl catcon.pl -n 1
-e -b utlrp -d ''.''' utlrp.sql
```

Listing 20

```
SQL> alter session set container=PDB2;
Session altered.
SQL> create undo tablespace PDB2_UNDO;
create undo tablespace PDB2_UNDO
*
ERROR at line 1:
ORA-28374: typed master key not found in wallet
```

Listing 21

PDB-Klon von 12.1 nach 12.2

Ist ein Klonen einer NONCDB/PDB von der Version 12.1 nach 12.2 möglich? Die Oracle-Dokumentation gibt zu diesem Thema leider keine Auskunft. Der Autor hat es am Beispiel einer PDB mit der Version 12.1 getestet. Dazu hat er auf der Quell-PDB einen User-MIG mit den Rechten „create session“ und „create pluggable database“ angelegt und die PDB im „read only“-Modus geöffnet (siehe Listing 16).

Auf der Ziel-CDB wird ein Datenbank-Link auf die Quell-DB erstellt und der Klonvorgang gestartet (siehe Listing 17). Das Klon-Kommando schlägt mit folgender Fehlermeldung fehl (siehe Listing 18). Es handelt sich hierbei um den Bug 18633374. Weitere Informationen findet man in der Oracle-Support-Note 18633374 „Error copying across remote servers with ASMCMD CP command (ASMCMD-8016, ORA-17628, ORA-17630, ORA-6512 Doc ID 18633374.8)“.

Nach dem Einspielen des Patches, der übrigens auch hilft, wenn man kein ASM im Einsatz hat, konnte der Klon der PDB erstellt werden. Wie die folgende Abfrage der View „PDB_PLUG_IN_VIOLATIONS“ zeigt, ist die geklonte PDB noch von der Version 12.1 und verwendet noch kein „LOCAL UNDO“ (siehe Listing 19).

Der nächste Schritt besteht darin, die PDB auf die aktuelle Version zu migrieren. Dazu muss die PDB im Modus „OPEN UPGRADE“ geöffnet werden. Im Anschluss kann das Upgrade der PDB mittels „DBUPGRADE“-Kommando erfolgen (siehe Listing 20). Nach erfolgreicher Migration der PDB auf die Version 12.2 wird im nächsten Schritt der lokale UNDO-Tablespace erzeugt (siehe Listing 21).

Das Erstellen des UNDO-Tablespace bricht mit der mittlerweile bekannten Fehlermeldung ab, da der Master-Key für die erstellte PDB im Wallet fehlt. Leider lässt sich der neue Master-Key nicht erzeugen, da die PDB im Restricted Mode geöffnet ist (siehe Listing 22).

Der Grund für den Restricted Mode ist der fehlende lokale „UNDO“-Tablespace, dieser kann aber, wie gesehen, wegen des fehlenden Master-Keys nicht erzeugt werden. Hier liegt ein klassischer Dead-lock vor. Die Lösung dieses Dilemmas besteht aus den folgenden Schritten:

- „LOCAL UNDO“-Mode ausschalten
- Master-Key für die PDB erzeugen
- „UNDO“-Tablespace für die PDB erzeugen

- User-Tablespaces, in diesem Fall „USERS“ und „KUF“ der PDB, verschlüsseln
- „LOCAL UNDO“-Mode wieder anschalten

Noch ein Hinweis zum Schluss: Die Verschlüsselungsproblematik nach dem Klonen von PDBs ist kein reines Oracle-Datenbank-Cloud-Problem, sondern tritt bei der Verwendung der Transparent-Database-Encryption-Option bei CDB-Architektur On-Premise ebenfalls auf. Trotzdem ist es schade, dass Oracle im Handbuch „Using Oracle Database Cloud Service“ auf diese Problematik nicht eingeht. Das Kapitel „Remote Cloning einer PDB“ endet, nachdem die Pluggable Datenbank geöffnet wurde. Dass zu diesem Zeitpunkt weder ein Master-Key für die PDB existiert noch die User-Tablespaces unverschlüsselt sind, wird dabei nicht erwähnt.

Fazit

Wie man sieht, ist mehr als ein Knopfdruck notwendig, um eine Datenbank beziehungsweise eine Pluggable Datenbank

```
SQL> administer key management set keystore open force keystore identified by "xxxx";
keystore altered.
SQL> administer key management set key force keystore identified by "xxxx" with backup;
administer key management set key force keystore identified by "xxxx" with backup
*
ERROR at line 1:
ORA-28442: Rekey of the TDE master encryption key is not permitted when the database is in restricted mode.
```

Listing 22

in die Oracle-Cloud zu klonen. Es ist jedoch auch zu erkennen, dass sich Oracle auf dem richtigen Weg befindet und mit der Version 12.2 einige Schwachstellen der vorherigen Version beseitigt sind. Zum Abschluss noch einmal die Vorteile der CDB-Architektur des aktuellen Release im Überblick:

- Mittels „Hot Clone“ nahezu ohne Downtime eine PDB klonen
- Vereinfachtes Key-Management bei Verwendung von Transparent Data Encryption (TDE)

- Unterschiedliche Zeichensätze pro PDB
- Lockdown Profile pro PDB



Kai Uwe Fischer
kai-uwe.fischer@logicalis.de

DOAG Legal Council gestartet

Juristische Aspekte und vielfältige rechtliche Herausforderungen spielen auch in der Arbeit der DOAG und der Vertretung der Mitglieder-Interessen eine immer größere Rolle. Die fachlichen Kompetenzen der DOAG Competence Center werden zukünftig durch das am 20. April 2018 gegründete DOAG Legal Council ergänzt. Dieses wird die Arbeit der DOAG-Gremien und Competence Center mit einer Expertise im IT-Recht begleiten. Ob Vertragsgestaltung, Lizenzierung oder Datenschutz – das DOAG Legal Council wird rechtliche Themen für IT-Spezialisten zugänglicher machen und der DOAG bei der Vertretung der Interessen der Mitglieder zur Seite stehen. Denn ein Anlass zu dieser Initiative sind die seit Jahren diskutierten Anliegen zu rechtlichen Fragen der DOAG, für die Oracle bisher keine befriedigenden allgemeingültigen Lösungen lieferte.

Die Gründungsversammlung mit den ersten Mitgliedern des DOAG Legal Council fand parallel zur diesjährigen Delegierten-

versammlung des Vereins statt. Die zukünftig noch um weitere Mitglieder ergänzte Runde bringt fundiertes Wissen zusammen und bildet die Plattform eines Austausches fachlicher und rechtlicher Betrachtungen. Schon beim ersten Treffen wurde die Erörterung der viel diskutierten Themen „Lizenzierung von Oracle-Produkten in virtualisierten Umgebungen“ und „Datenschutz in der Oracle-Cloud“ als prioritäre Aufgaben des Council identifiziert und mit ersten Überlegungen begonnen.

Als Koordinator des DOAG Legal Council konnte der ehemalige DOAG-Vorstandsvorsitzende Dr. Dietmar Neugebauer Rechtsanwältin Dr. Jana Jentzsch aus der Sozietät Jentzsch IT, Hamburg, und Rechtsanwalt Dr. Thomas Thalhofer aus der Sozietät Nerr, München, begrüßen. Rechtsanwalt Dr. Jan Bohnstedt von Bartsch Rechtsanwälte, Frankfurt, ist weiteres Mitglied. Der langjährige Berater der DOAG, Rechtsanwalt Carsten J. Diercks, wird in dieser Runde die juristische Seite koordinieren.

Dr. Dietmar Neugebauer betonte bei der Gründung: „Wir freuen uns, mit dem DOAG Legal Council ab sofort bei rechtlichen Fragen auf die Fachkompetenz spezialisierter Rechtsanwälte zurückgreifen zu können. Wir sind sicher, dass wir gemeinsam mehr für unsere Mitglieder erreichen können, sei es über unsere Informationsplattform, oder vor allem im Gespräch mit Oracle.“

Die Mitglieder des Council sind von der DOAG ausgewählt worden und engagieren sich ehrenamtlich. Dabei wird keine Rechtsberatung im Einzelfall erfolgen, sondern allgemeine Betrachtungen von rechtlichen Fragen. Die Ergebnisse werden unter anderem in Form von Artikeln oder Vorträgen im Rahmen der DOAG in das DOAG Netzwerk eingebracht. Für darüber hinausgehende Beratung können die Mitglieder der DOAG direkten Kontakt zu den Anwälten aufnehmen. Weitere Informationen unter „<https://www.doag.org/index.php?id=1072>“.



Die autonome Datenbank: ADWC für Warehouse-Anwendungen

Ulrike Schwinn, ORACLE Deutschland B.V. & Co. KG

Während der Oracle Openworld 2017 hat Larry Ellison in seiner Keynote die erste „Self-Driving“-Oracle-Datenbank angekündigt – die sogenannte „autonome Datenbank“. Jetzt ist es soweit: Seit Ende März gibt es als erste Fassung der autonomen Datenbank im Datacenter in Ashburn den Autonomous Data Warehouse Cloud Service (ADWC). Mit der Inbetriebnahme im Frankfurter Rechenzentrum wird sehr zeitnah gerechnet; zudem werden weitere autonome Technologien in diesem Jahr folgen. Der Artikel zeigt die Funktionsweise des ADWC und mögliche Anwendungsfälle.

Die Menge der produzierten Daten wächst ungebrochen. Gründe dafür sind neben der Digitalisierung von Inhalten und dem Austausch von Daten vor allem die Integration digitaler Mess-, Steuer- und Regelsysteme (eingebettete Systeme) in Alltagsgegenstände sowie der Austausch und die Verarbeitung deren Daten. Die Analyse und Auswertung großer Datenmengen stellen dabei neue Herausforderungen an das Datenmanage-

ment und die gesamte IT-Infrastruktur dar.

Gefragt ist eine Infrastruktur, die dynamisch abrufbare Rechen- und Speicher-Ressourcen erlaubt. Es soll nur so viel gezahlt werden, wie tatsächlich an Ressourcen benötigt wird. Darüber hinaus ist eine gute Abfrage-Performance zu gewährleisten, um schnelle und flexible Abfrage-Ergebnisse zu generieren. Zu guter Letzt soll der Mitarbeiter ausreichend

Freiraum erhalten, um sich um die Analyse-Systeme kümmern zu können, statt seine Zeit mit den erforderlichen Datenbank-Maintenance-Aufgaben zu verbringen.

Hier kommt nun die autonome Datenbank ins Spiel. Mit ihren Grundeigenschaften „Self-Driving“, „Self-Securing“ und „Self-Repairing“ verspricht sie den genannten Anforderungen gerecht zu werden. Das Ziel ist, durch Automatisierung sich wiederholender, monotoner Datenbank-Aufgaben

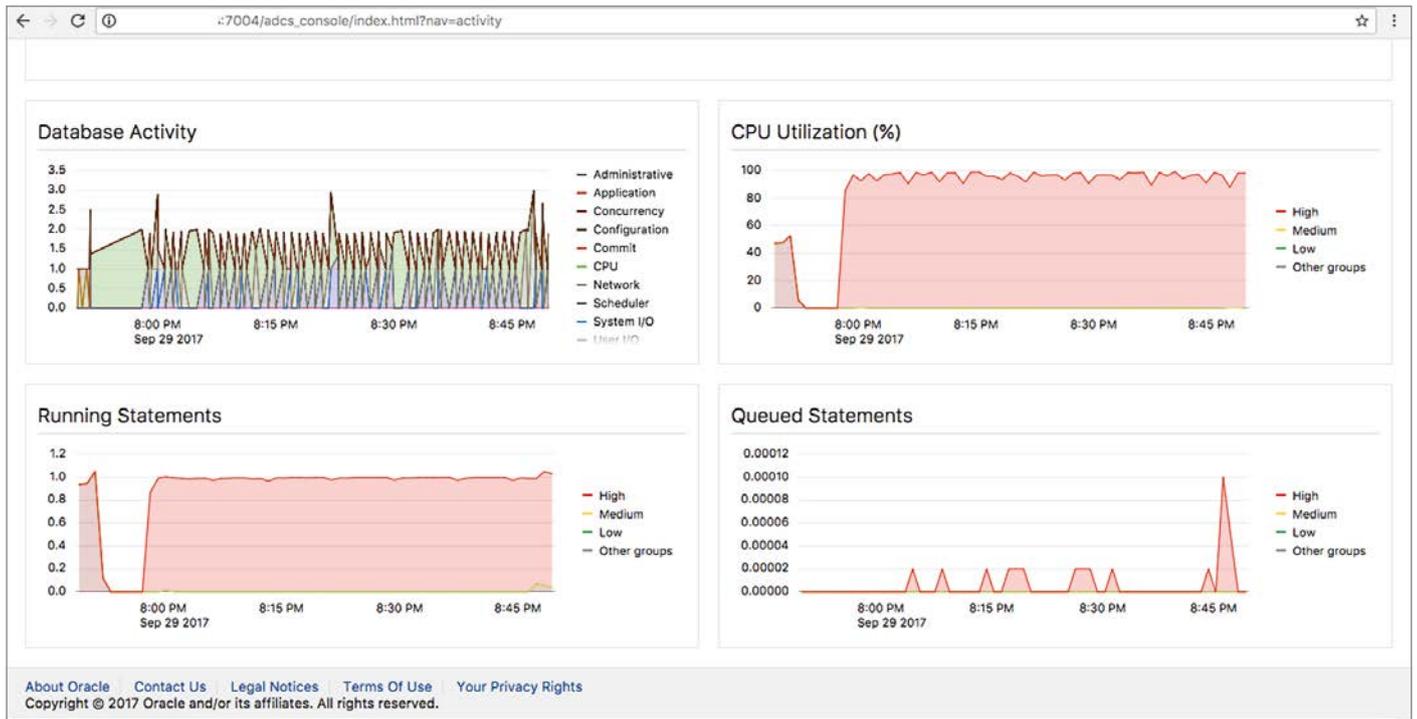


Abbildung 1: Monitoring über die Service-Konsole

und routinemäßiger Datenbank-Wartungsarbeiten Freiräume für andere Tätigkeiten zu schaffen. So werden beispielsweise manuelles Tunen, Patchen, Upgraden oder ein Backup/Recovery im Fehlerfall überflüssig.

Die Verschlüsselung der Daten, sichere Datenbank-Verbindungen (auch Connections) und das automatisierte Einspielen

der aktuellsten Security-Patches sorgen zusätzlich für hohe Sicherheit. Das Anfordern der Rechenpower und des Speichers nach Bedarf bildet dabei das Grundgerüst einer autonomen Oracle-Datenbank. Die Datenbank kann im laufenden Betrieb wachsen oder auch verkleinert werden, um Ressourcen einzusparen.

Das bedeutet ADWC

Die Automatisierung der Datenbank auf Basis des aktuellsten Datenbank-Release 18c, die Nutzung der Cloud-Infrastruktur und die Verwendung der Exadata-Infrastruktur sind Hauptmerkmale. ADWC ist also ein Public Cloud Service – die Inbetriebnahme für Cloud@Customer ist für die Zukunft eingeplant. Schon bei der Durchführung der Provisionierung zeigt sich, wie einfach und schnell der Service verfügbar ist. Nach ein paar Klicks und der Eingabe von wenigen Informationen wie Größe (in Terabyte), Anzahl CPUs, Name der Datenbank und Passwort des Administrators ist der Service innerhalb von 30 Sekunden erstellt und bereit für den Zugriff via SQL*Net.

Im Unterschied zum manuellen Konfigurieren und Erstellen einer Datenbank oder auch eines normalen Datenbank-Service ist beispielsweise weder die Festlegung auf ein bestimmtes Datenbank-Release noch das Einrichten des Backups erforderlich. Zudem ist es unwichtig, wie groß Speicheranforderungen und Rechenpower gewählt wurden; die gewählten Größen sind im laufenden Betrieb änderbar.

Mit ADWC muss keine Hardware konfiguriert und verwaltet oder Software installiert werden – es handelt sich um eine Exadata-Umgebung, die „ready-to-use“ ist.

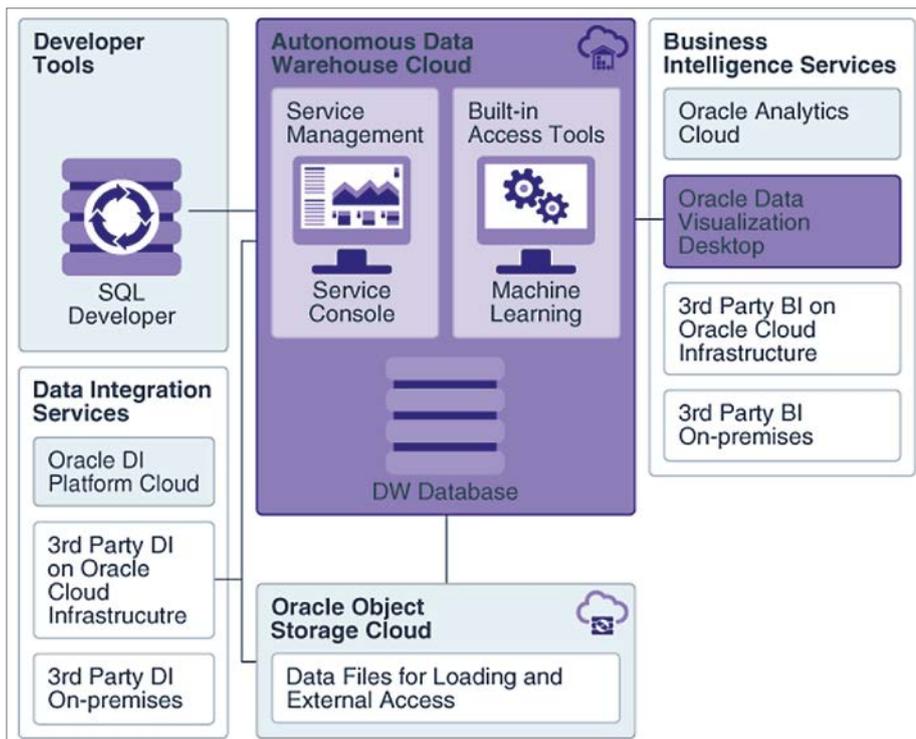


Abbildung 2: Mögliche Anbindungen

Die langjährige Erfahrung von Oracle im Data-Warehouse-Umfeld fließt dabei in das Setup des Service ein und garantiert von Anfang an hohe Performance. Technisch gesehen hat man es mit einer Plugable Database (PDB) zu tun, die um weitere Datenbank-Schemata und zusätzliche User-Tabellen erweitert werden kann.

Die Daten werden entweder aus dem Oracle-Object-Storage oder Amazon S3 direkt in die Datenbank geladen und automatisch komprimiert sowie verschlüsselt abgelegt. Aber auch ohne Ladevorgang ist es möglich, über eine spezielle External-Table-Konfiguration auf die Daten zuzugreifen. Das Setzen von Initialisierungsparametern sowie das Anlegen oder Erweitern von Tablespaces ist überflüssig geworden. Die Cloud-Console sowie die Verwendung über Rest-API ermöglichen Basisaufgaben wie Starten, Stoppen, Löschen und Skalieren der Rechenpower beziehungsweise des Storage.

Allgemeine Informationen über die Nutzung des Service – wie beispielswei-

se verbrauchter Speicherplatz und CPU in Prozent, Anzahl der SQL-Statements und durchschnittliche Laufzeit – finden sich in der Service-Konsole (siehe Abbildung 1). Weitere Monitoring-Informationen erhält man über den Menüpunkt „Activity“ oder im SQL Developer.

Die drei verfügbaren Service-Kategorien „LOW“, „MEDIUM“ und „HIGH“ bieten die Zuteilung der Ressourcen an unterschiedliche Anforderungen. Wie schon erwähnt, ist kein manuelles Tuning erforderlich; so ist standardmäßig Parallelität aktiviert. Der Grad der Parallelität der SQL-Anweisungen wird abhängig von der Anzahl der OCPUs im System und dem gewählten Datenbank-Service, mit dem sich der Benutzer verbindet, festgelegt. Außerdem werden automatisch bei Direct-Load-Operationen die Optimizer-Statistiken gesammelt.

ADWC sichert die Datenbank automatisch – wobei Voll-Backups im Wochenrhythmus und inkrementelle Backups täglich eingeplant sind. Die Aufbewah-

rungsfrist für Backups beträgt dabei 60 Tage. In dieser Zeit lässt sich die Datenbank zu einem beliebigen Zeitpunkt wiederherstellen. Wird ein Point-in-Time-Recovery eingeleitet, entscheidet ADWC, welches Backup für eine schnellere Wiederherstellung zum Einsatz kommen soll.

Natürlich gibt es auch einige Einschränkungen bei der ADWC-Verwendung. Dabei handelt es sich etwa um Kommandos, die entweder im autonomen Betrieb nicht erwünscht beziehungsweise überflüssig sind oder mit der Verfügbarkeit in der ersten Release-Version zusammenhängen. So ist es beispielsweise nicht möglich, Tablespaces anzulegen und zu erweitern oder Indizes zu erstellen. Beim Versuch erhält man die Fehlermeldung „ORA-01031: insufficient privileges“. Auch Datenbank-Links zu anderen Datenbanken sind aus Sicherheitsgründen nicht erlaubt. Alle Einschränkungen sind im Handbuch (siehe „https://cloud.oracle.com/en_US/datawarehouse/documentation“) vermerkt.

ORAWORLD

Das e-Magazine für alle Oracle-Anwender!

EOUC
E MEA
O RACLE
U SERGROUP
C OMMUNITY

- Neues aus der Oracle-Welt
- Einblicke in andere User Groups weltweit
- Leben und Arbeiten heute und morgen
- Neues (und Altes) aus der Welt der Nerds

Jetzt Artikel
einreichen oder
Thema vorschlagen!

Bis
9. August 2018

Jetzt e-Magazine herunterladen

www.oraeworld.org



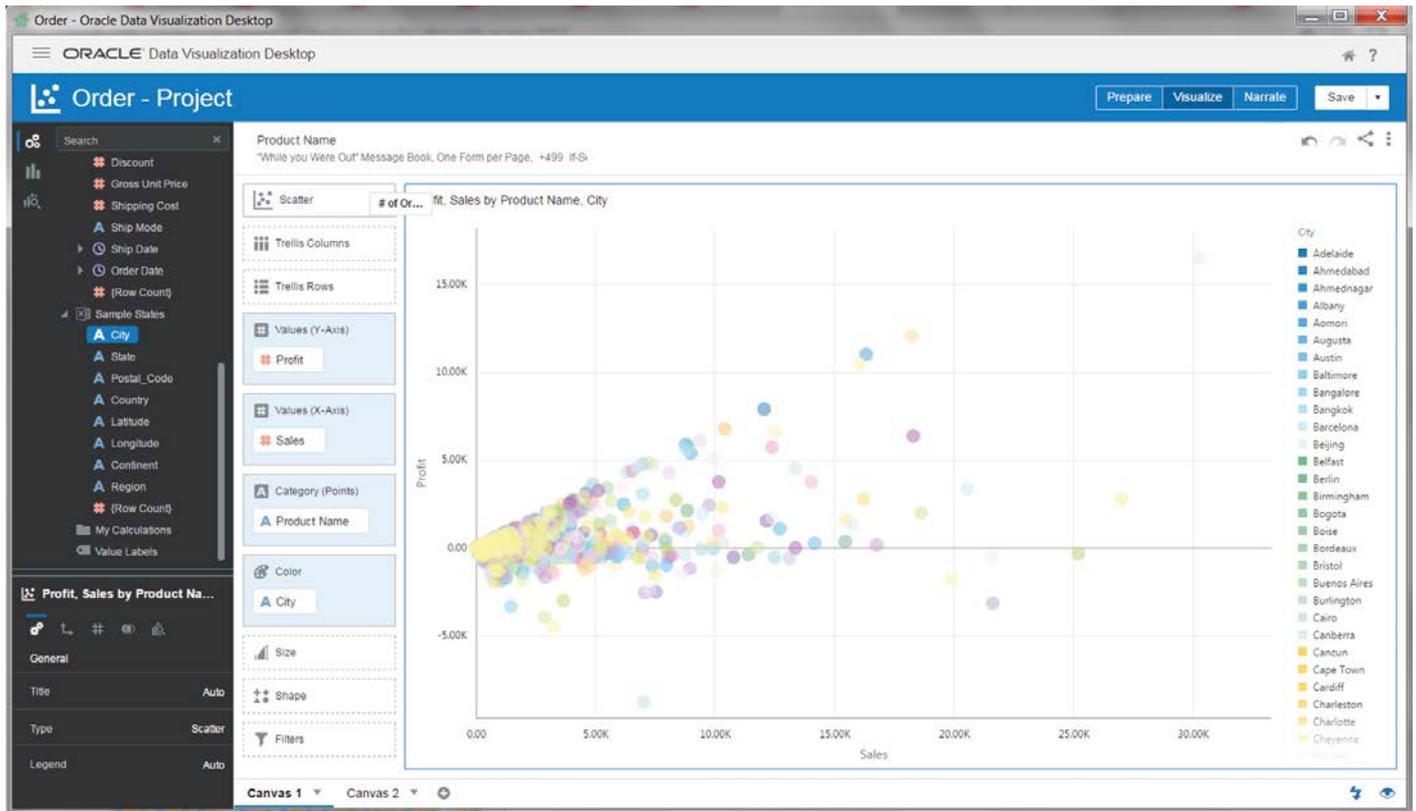


Abbildung 3: ADWC mit Oracle Data Visualization Desktop

ADWC im praktischen Einsatz

Da ADWC auf der Oracle-Datenbank basiert, können Werkzeuge verwendet werden, die diese unterstützen. Tools und Anwendungen, die entweder im eigenen Rechenzentrum oder in einer Public Cloud verfügbar sind, verbinden sich mit der ADWC über Standard-SQL*Net-Verbindungen (siehe Abbildung 2). Oracle-Analytics-Cloud und andere Oracle-Cloud-Services sind dabei für ADWC vorkonfiguriert.

Im SQL Developer (ab Version 17) beispielsweise wird im Feld „Connection Type“ der Wert „PDB-Cloud“ ausgewählt, mit dem die Konfigurationsdatei, die man vorab von der Konsole geladen hat, angegeben werden kann. SQL Developer zeigt dann eine Liste der verfügbaren Verbindungen im Feld „Service“ an. ADWC selbst beinhaltet eine Cloud-basierte Notebook-Anwendung, die eine einfache Abfrage und Daten-Visualisierung ermöglicht. Das Notebook ist für den Einsatz mit anderen Business-Intelligence-Anwendungen konzipiert. Darüber hinaus kann man mit Oracle Data Visualization Desktop auf einfache Weise Visualisierungen und Projekte erstellen, um Trends in den Unter-

nehmensdaten aufzuzeigen (siehe Abbildung 3).

Fazit

ADWC zeichnet sich durch leichte Bedienbarkeit, schnelle Provisionierung und hohe Performance aus. Dabei bietet ADWC ohne Investition in Hardware „out of the box“ Exadata-Performance. Die Schnittstellen sind einfach handhabbar, und der Service kann von Fachabteilungen verwaltet werden.

In der ersten Version ist noch mit Einschränkungen und gewissen Begrenzungen von Features zu rechnen. Diese sind in kommenden Cloud-Release-Zyklen sicherlich erweitert oder behoben.

Dadurch, dass die Ressourcen sehr flexibel einsetzbar und dynamisch anpassbar sind, lässt sich der Service sehr kostengünstig nutzen. Im Cost Estimator (siehe „<https://cloud.oracle.com/cost-estimator>“) lässt sich prüfen, welche Kosten für die Anwendungen veranschlagt werden. ADWC eignet sich beispielsweise für einzelne Data Marts, die neu aufgebaut werden sollen, Ad-hoc-Test-Umgebungen oder auch spontan verfügbare Rechen-Umgebungen für Spitzenlasten. Am besten einfach ausprobieren ...

Weitere Informationen

- <https://www.oracle.com/database/index.html>
- https://cloud.oracle.com/en_US/datawarehouse/features
- Cost Estimator: <https://cloud.oracle.com/cost-estimator>
- Demo als Features des Monats: <https://youtu.be/oFRisY20Ff8>
- Dokumentation: https://cloud.oracle.com/en_US/datawarehouse/documentation



Ulrike Schwinn
ulrike.schwinn@oracle.com



Architecting 4 the Cloud

Kai Weingärtner und Stefan Kühnlein, OPITZ CONSULTING Deutschland GmbH

Immer mehr Unternehmen erkennen die Vorteile der Cloud-basierten Services und sind bereit, ihre bestehenden Anwendungen in die Cloud zu migrieren. Mithilfe eines Lift-and-Shift-Ansatzes lassen sich inzwischen monolithische Anwendungen in der Cloud betreiben – allerdings profitieren diese nicht von den Vorteilen der Cloud wie Flexibilität, Skalierbarkeit und Elastizität. Um Anwendungen erfolgreich in der Cloud zu betreiben und somit die Vorteile der Cloud ausnutzen zu können, müssen sie als native Cloud-Anwendungen entwickelt sein.

Native Cloud-Anwendungen werden speziell für eine Cloud-Architektur entwickelt und nutzen deren Vorteile durch den Einsatz entsprechender Frameworks vollständig aus. Somit unterscheiden sie sich sehr stark von traditionellen Anwendungen, die auf Basis einer Schichten-Architektur aufgebaut sind. Die Open Data Center Alliance hat für die Bewertung der Reife einer Cloud-basierten Anwendung bestimmte Grade definiert (siehe Abbil-

dung 1). Um den höchsten Reifegrad zu erreichen, ist die Verwendung von Microservices beziehungsweise Self-Contained-Systems (SCS) zwingend erforderlich,

SCS vs. Microservice

Die wesentlichen Architektur-Patterns der SCS leiten sich von denen der Microservices ab. Insbesondere gehören die Ide-

en zur Durchsetzung der Isolation unabhängig einsetzbarer Einheiten dazu sowie die Ausrichtung organisatorischer und architektonischer Grenzen, der sogenannte „Bounded Context“. Auch in Bezug auf die Auswahl der Technologie, der verwendbaren Frameworks sowie der fehlenden Infrastruktur unterscheidet sich ein SCS nicht von einer Microservices-Architektur. Somit kann eine SCS sehr wohl als Spezialisierung der Microservices betrachtet

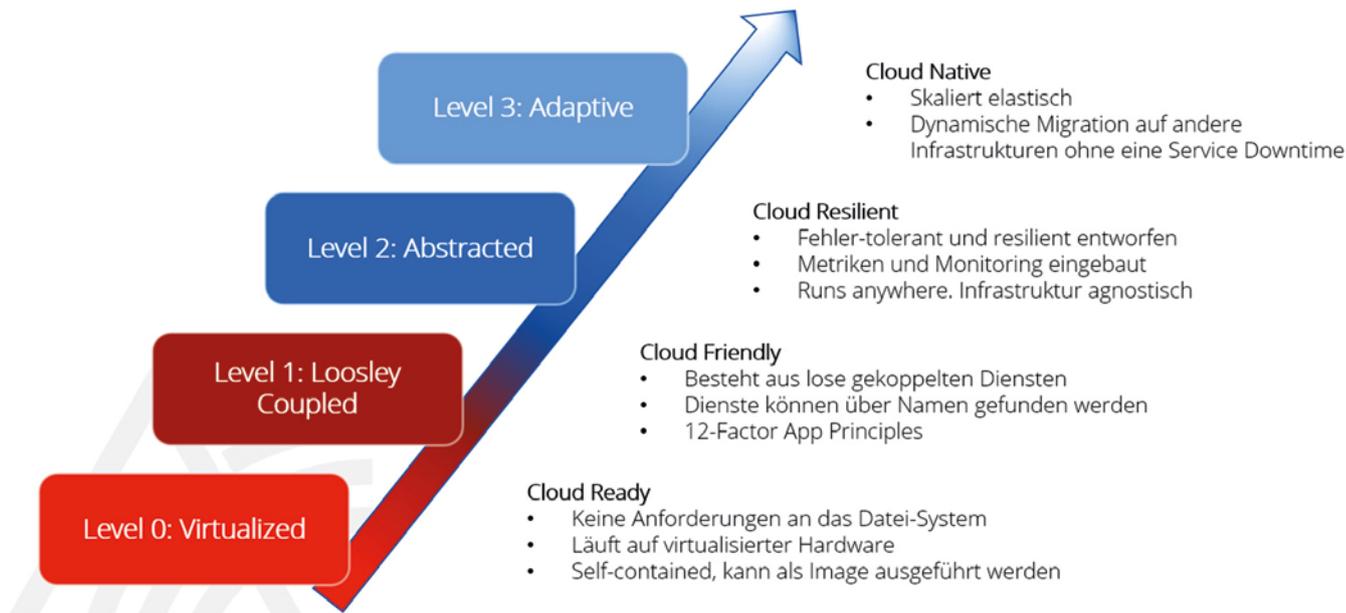


Abbildung 1: Reifegrade einer Cloud-basierten Anwendung (Quelle [1])

werden. Allerdings unterscheidet sich ein SCS in diesen Schlüssel-Attributen von einer Microservices-Architektur:

- Microservices-Architekturen sind in der Regel kleiner als ein SCS. Ein SCS sollte einen sinnvollen Mehrwert liefern und ein Team sinnvoll beschäftigen. In der Regel ist eine SCS eine eigenständige Web-Anwendung mit UI, Geschäftslogik und Daten.
- Während Microservices-Architekturen generell über keine UI verfügen, sollte der UI-Layer in ein SCS integriert sein.
- Idealerweise sollte zwischen den einzelnen SCS keine Kommunikation erforderlich sein. Eine Integration von unterschiedlichen SCS erfolgt auf Ebene der Web-Anwendung am besten über Hyperlinks.

Wie bei einer Microservices-Architektur sollte auch von einem SCS ein API zur Verfügung gestellt werden, damit die Funktionalität von anderen Services genutzt werden kann. Ebenso wie bei Microservices führt die Integration unterschiedlicher SCS zu einer sehr losen Kopplung und die einzelnen SCS lassen sich zu einem späteren Zeitpunkt unabhängig voneinander weiterentwickeln sowie gegebenenfalls austauschen. Sowohl Microservices als auch SCS eignen sich als Architektur-Pattern, um einen Monolithen in eine native Cloud-Anwendung zu überführen.

Domain Driven Design

Einer der wichtigsten Erfolgsfaktoren für eine gelungene native Cloud-Anwendung ist der richtige Service-Schnitt – und nicht zwangsläufig dessen Größe gemessen an den „Lines of Code“. Die zentrale Frage dabei ist: Wie zerlege ich den Monolithen, sodass die einzelnen Bausteine als unabhängige Einheiten eingerichtet und betrieben werden können.

Ein Ziel des Service-Schnitts ist die Aufteilung der Anforderungen auf die jeweiligen Fachdomänen, damit ein eintretendes Ereignis eine Kette von wertschöpfenden Ereignissen auslöst. Daher sollte zu Beginn eines neuen Cloud-Projekts eine gemeinsame Modellierung erfolgen.

Bei der Zerlegung einer Anwendung in ihre fachlichen Bestandteile kann der Ansatz des Domain Driven Design (DDD) helfen, ein standardisiertes Vorgehen, das eine interdisziplinäre Modellierung unterstützt, in der sowohl Entwickler als auch Fachexperten involviert sind. Im Jahr 2003 stellte Eric Evans in seinem Buch „Domain Driven Design: Tackling Complexity in the Heart of Software“ einen Weg zur Modellierung von Anwendungen vor [2].

Ein wesentlicher Gedanke von Domain Driven Design ist, dass die zu entwickelnde Software die gesprochene Sprache der Kunden-Domäne möglichst direkt abbildet. Dazu ist es notwendig, im Zusammenspiel mit den jeweiligen Experten eine einheitliche Sprache zu ermitteln, die

sogenannte „Ubiquitous Language“ – Ziel ist eine formalisierte Sprache. Allerdings wäre es ein Irrweg, eine universelle Sprache für alle Projekte zu suchen; spezielle Sprachen für die jeweiligen Anwendungsfälle sind wesentlich exakter.

Neben der Ubiquitous Language gehören auch die Entitäten zu den zentralen Elementen des Domain Driven Design. Sie zeichnen sich dadurch aus, eindeutig identifizierbar zu sein und über eine Historie zu verfügen, etwa wie beim Kunden einer Bücherei. Zudem sind Entitäten die hauptsächlichen Träger der Geschäftslogik, die innerhalb der Domäne abgebildet werden.

Oft ist es notwendig, zusätzlich zu den eigentlichen Entitäten einer Domäne noch weitere Dinge zu modellieren. Diese Objekte werden im Domain Driven Design als „Value Objects“ bezeichnet. Es sind in der Regel Attribute, die eine Entität genauer beschreiben. Für den Kunden einer Bücherei könnte es sich dabei um die Anzahl der ausgeliehenen Bücher oder um das Geburtsdatum handeln.

Ein Value Object ist oft in einem oder mehreren primitiven Datentypen gekapselt, sodass darauf gegebenenfalls weitere Geschäftslogik implementiert werden kann. *Abbildung 2* zeigt die möglichen Domänen einer Bücherei sowie deren Entities. Es ist zu erkennen, dass zum Beispiel die Entity „Ausleiher“ sowohl in der Domäne „Registrierung“ als auch in „Ausleihe“ vorkommt. Die Ausprägung der En-

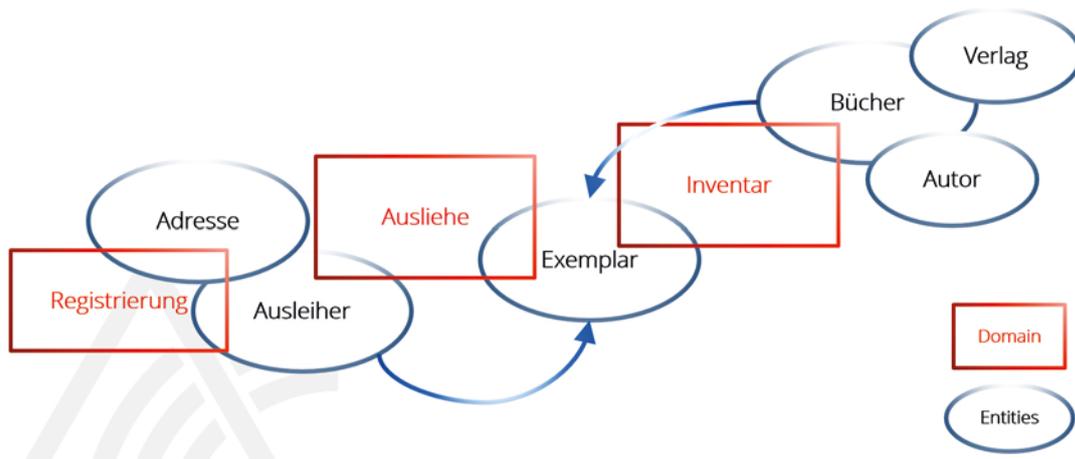


Abbildung 2: Datenmodell einer Bücherei nach DDD

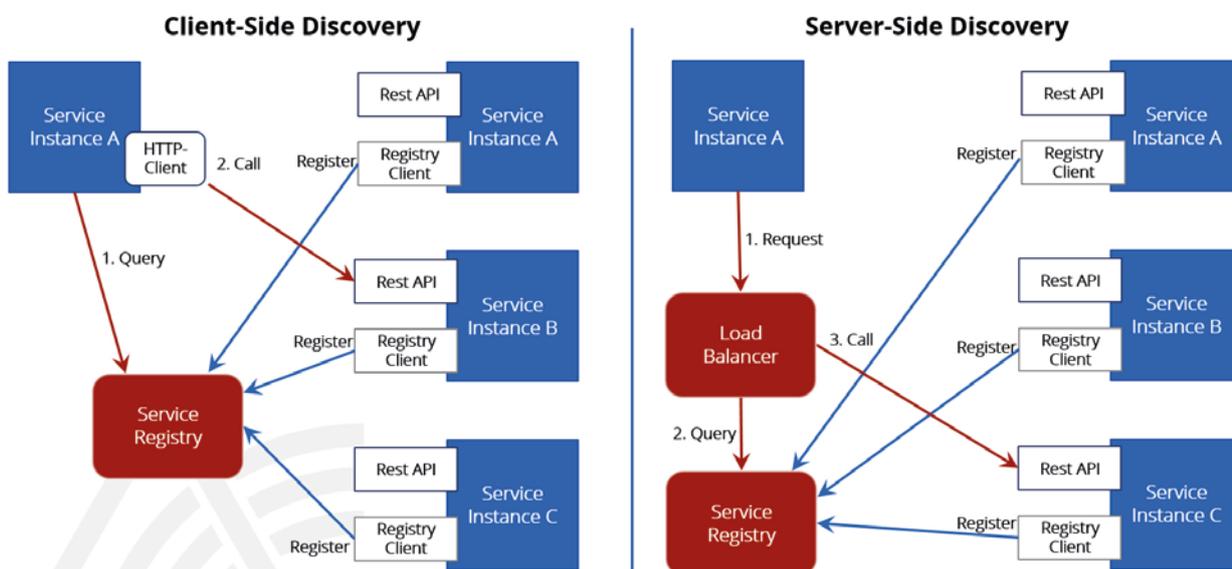


Abbildung 3: Gegenüberstellung Client-Side vs. Server-Side Discovery

tität ist allerdings unterschiedlich: So werden in der Domäne „Registrierung“ alle Informationen über den Ausleiher benötigt, in der Domäne „Ausleihe“ hingegen nur die Kennung des Ausleihers.

Service Discovery

In einer Cloud-Umgebung sind den Microservices oder SCS die Netzwerk-Adressen dynamisch zugewiesen. Im Gegensatz zu einer traditionellen Anwendung sind also den Aufrufern die Netzwerk-Adressen der zu benutzenden Services nicht mehr bekannt. Das hat zur Folge, dass zum Aufruf eines weiteren Service dessen Adresse zur Laufzeit neu ermittelt werden muss. Dies erfolgt mithilfe einer sogenannten „Ser-

vice Discovery“, die über folgende zwei Patterns implementiert werden kann:

- **Client-Side Discovery**
Hier ist der Client für die Ermittlung der Adresse der aufzurufenden Services verantwortlich. Dazu stellt er eine Anfrage an die Service Discovery, um die Adressen der verfügbaren Instanzen der Services zu erhalten. In der Regel erhält er auch noch Informationen über die Auslastung der verfügbaren Services und kann somit die Anfragen an die verfügbaren Services verteilen.
- **Server-Side Discovery**
Hier stellt der Client die Anfrage direkt an einen Load-Balancer. Dieser ermittelt mithilfe der Service Discovery die

Adressen der verfügbaren Instanzen und leitet die Anfrage an den Service weiter.

Abbildung 3 zeigt die Unterschiede zwischen den beiden Patterns.

Resiliente Services

Eine resiliente native Cloud-Anwendung kann Fehler mit minimalen Funktionseinbußen für den Benutzer abfedern und bietet Mechanismen, sich selbst zu heilen. In einer Cloud-Anwendung, die aus vielen separat entwickelten, eingerichteten und skalierten Services besteht, ist Resilienz also eine wichtige Eigenschaft. Daher sollten die Resilienz-Prinzipien

Protokoll-Abhängigkeit	Statt Remote-Procedure-Calls besser ein Datenprotokoll nutzen, das eine kompatible Weiterentwicklung ermöglicht (wie XML oder JSON) und das – statt einer „1:1“-Kommunikation zwischen Services, die bei vielen Services exponentiell wächst – durch Message Queues oder Topics die direkte Abhängigkeit umgeht.
Zeitliche Abhängigkeit	Statt blockierender synchroner Aufrufe ist eine asynchrone Kommunikation zu bevorzugen. Kurze Endpunkt-Ausfälle lassen sich durch Retry abfedern und synchrone Aufrufe mit Timeout begrenzen.
Instanz-Abhängigkeit	Statt spezifische Instanzen zu referenzieren, werden Endpunkte angesprochen, die den Request dynamisch an konkrete Service-Instanzen weiterleiten.
Daten-Abhängigkeit	Durch Aufweichen der Anforderungen an die Daten-Konsistenz zwischen Services ist es möglich, Puffer wie Caches oder Spiegelbestände aufzubauen.

Tabelle 1

„Isolation“ und „Self Healing“ bei der Migration berücksichtigt werden.

Ohne Isolation der einzelnen Services führen Fehler in einem verteilten System oft zu kaskadierenden Effekten. In diesem Fall muss in die Verfügbarkeit eines Service die Verfügbarkeit aller von ihm genutzten Services mit eingerechnet werden. *Tabelle 1* zeigt typische Formen der Abhängigkeiten zwischen Services sowie Ansätze, um diese Abhängigkeiten zu reduzieren.

Darüber hinaus sollten Anwendungs-Ressourcen immer explizit limitiert sein. Unlimitiert neigen sie dazu, diese in Ausnahmesituationen über die üblichen Normwerte hinweg zu verbrauchen. So schlägt sich dies, sofern die Zugriffswege nicht nach Services getrennt sind, schnell auf andere Services nieder – Requests in Thread Pools beim Aufrufer stauen sich, wenn ein aufgerufener Service die Anfragemenge nicht bedienen kann, und stehen bei fehlender Aufteilung des Pools auch für andere Services nicht mehr zur Verfügung. Gleiches passiert, wenn Aufrufe in einem solchen Fall immer länger brauchen und keine Request-Timeouts definiert wurden. Daher empfiehlt es sich in bestimmten Fällen, Ressourcen-Limits zu definieren, zum Beispiel bei Thread Pools und Connection Pools, Queue-Längen, Request Timeouts und Connection Timeouts, maximaler CPU, Speicherverbrauch und Datensatzgrößen.

Wenn ein Service dennoch nicht verfügbar sein sollte, gilt der Grundsatz: „Ein reduzierter Funktionsumfang ist besser als keine Funktion.“ Daher sollte versucht werden, die Auswirkung eines Service-Ausfalls auf ein Mindestmaß zu begrenzen. Statt einen Fehler bis zum Benutzer durchzuleiten, ist zu prüfen, ob an geeigneter Stelle ein Fallback-Verhalten oder eine alternative Funktionalität genutzt werden kann. Ist eine Web-Komponente

nicht funktionsfähig, kann diese auf der Website ausgeblendet werden. Ist eine Datenabfrage nicht möglich, können die Daten aus einer lokalen Kopie geliefert werden.

Das Circuit Breaker Pattern hilft, nach einem solchen Ausfall rechtzeitig wieder in den Normalbetrieb überzugehen. Das funktioniert folgendermaßen: Wenn ein aufgerufener Service mehrfach Fehler liefert, wird temporär auf eine Fallback-Funktion umgeleitet, damit der Service nicht weiter belastet wird. Durch regelmäßige Aufrufversuche gegen den Service wird sofort erkannt, wenn der Service wieder erreichbar ist (*siehe Abbildung 4*).

Self Healing

Cloud-Provider gewährleisten zwar die Verfügbarkeit der Cloud-Infrastruktur, jedoch nicht der Services einer nativen Cloud-Anwendung. Sogenannte „selbstheilende Systeme“ sind in der Lage, den Ausfall einzelner Microservices beziehungsweise SCS zu erkennen und selbstständig in einen Normalbetrieb zurückzukehren. Bei einer für Microservices-Architekturen typischen großen Zahl an Services ist es kaum machbar, durch manuelles Monitoring und manu-

elle Eingriffe die Verfügbarkeit der jeweiligen Services zu gewährleisten. Daher ist Self Healing wichtig für die Resilienz einer nativen Cloud-Anwendung.

Im ersten Schritt gilt es zu erkennen, ob Handlungsbedarf besteht. Dazu sind für den Service Metriken zu allen relevanten Ressourcenverbräuchen und Kennzahlen zu erfassen. Welche Ressourcen typischerweise zu Engpässen werden können, hängt sehr von der Anwendung ab und muss zunächst durch Last-Tests und im überwachten Regelbetrieb ermittelt werden. Stehen die Normwerte fest, sind Schwellwerte zu definieren und bei deren Unter- oder Überschreitung Alarme auszulösen. Diese können dann zu automatischen Operationen wie Einschalten eines Fallback oder Skalierung eines Service führen. Ein zustandsloses Service-Design vereinfacht das Auto-Scaling, da ausgefallene Instanzen oder zusätzliche Last leicht durch weitere Service-Instanzen abgedeckt werden können.

Um Self Healing zu ermöglichen, ist es außerdem notwendig, die Auswirkungen von Fehlern zu begrenzen, um so fehlerhaften Komponenten die Chance zu geben, in einen Normalzustand zurückzukehren. Das Bulkhead-Muster kann dabei helfen: Ressourcen wie Thread Pools oder Connection Pools werden

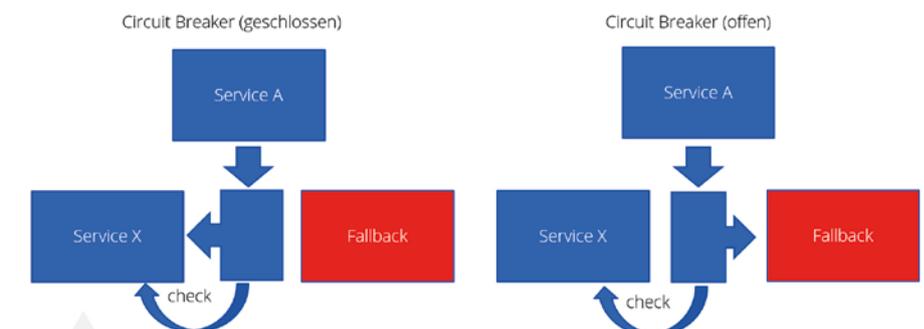


Abbildung 4: Circuit Breaker Pattern

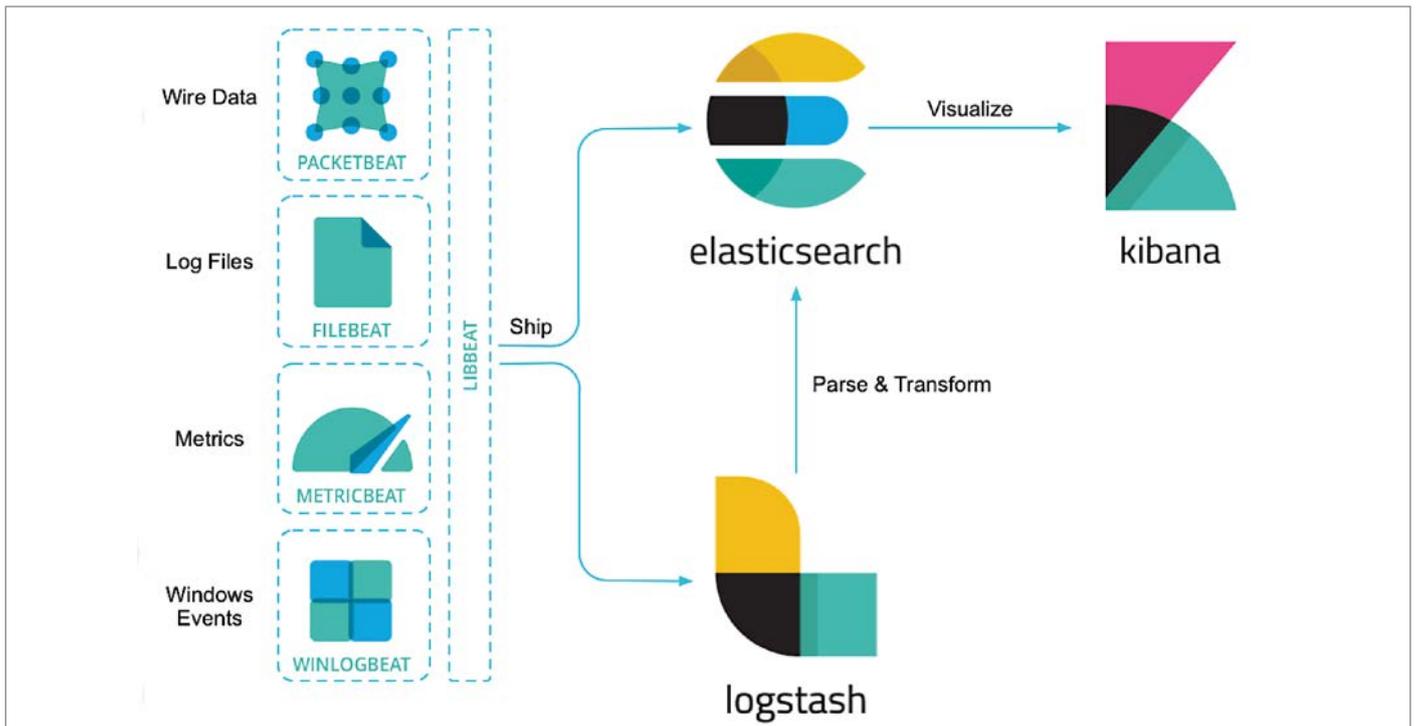


Abbildung 5: Zentrales Log-Management mit dem ELK-Stack

nach aufgerufenem Service partitioniert, sodass andere Services bei Ausfall nicht beeinträchtigt sind. Zudem kann eine Aufruf-Drosselung dafür sorgen, dass nicht mehr Anfragen an eine Schnittstelle gehen, als diese im aktuellen Zustand verkraften kann. Viele der genannten Designmuster wie Bulkhead oder Circuit Breaker sind von Libraries wie Polly (.Net) oder Hystrix (Java), das auch in Spring Cloud enthalten ist, unterstützt. Es empfiehlt sich ein Blick in die Dokumentation.

Logging

Beim Einsatz von Microservices-Architekturen beziehungsweise SCS erfolgt die Ausführung nicht mehr in realen Servern, sondern in virtuellen Containern. Somit verändert sich auch die Lebensdauer der Umgebungen. Davon sind auch die Logs der Anwendung betroffen, die im Extremfall bei einem Wegfall der Umgebung mit verloren gehen. Da die Logs notwendig sind, um den Betrieb und die Entwicklung bei der Analyse von Fehlern zu unterstützen, sind diese durch ein zentrales Log-Management zu sammeln.

Neben der eigentlichen Sammlung von Logs kommt dem zentralen Log-Management eine weitere Bedeutung zu:

Der Einsatz einer zentralen Event-Processing-Engine bietet neue Möglichkeiten für die Auswertung der Logs. So können beispielsweise durch die Korrelation von Firewall und Logs Indizien erkennbar sein, die auf einen Angriff schließen lassen.

Beim Aufbau eines zentralen Log-Managements sind einige Herausforderungen zu meistern:

- **Unterschiedliche Formate**
Log-Daten sind semi-strukturiert. Das Format ist von Applikation zu Applikation unterschiedlich.
- **Verteilte Daten**
Log-Daten sind über unterschiedliche Konten sowie Schichten (Frontend, Backend und Middleware) verteilt und auf Grund von Lastverteilung und Ausfallsicherheit mehrfach vorhanden.
- **Kontinuierliche Änderung**
Log-Datenquellen ändern sich laufend. Gerade im Bereich „Cloud-Infrastrukturen“ variiert die Anzahl der Knoten.
- **Sensitive Informationen**
Log-Daten können sensitive Daten enthalten. Insbesondere wenn ein größerer Kreis von Benutzern Zugriff auf Log-Daten bekommt, können sensitive Informationen gefiltert werden.

Log-Management-Lösungen sind auf die Bewältigung der genannten Herausforderungen ausgerichtet und helfen damit, diese zu bewältigen.

Elastic Stack

Der Elastic Stack (ELK) ist ein Open-Source-Stack, der sich aus mehreren abgestimmten Komponenten zusammensetzt und somit den Aufbau eines zentralen Log-Managements unterstützt. *Abbildung 5* zeigt dessen Architektur mit dem ELK-Stack. Die wesentlichen Komponenten sind:

- **Beats**
Beats [3] ist eine Plattform, die es ermöglicht, leichtgewichtige Daten-Shipper zu erstellen, die für spezifische Anwendungsfälle optimiert sind. Elastic stellt Beats-Shipper für Log-Dateien, System-Metriken, Windows-Events sowie Netzwerk-Events bereit und Beats ein API, um zusätzlich kundenspezifische Anwendungsfälle zu implementieren.
- **Logstash**
Logstash ist eine Event-Pipeline, die in der Lage ist, unterschiedlichste Datenquellen wie Log-Dateien zu konsumieren und aufzuarbeiten. Sie extrahiert infor-

mationsrelevante Felder und reicht sie an den Suchserver Elasticsearch weiter.

• *ElasticSearch*

ElasticSearch ist ein sehr mächtiger Volltext-Suchserver, der von namhaften Internetgrößen wie Xing oder Wikipedia für die Volltextsuche eingesetzt wird.

• *Kibana*

Kibana ist eine webbasierte Anwendung. Sie ermöglicht den Zugriff auf die indizierten Daten von Elasticsearch. Eine Vielzahl von Visualisierungskomponenten lässt sich zu individuellen Dashboards zusammenstellen.

Alternative Lösungen

Neben dem ELK-Stack wären als weitere bekannte Lösungen die Open-Source-Software Graylog [4] und die kommerzielle Software Splunk [5] zu nennen. Splunk bietet als kommerzielle Lösung viele Angebote, die über das rein zentrale Log-Management hinausgehen. Diese sind spezifisch auf unterschiedliche Bedürfnisse und Stakeholder zugeschnitten.

Fazit

Bei der Entwicklung von nativen Cloud-Anwendungen sind neben der eigentlichen Geschäftslogik noch eine Reihe weiterer architektureller Aspekte zu berücksichtigen. Wer alle Aspekte bedenkt, ist in der Lage, mit kurzen Innovationszyklen innovative Anwendungen zu entwickeln, die das Potenzial besitzen, die Wettbewerbsfähigkeit eines Unternehmens signifikant zu erhöhen. Neben den in diesem Artikel angesprochenen Architektur-Fragmenten sind weitere Aspekte wie automatisiertes Testen sowie Build und Deployment zu berücksichtigen, um den Paradigmenwechsel entsprechend zu unterstützen.



Kai Weingärtner
kai.weingaertner@opitz-consulting.com

Weitere Informationen

- [1] <https://dzone.com/articles/cloud-native-application>
- [2] Eric J. Evans: Domain Driven Design: Tackling Complexity in the Heart of Software. Boston, 2003
- [3] <https://www.elastic.co/guide/en/beats/libbeat/6.2/beats-reference.html>
- [4] <https://www.graylog.org/overview>
- [5] https://www.splunk.com/en_us/solutions/solution-areas/log-management.html



Stefan Kühnlein
stefan.kuehnlein@opitz-consulting.com

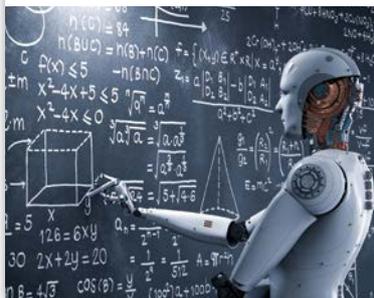
20. - 21. September 2018 in Dresden

DOAG BIG DATA Days

Daten, der Treibstoff der digitalen Gesellschaft

Unter dem Motto „**Daten, der Treibstoff der digitalen Gesellschaft**“ erhalten Sie die neuesten Informationen rund um **Big Data, Reporting/Visualisierung und Geodaten** sowie weitere Trends und Neuigkeiten. Seien Sie dabei, **der Frühbucher-Rabatt ist noch bis 10. August** verfügbar.

BIG DATA



VISUALISIERUNG/
REPORTING

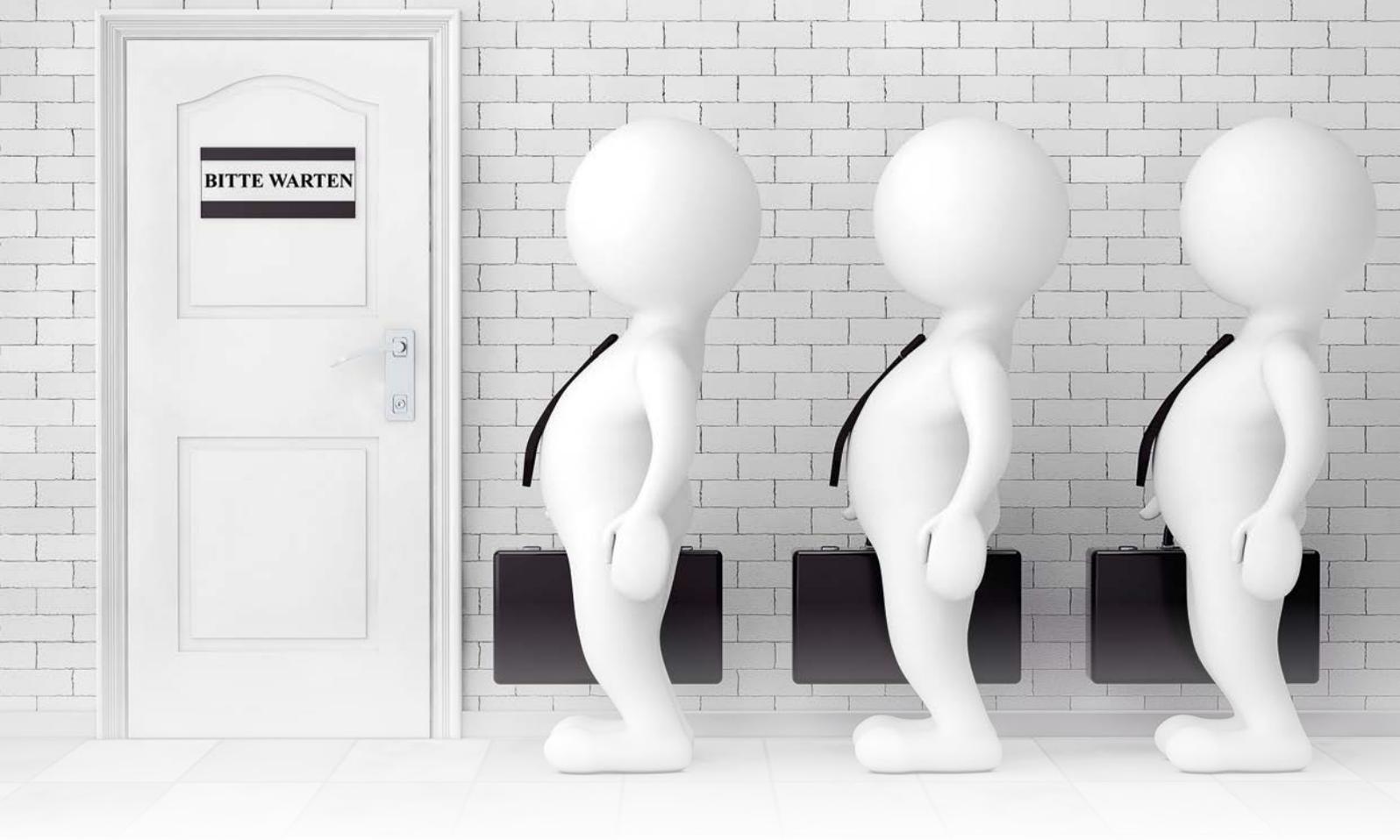


GEODATEN



Informationen und Anmeldung unter:
www.doag.org/go/bigdatadays





Leute hassen es zu warten

Jan-Peter Timmermann, Leiter der DOAG Infrastruktur und Middleware Community

In der heutigen Zeit erwarten Leute immer schnellere Antworten. Im Hinblick auf eine Kundenbindung kann dies für viele Unternehmen zu einem Problem führen. Jede Firma stellt mittlerweile eigene Webseiten zur Verfügung, aber ist nicht Google das Maß aller Dinge? Können Kunden mit der Webseite genauso umgehen, wie sie es mit Google können? Unternehmen mit einer modernen IT werden diese Probleme sicherlich nicht haben und können so Kunden an sich binden.

Im Jahr 2015 schätzte PayPal, dass für jede Sekunde Wartezeit zwölf Kunden ihren Online-Kauf abbrechen. Dieser Ungeduldsfaktor nimmt weiter zu. Eine im Jahr 2016 durchgeführte Umfrage unter 12.500 Facebook-Nutzern aus vierzehn Ländern kam zu dem Ergebnis, dass 53 Prozent ihrer Kunden eher mit einer Organisation Geschäfte machen, mit der sie Direkt-Nachrichten austauschen können.

Wenn der Computer eines Unternehmens im nächsten Jahr nicht in der Lage sein wird, direkt und schnell mit den Kunden über ihre grundlegenden Fragen zu

kommunizieren, dann wird dieses Unternehmen wirklich den Zug verpasst haben. Eine große Welle kommt auf die Firmen zu: Die Nutzer verlangen zunehmend eine Schnittstelle für die direkte Kommunikation. Wird die Organisation in der Lage sein, diese Welle zu reiten? Das ist ein schwieriges Problem für Unternehmen mit Legacy-Plattformen bei der Business-Logic wie Oracle Forms oder Ähnlichem, Oracle E-Business Suite (EBS), JD Edwards oder Primavera. Es gibt allerdings Wege, das zu überwinden und wesentlich schneller und besser zu kommunizieren.

Bis vor Kurzem verhinderten technologische Beschränkungen die schnellen Antworten auf grundlegende Informationen, die Nutzer von Unternehmen brauchen und wollen. Die moderne Technologie von Oracle hat einen großen Teil dieser Arbeit durch Nutzung von maschinellem Lernen und Natural Language Processing (Verarbeitung natürlicher Sprache) auf Chatbots verlagert. Den Unternehmen stehen somit jetzt riesige Chancen für die Umwandlung von Geschäftsanwendungen in Konversations-Schnittstellen zur Verfügung.

Chatbots sorgen für Geschwindigkeit und Effizienz

Sprache ist das ursprüngliche menschliche Interface; heute sprechen die Menschen mit Alexa, Cortana, Google Assistant und Siri, weil diese KI-Bots immer kompetenter und hilfreicher werden. Dabei ist zu beachten, dass die Menschen mit diesen Computer-Bots und nicht zu ihnen sprechen. Es findet also eine Konversation statt. Generationen haben sich an Computer gewöhnt; jetzt sind diese so leistungsfähig und die Software gut genug, um mit Menschen eine Unterhaltung zu führen.

Messaging-Tools wie die von Facebook, Google, Microsoft und Slack erzielen erstaunliche Dinge mit der Rechenleistung, über die ihre Nutzer bereits verfügen. Nutzer betrachten sie als bevorzugtes Mittel für andere Transaktionen. Die Zukunft des Business hängt davon ab, diese Tools für die Nutzerbindung zu verwenden – und viele weitere hinzuzufügen.

Oracle hat eine umfangreiche Plattform geschaffen, um Chat-Unterhaltungen zu ermöglichen. Auf der OpenWorld 2016 angekündigt, bietet der Oracle Intelligent Bots Cloud Service (IBCS) eine Konversations-Schnittstelle zu den wichtigsten und bewährtesten Chat-Anwendungen. Verbindung und Integration sind die Schlüssel zum Erschließen des Geschäftspotenzials von Chats. Während Chatbots in ihrer eigenen – nicht verbundenen – Umgebung existieren können, erschließen sie wesentlich größeren Mehrwert, wenn sie mit Organisationsdaten und Business-Logik verbunden sind.

Mobile Geräte sind die besten Chat-Geräte und das nicht nur, weil die Benutzer mit ihrer Bedienung vertraut sind. Mobile Geräte verfügen über Informationen, die das Unternehmen gewöhnlich verarbeiten muss, wie authentifizierte Identität, Datum und Uhrzeit, Kalender-Informationen und vieles mehr. Fügt man diese Informationen dem hinzu, was der Chatbot lernt, erreichen die Geschäftsvorgänge eine höhere Genauigkeit und personalisierte Antworten (siehe Abbildung 1).

Wenn beispielsweise „Buche mir einen Flug nach SFO“ zu einer qualifizierten und informierten Anwendung gesagt wird, kann ein Chatbot hinzufügen, was er über Standort- und Kalenderereignisse auf dem Mobilgerät weiß, was er über Sitzplatz-

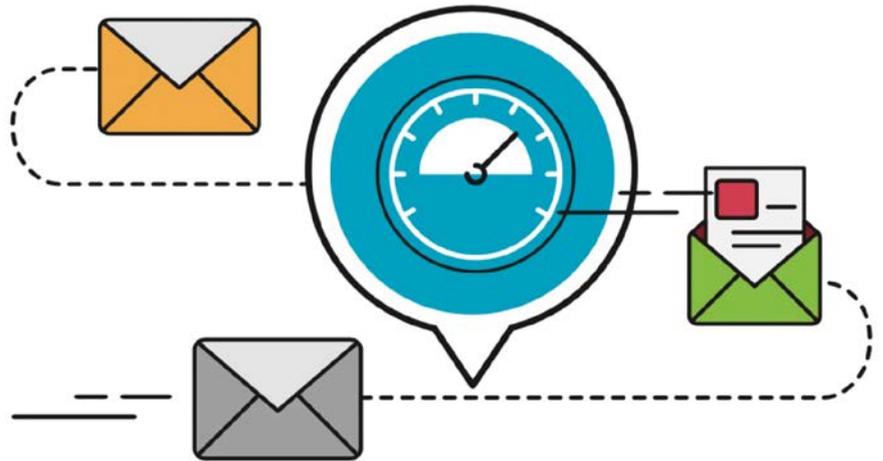


Abbildung 1: Der Chatbot lernt hinzu

und Fluggesellschaftspräferenz gelernt hat, um intelligente Ergebnisse zu präsentieren, die am wahrscheinlichsten zu einer Kaufentscheidung führen. Er kann sogar personalisierte Preisgestaltung einsetzen, basierend auf dem, was er über Reisegewohnheiten weiß oder gelernt hat.

Die Informatik hinter der Chatbot-Technologie ist faszinierend. Oracle hat den größten Teil der Arbeit schon erledigt. Sein Framework stellt viel von dem zur Verfügung, was Kunden benötigen, um eine solide Chat-Anwendung zu erstellen. Oracle IBCS bietet den Kunden eine grafische Benutzeroberfläche, um vier Aspekte des Prozesses zu definieren. Sobald das Schema erstellt ist, generiert Oracle die AI, um mit den APIs der meisten Chat-Anwendungen zu interagieren, und kann den Anwendungsfall sowohl auf Sprache als auch auf Tastatur-Messaging erweitern.

Das IBCS-Dashboard listet die aktuellen Bots auf. Jeder Bot besteht aus vier Segmenten: Intents, Flows, Entities und Components. Diese Segmente bilden einen Chatbot, der in einem Nachrichtenkanal wie Facebook Messenger oder Slack aktiviert ist. Um zu veranschaulichen, wie IBCS funktioniert, ein einfaches Beispiel wie die Abfrage des Kontostands. Ein typischer Austausch kann mit „Wie viel ist noch auf meinem Girokonto?“ beginnen. Der Chatbot muss die Intention dieser Anfrage entschlüsseln und das System in der Lage sein, dies in eine logische Aussage zu übersetzen. Wenn das nicht möglich ist, muss es nach weiteren Informationen fragen, etwa „Welches Girokonto?“.

Sobald die Intention bestimmt ist, muss der Chatbot die richtige interne Syntax bilden, um eine Aktion zu erzeugen,

dann diese Anweisung an die Unternehmensanwendung übergeben, auf Daten von diesem System warten, sie empfangen und die entsprechende Antwort bilden. Er muss sowohl die menschliche Sprache als auch die Computersprache gut genug interpretieren, um die Interessen beider Parteien genau darzustellen.

Intents

Vor einer Absicht oder einer Intention stehen Äußerungen. Beim Entwickeln des Chatbot muss der Entwickler an die vielen Möglichkeiten denken, wie die Absicht ausgedrückt werden kann. Das bedeutet, sich genau zu überlegen, wie die Endnutzer die Fragen formulieren werden, um sie dann bestimmten Themen zuzuordnen, auf die der Bot reagieren kann. Im Laufe der Zeit muss der Chatbot lernen, Absichten aus Äußerungen zu formen.

Maschinelles Lernen und die Verarbeitung natürlicher Sprache (Natural Language Processing, NLP) bieten die beste Möglichkeit, eine lebensnahe Konversation zu erreichen. Mit dieser Technologie wird eine Liste von Äußerungen, die zu Absichten werden, gegen eine Reihe von Absichten bewertet.

Zum Beispiel die Aufforderung „Sag mir, wie viel Geld ich auf meinem Girokonto habe“ kann eine 90-prozentige Korrelation mit der Absicht „Wie ist der Kontostand auf meinem Girokonto?“ haben. Die Chatbot-Engine stellt die Berechnung und den Confidence-Faktor (wie wahrscheinlich ist dies die gewünschte Information?) der richtigen Antwort zur Verfügung (siehe Abbildung 2).

Diese NLP- und Machine-Learning-Rechenleistung und Intelligenz sind Kern-

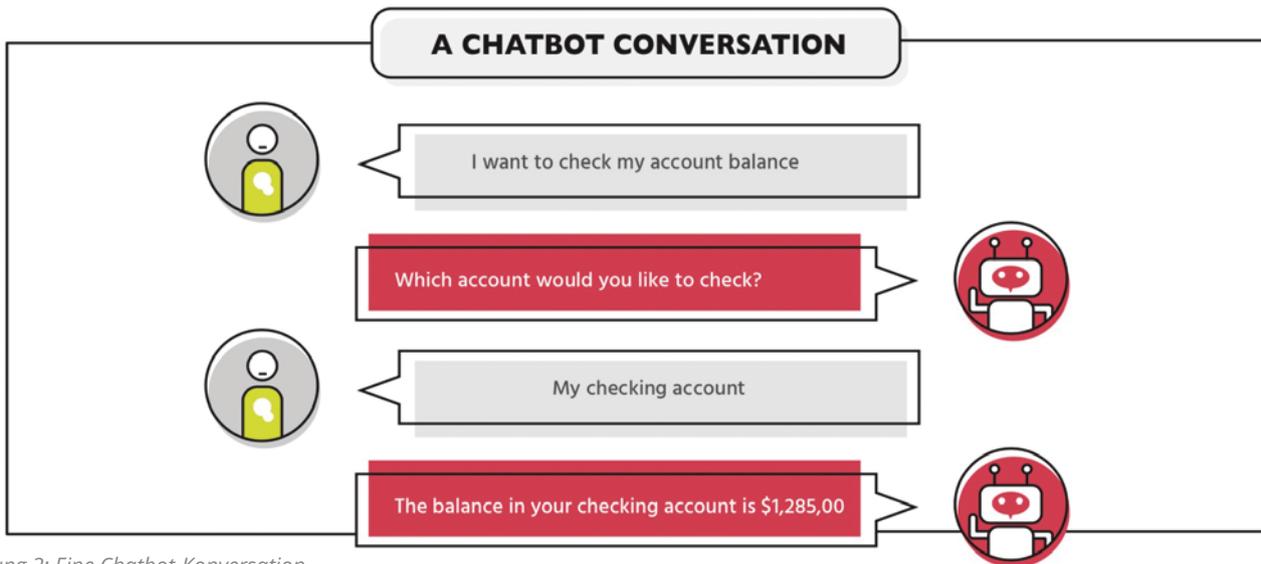


Abbildung 2: Eine Chatbot-Konversation

technologien von Oracle IBCS; die zugrunde liegenden Algorithmen sind eingebaut. Sie repräsentieren Worte und Ketten von Wörtern als Vektoren. Weil sie kein Sprachverständnis haben, sind diese Algorithmen vollkommen sprachunabhängig.

Entities, Flows und Components

Vorkonfigurierte Entities dienen als Parameter für eine Absicht. Es sind die Felder und Datenpunkte, die aus den Absichten extrahiert werden und die man seinen REST-Diensten zuordnen muss. Im Fall des Beispiels kann eine Entity ein Bankkonto oder eine Kreditkarte enthalten. Werte für Entities werden als Parameter verwendet, wenn Daten angefordert und Prozesse ausgeführt werden. Oracle IBCS wird mit integrierten Standardsets geliefert, die Datum, Uhrzeit, Währung und dergleichen enthalten. Nutzer können zusätzliche Sets erstellen und auch Listen konfigurieren, die das System während der Chatbot-Funktion zur Validierung oder als Vorschlag verwendet.

Flows sind der Teil, in dem ein großer Teil der Magie stattfindet. Damit der Chatbot funktioniert, muss er einem Pfad folgen können, einem vorgeplanten Konversationsfluss, dem Prozess der normalisierten oder standardmäßigen Datenbildung. Der Flow kann zu mehr Fragen führen, deren Antworten ein Set von Attributen erhöhen, die dann Entities zugeordnet werden. Zum Beispiel erfordert die Äußerung „Wann fährt der nächste Zug?“ die

„von“- und „bis“-Stations-Attribute, bevor der API-Call durchgeführt werden kann.

Viel Mehrwert kann von gut durchdachten Flows ausgehen. Da Oracle-IBCS Datenwerte im lokalen Cache speichert, werden die auf dem Weg gesammelten Informationen zur Personalisierung der Beziehung verwendet. Der Chatbot muss nicht mehrmals nach dem Benutzernamen fragen. Nutzerdefinierte Oracle-IBCS-Komponenten sind die Elemente, die mit Informationsspeichern und Verarbeitungs-Engines verbunden sind. Der API-Speicher stellt den Zugriff auf Daten und Prozesse unter Verwendung der standardmäßigen „GET“- und „POST“-Operatoren in einer Organisation und sogar außerhalb davon sicher. Es gibt nur ein Problem mit den Oracle-IBCS-Komponenten: Sie können nicht direkt mit Oracle Forms, Oracle E-Business Suite (EBS), JD Edwards oder Primavera kommunizieren.

Die meisten Unternehmensanwendungen sind stark individualisiert. Sie machen die Arbeit gut. Ihr einziges Problem (wenn auch ein großes Problem) ist ihre Front-End-Schnittstelle. AuraPlayer ist eine einzigartige, zum Patent angemeldete Lösung, um die Lücke zwischen Oracle-Back-Ends wie Oracle Forms, E-Business Suite, JD Edwards, Primavera und Front-Ends der nächsten Generation zu schließen. Es ermöglicht die Freischaltung der Geschäftslogik, die in diesen Desktop-gebundenen Anwendungen enthalten ist, ohne Kosten oder Risiken für eine Neuentwicklung.

AuraPlayer ist die einzige Lösung auf dem Markt, die ältere Oracle-Forms-Anwendungen zukunftssicher machen kann, indem sie diese zu Chatbots oder mo-

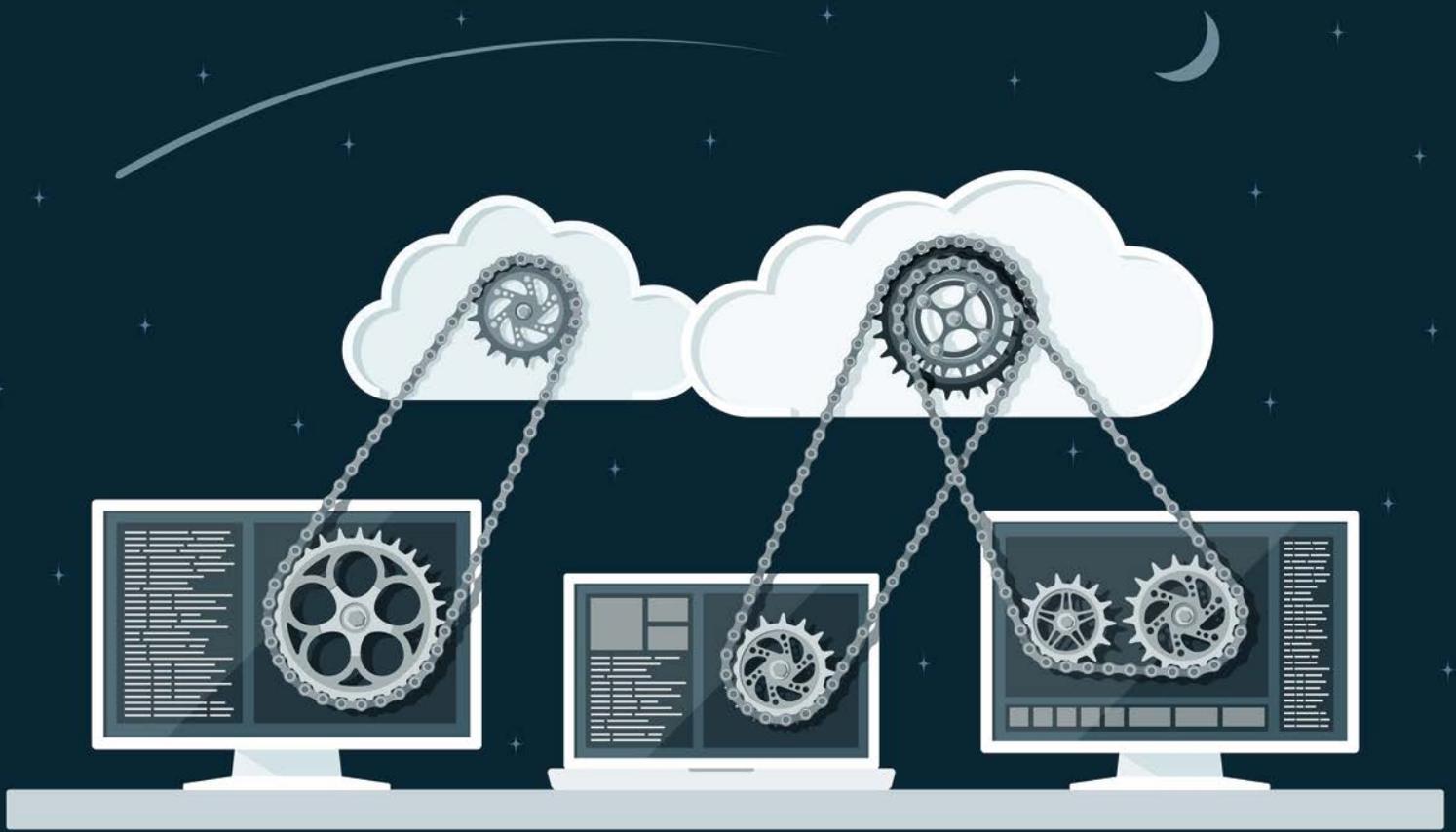
bilien Anwendungen erweitert. AuraPlayer rührt keine einzige Codezeile an und ist eine nahtlose Lösung, die Geschäftsanwendungen als Webservices verfügbar macht, indem Nutzeraktionen aufgezeichnet werden, um einen SOAP- oder REST-Webdienst zu erstellen, der als Backbone für jede Konversationsanwendung dienen kann.

Auf der grundlegendsten Ebene erfolgt die Aktivierung von Oracle-Back-Ends über Webservices. Sobald die aufgezeichneten Geschäftsprozesse als Webservice verfügbar sind, werden sie in jedem modernen Front-End konsumiert. Dadurch werden die Geschäftsregeln für jeden Verbraucher des Prozesses freigegeben.

Derselbe Geschäftsprozess, der Oracle vor Ort durchläuft, findet jetzt durch die Dialog-Anwendung über Webservices statt. Nutzer führen die Forms-Komponente aus, ohne den Desktop-Bildschirm auszuführen. Im Oracle-Forms-Sprachgebrauch bedeutet dies, dass Nutzer dasselbe Formular auf dem Desktop wie von einer Mobil- oder Konversations-Umgebung heraus ausführen.



Jan-Peter Timmermann
jan.timmermann@jptu.de



Cloud meets On-Premises – Herausforderungen in hybriden Infrastrukturen

Danilo Schmiedel, OPITZ CONSULTING Deutschland GmbH

In der heutigen digitalen Welt entsteht eine Vielzahl neuer IT-Lösungen, die Cloud-Dienste mit vorhandenen lokalen Systemen kombinieren. Die Folge sind hybride Architekturen, in denen die konsistente End-to-End-Verwaltung, Überwachung und Protokollierung über die gesamte Anwendungslandschaft hinweg besondere Herausforderungen mit sich bringen. Zusätzliche Anforderungen bestehen hinsichtlich Datenschutz, System-Integration und Kosten-Management. Der Artikel erläutert verschiedene Lösungsstrategien und Best Practices.

Die Lösungen zur Digitalisierung verändern auch die klassische Infrastruktur – alles wird Software. Wir sprechen heute von „Software-Defined Network“ und „Software-Defined Infrastructure“. Die meisten Unternehmen nutzen dafür Cloud-Lösungen mit unterschiedlichen Liefer- und Service-Modellen. In Cloud-Umgebungen ist für die Bereitstellung neuer virtueller Maschinen nur noch ein einfacher API-Aufruf nötig; diese Maschinen werden also nicht mehr von Hand eingerichtet, sondern au-

tomatisch durch sogenannte „Infrastructure-as-Code-Lösungen“ (wie Oracle Cloud Stack Manager oder Terraform) provisioniert.

Die Evolution von Cloud und Software verstärkt sich gegenseitig. Anstatt auf diese Entwicklung mit Bestrebungen zur Zentralisierung und Standardisierung zu antworten, um somit in der Folge wieder die Geschwindigkeit zu hemmen, sollte die IT einen „Design for Change“-Ansatz verfolgen und diesen in eine reaktive

Infrastruktur integrieren. Hybride Infrastrukturen vereinen die Stärken von On-Premises-Ansätzen mit der Flexibilität des Cloud Computing.

Cloud meets On-Premises

Die Vorteile und das Versprechen des Cloud Computing liegen auf der Hand: IT-Power aus der Steckdose. Die Kosten sind nun reine Verbrauchskosten und be-

ziehen sich auf die tatsächliche Nutzung, basierend auf nutzungsbezogenen Preismodellen. Betriebswirtschaftlich ist somit eine Wandlung der Kapitalbindung durch IT-Infrastruktur (CAPEX) in operative Betriebskosten (OPEX) möglich. Neben den betriebswirtschaftlichen Überlegungen hinsichtlich Investitionsschutz und Abschreibungszeiträumen bedingen im Endeffekt zwei Kriterien die Implementierungsentscheidung für oder gegen eine Cloud-Lösung: Datenschutz und Kosten.

Datenschutz und Datensicherheit

Zunächst ist es wichtig, die Begriffe „Datenschutz“ und „Datensicherheit“ voneinander abzugrenzen. Je nach Betrachtungsweise wird Datenschutz verstanden als Schutz vor missbräuchlicher Datenverarbeitung, Schutz des Rechts auf informationelle Selbstbestimmung, Schutz des Persönlichkeitsrechts bei der Datenverarbeitung und auch Schutz der Privatsphäre. Das Bundesdatenschutzgesetz (BDSG) regelt die Erhebung, Verarbeitung und Nutzung personenbezogener Daten, die Auskunft zu sachlichen oder persönlichen Verhältnissen einer Person geben. Schutzziel ist die informationelle Selbstbestimmung; betroffene Personen haben also das Recht zu erfahren, wo und wie die Daten verarbeitet werden.

Die Datensicherheit bezieht sich auf den technischen Schutz von Daten jeglicher Art vor Verlust, Manipulationen und andere Bedrohungen. Hinreichende Datensicherheit ist eine Voraussetzung für effektiven Datenschutz. Als Schutzziele sind insbesondere Verfügbarkeit, Integrität und Vertraulichkeit zu nennen.

Für beide Aspekte ist es notwendig, die gewachsenen, oft schon veralteten Regelungen zu hinterfragen, um nicht ohne Not eine Entscheidung zu Ungunsten einer Cloud-Lösung zu treffen. Mit der neuen EU-Datenschutz-Grundverordnung (DS-GVO) und dem Bundesdatenschutzgesetz 2018 (BDSG-2018), die ab 25. Mai 2018 in Kraft treten, gewinnt diese Frage an Bedeutung. Das gesamte System wird durch die neuen Regeln zwar weitestgehend nach deutschen Standards, aber grundlegend neu gestaltet. Aufgrund der umfassenden technischen und organisatorischen Maßnahmen der Anbieter kann der Weg in die Cloud sogar zu einer Verbesserung des eigenen Datenschutzes und der Datensicherheit beitragen. Schließlich verfügen unternehmenseigene Rechenzentren häufig nicht über die strengen Qualitätsstandards und aktuellen Zertifizierungen auf dem Niveau der Cloud-Anbieter [1].

Nach §11 BDSG ist Cloud Computing eine Auftragsdatenverarbeitung (ADV). Es wird zwischen Nutzer (= Auftraggeber) und Auftragsdatenverarbeiter (= Auftrag-

nehmer beziehungsweise Cloud-Anbieter) unterschieden. Der Nutzer ist dabei die verantwortliche Stelle und trägt die Sorgfaltspflicht für eine ordnungsgemäße Datenverarbeitung (DV). Er muss den Zweck der DV definieren und Weisungen erteilen. Der Auftragsdatenverarbeiter handelt ausschließlich auf Weisung des Cloud-Nutzers. Kundendaten werden nur im Umfang des gewählten Cloud Service verarbeitet. Konkret müssen in der ADV die folgenden Punkte geregelt sein:

- Zweck der Auftragsverarbeitung
- Weisungen des Auftraggebers
- Beschreibung und Überprüfung konkreter technologischer und organisatorischer Sicherheitsmaßnahmen
- Vereinbarung zur Löschung der Daten
- Regelung zu Subunternehmern
- Kontrollrechte des Cloud-Nutzers und Informationspflichten des Cloud-Anbieters
- Wahrung der Betroffenenrechte
- Rückgabe nach Vertragsbeendigung

Für die Nutzung von Cloud Services sind konkret technisch-organisatorische Sicherheitsmaßnahmen zu ergreifen, die es dem Anbieter ermöglichen, seine Sorgfaltspflichten zu wahren und die Kontrolle über die Datenverarbeitung zu behalten. Dazu müssen insbesondere die folgenden Datenschutzziele eingehalten werden:

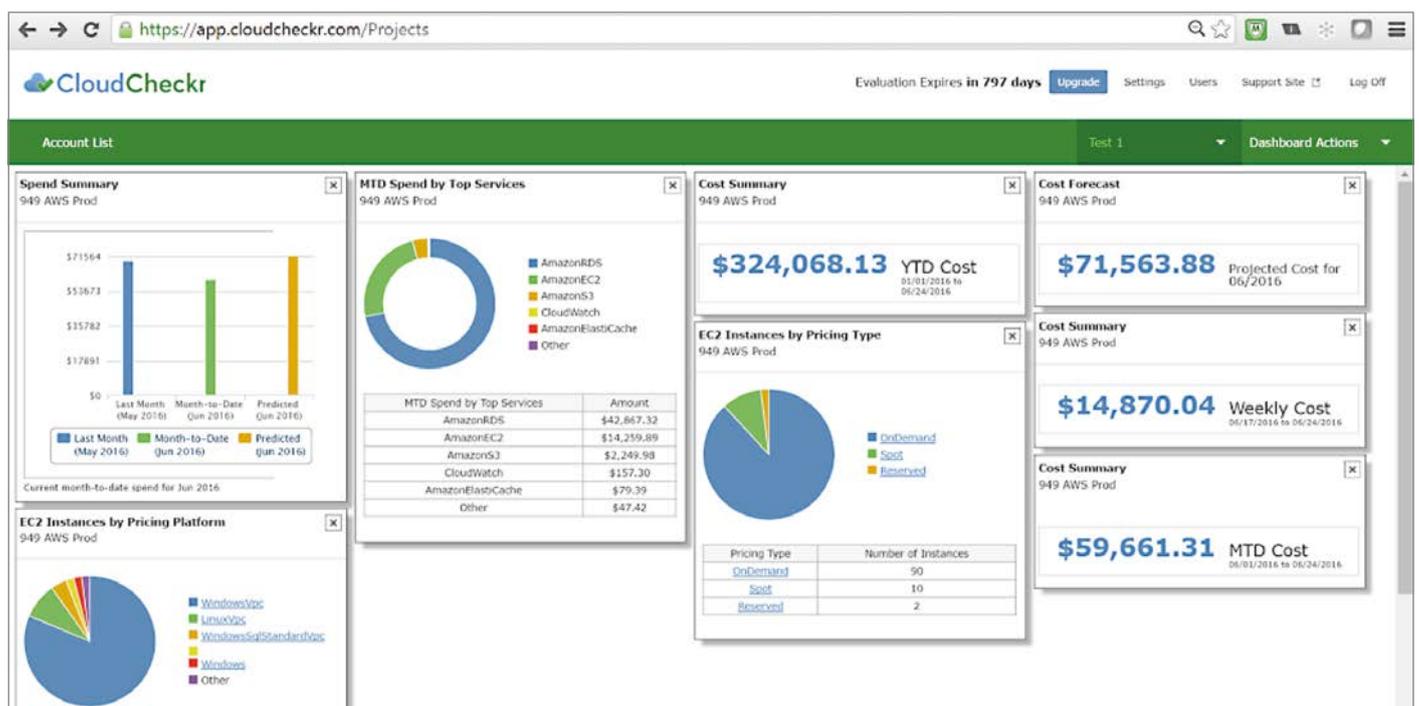


Abbildung 1: Multi-Cloud-Kosten-Management mit CloudCheckr

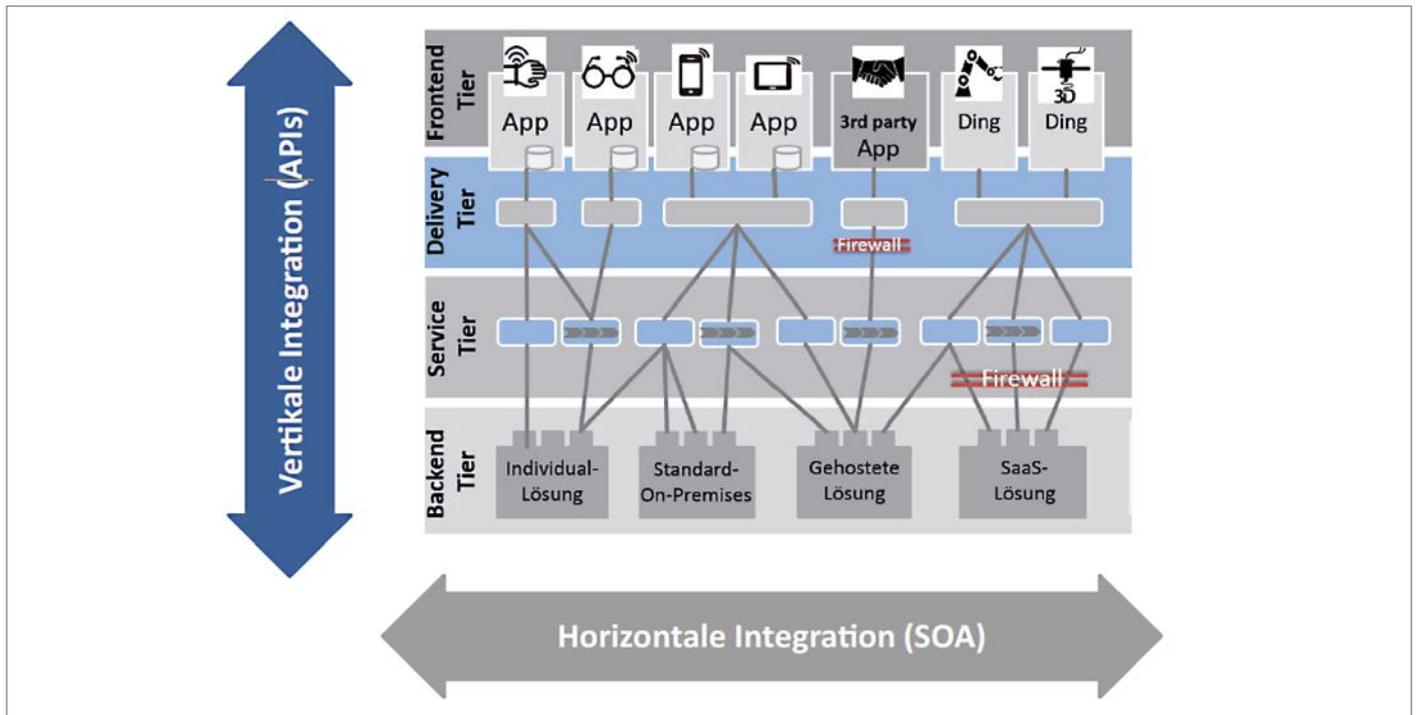


Abbildung 2: Horizontale und vertikale Integration

Zutrittskontrolle, Zugangskontrolle, Zugriffskontrolle, Weitergabekontrolle, Eingabekontrolle, geteilte Verarbeitung, Auftragskontrolle und Verfügbarkeitskontrolle.

Modell der geteilten Verantwortung

Wer nun glaubt, dass Daten in der Cloud zwangsläufig Kontrollverlust bedeuten, der irrt. Dieser Mythos hält sich hartnäckig. In Wahrheit ist es jedoch so, dass ein Cloud-Nutzer die Kontrolle über die Daten behält und auch für diese

verantwortlich ist (Shared Responsibility Model). Oracle unterteilt die Verantwortlichkeiten von Anbieter und Nutzer in folgende Bereiche: Identity and Access Management (IAM), Workload Security, Data Classification and Compliance, Host Infrastructure Security, Network Security, Client and End-Point Protection sowie Physical Security [2].

Zusammenfassend ist der Cloud-Anbieter verantwortlich für die Sicherheit der zugrunde liegenden Cloud-Infrastruktur. Diese besteht aus den Einrichtungen, der Hardware, der Software und dem Netzwerk. Demgegenüber ist der Cloud-Nutzer verantwortlich für die Sicherheit in der Cloud. Durch Verschlüsselung der Daten und Speicherung der Schlüssel „im

eigenen Haus“ hat der Cloud-Nutzer die alleinige Hoheit über die Daten. Er ist verantwortlich für Management und Überwachung des Zugriffs durch individuelle Benutzerkonten sowie der Aktivitäten in diesen Konten. Für die meisten Cloud Services ist das Risiko des Kontrollverlustes demnach vergleichbar mit dem Risiko bei herkömmlichen IT-Infrastruktur- oder IT-Outsourcing-Anbietern [1]. Zwingende Voraussetzung für den erfolgreichen Aufbau hybrider Cloud-Infrastrukturen ist die sorgfältige Betrachtung folgender Rahmenbedingungen:

- Klärung gesetzlicher Anforderungen
- Prüfen der vertraglichen Gegebenheiten zwischen Cloud-Anbieter und Nutzer
- Prüfung der technischen und organisatorischen Sicherheitsmaßnahmen des Cloud-Anbieters und Mapping zu Datenschutzzielen (unterteilt in Verantwortlichkeiten nach Cloud-Anbieter und Cloud-Nutzer)
- Ermittlung des Schutzbedarfs und Zuordnung zu den Schutzklassen
- Ermittlung der technisch-organisatorischen Sicherheitsmaßnahmen, die für die jeweiligen Schutzklassen zu ergreifen sind, beziehungsweise Klärung, ob Daten einer Schutzklasse überhaupt in der Cloud verarbeitet werden dürfen
- Risiko-Analyse

Kosten

Eine besondere Herausforderung in hybriden Infrastrukturen stellt das Kostenmanagement dar. „Pay as you go“, „OPEX“ statt „CAPEX“, Orchestrierung von Services in Minuten, Abfangen von Lastspitzen, automatische Skalierung und Datensicherheit sind zwar Vorzüge, die eine Public Cloud zu bieten hat, aber nur wer sie richtig nutzt, spart Geld, ist schneller, flexibler und damit innovativer. In der Public Cloud ersetzen Services und Subscriptions die alten Lizenzmodelle. Man bezahlt in der Public Cloud also nur für das, was man nutzt. Nicht mehr und nicht weniger. Auch hier gibt es jedoch Unterschiede, über welche Metriken die Abrechnung erfolgt:

- Minutengenaue Abrechnung
- Vergünstigungen beim Erwerb von Jahrespauschalen
- Anrechnung eigener Lizenzen (BYOL)
- Ersteigern von Rechenkapazität für kurzfristige Nutzung
- Abhängige Preisgestaltung vieler Anbieter je nach Durchsatzrate, Art des Speichers und Datenverkehr

Die Liste ließe sich noch beliebig fortführen. Hier zeigt sich die Komplexität der Preisgestaltung in der Public Cloud. Kostentransparenz ist daher ein entscheidender Faktor:

Welche Services können welcher Kostenstelle zugeordnet werden? Wie stehen die geplanten Kosten zu den tatsächlich angefallenen Kosten im Verhältnis? Wie lässt es sich vermeiden, dass ungenutzte Cloud-Ressourcen unnötig Kosten verursachen? Welche Auswirkung haben Auto-Scaling und Serverless-Ansätze auf die Kostenentwicklung?

Die Implementierung hin zu mehr Kostenkontrolle erfolgt in zwei Schritten. Zunächst sollte die Organisation Rollen und Verantwortlichkeiten für zentrale IT- und Finanzteams sowie Cloud-Ressourcenbesitzer in den Geschäftsbereichen definieren. Daraufhin ist die Etablierung einer kollaborativen Kostenmanagement-Plattform zur effizienten und kontinuierlichen Kontrolle und Optimierung der Kosten zu empfehlen. Cloud-Anbieter wie Oracle bieten eigene webbasierte Kosten-Explorer für das Monitoring ihrer Services an. Allerdings handelt es sich hierbei zumeist nur um eine Sicht auf die aktuellen „IST“-Kosten. Mit Cloud-Checkr [3], Cloudability [4] und RightScale [5] gibt es deutlich ausgereifere Multi-Cloud-Kostenmanagement-Lösungen am Markt. Zu ihren Kernfunktionalitäten gehören Dashboards zur Kosten-Analyse, Showback- und Chargeback-Berichte, Trigger zum Auslösen von automatisierten Aktionen, Kollaborationsmöglichkeiten, Alerting und zentralisierte Optimierungsempfehlungen (siehe Abbildung 1). Keine der genannten Lösungen bietet Stand heute eine Integration zur Oracle Cloud an. Zumindest bei [3] steht diese gemäß Herstellerangabe jedoch auf der Roadmap.

Interoperabilität und Integration

Der Trend von Cloud Computing und SaaS-Lösungen verschärft die Notwendigkeit hybrider Infrastruktur-Architekturen für die Applikationslandschaft. Dies erfordert eine modulare Architektur des Backends und die Entkopplung der Frontend-Komponenten. Die Release-Zyklen der einzelnen Komponenten dürfen nicht die Plattform als Ganzes kompromittieren. Hinsichtlich der Balance der On-Premises-Lösungen zu den Cloud-Lösungen spricht man vom „Center of Gravity“, womit der Anteil der Cloud-Lösungen in Bezug zu den On-Premises-Installationen ausgedrückt wird. Je höher der Einsatz unterschiedlicher Cloud-Lösungen ist, desto eher verschiebt sich das „Center of Gravity“ in die Cloud und somit auch die Integrations-Plattform, die näher an den Ursprung der Daten rückt [6].

Durch die gewachsene Dezentralisierung der Applikationen besteht die Gefahr von Wildwuchs und potenzieller Unbeherrschbarkeit. Deshalb empfiehlt sich eine Einteilung in horizontale und vertikale Integrationsaspekte (siehe Abbildung 2). Die horizontale Integration bezeichnet den klassischen System-zu-System-Ansatz. Ihm liegen oftmals schwergewichtige proprietäre Protokolle, technische Schnittstellen, Batch-basierter Datenaustausch und asynchrone Interaktionspatterns zugrunde. Klassischerweise kommen hierfür zentrale Integrationsplattformen (wie Oracle Integration

Cloud Service) mit standardisierten Adaptern zum Einsatz.

Bei der Einführung von zentralen Integrationslösungen in hybriden Cloud-Infrastrukturen ist besonderes Augenmerk auf das Design der Gesamt-Architektur zu legen, da der Datentransfer ein entscheidender Kostentreiber sein kann. Durch ungünstig gewählte Kommunikationsstrecken können doppelte Kosten beim Datenaustausch zwischen Systemen entstehen – etwa, wenn die gleichen Daten mehrfach über verschiedene Regionen und unterschiedliche Clouds transportiert werden. Dieses Problem verschärft sich speziell in Multi-Cloud-Landschaften. Kontinuierliche Kostentransparenz und eine durchdachte Integrations-Architektur sind hier die wesentlichen Erfolgsfaktoren.

Die vertikale Integration ist eher nutzergetrieben und basiert auf leichtgewichtigen Protokollen und fachlich ausgerichteten Schnittstellen, die nahezu in Echtzeit antworten müssen. In den letzten Jahren hat hier das für hybride Infrastrukturen relevante „API Management“ besonders an Bedeutung gewonnen [7]. Dabei steht die Bereitstellung von fachlich orientierten APIs im Fokus, die leicht verständlich und somit von einer breiten Masse nutzbar sein müssen. Dabei ist es sinnvoll, zwischen sogenannten „Single-“ und „Multi-Purpose“-APIs zu unterscheiden – so wie im Projekt „Open Modern Enterprise Software Architecture“ (OMESA) definiert (siehe Abbildung 3).

OMESA wurde mit dem Ziel ins Leben gerufen, architektonische Best Practices in

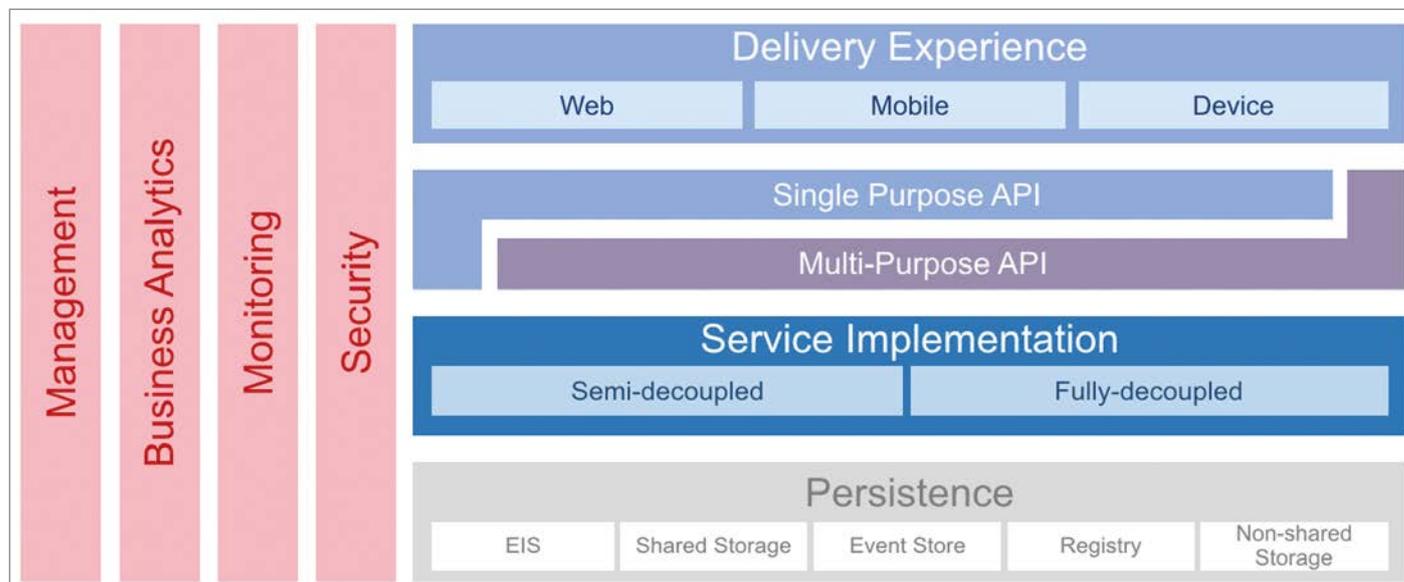


Abbildung 3: OMESA-Referenz-Architektur

moderne Architekturen zu integrieren und dabei zu berücksichtigen, dass neue und alte Komponenten koexistieren können. Es bietet Referenz-Architekturen und Leitprinzipien, die Architekten jeder Organisation dabei unterstützen, moderne Technologien und Architekturen aufzubauen und gleichzeitig die Schaffung von Mikrosilos oder Ad-hoc-Lösungen zu vermeiden. Das Projekt ist noch recht jung, bietet aber viele nützliche Ansätze zur Etablierung eines konsistenten API-Managements in hybriden IT-Infrastrukturen.

Bei der Erstellung von APIs ist es wichtig, einem API-First-Ansatz zu folgen. Dabei ist bereits im Vorfeld der App-beziehungsweise Backend-Entwicklung eine stabile API-Definition festzulegen, die den fachlichen Anforderungen gerecht wird. Abgestimmte und zentral erhältliche API-Style-Guides bilden dafür die Grundlage und definieren die Core-Prinzipien, Protokolle, Nachrichten-Formate, Datenmodelle, Typen, Testbarkeit, Wartung, Erweiterbarkeit, Skalierbarkeit, Ausführbarkeit und Benutzbarkeit. Für das konsistente Design gemäß API-First, kontinuierlichen Tests auf Basis integrierter API-Mock-Server, Versionierung und Kollaborationsmöglichkeiten bietet Oracle eine attraktive Lösung namens „Apiary“ an [8].

Nach Meinung des Autors ersetzen API-Management-Plattformen aber keinesfalls die klassischen Integrations-Plattformen – wenngleich die Feature-Sets entsprechender Produkte mitunter Überschneidungen aufweisen. Die wesentlichen Aufgaben der API-Plattformen lauten: Absicherung von Endpunkten, Auffindbarkeit, Verwaltbarkeit, Design-first und Monetarisierung. Integrations-Plattformen zielen auf Konnektivität, Transformation von Daten und Nachrichten-Formation sowie Caching und Skalierung ab. APIs sollten so schnell und schlank wie möglich sein. Sogenannte „API-Gateways“ stehen häufig in einer Demilitarized Zone (DMZ). Sie validieren Anfragen und wehren sie gegebenenfalls ab. Sensible Daten dürfen nicht auf der Festplatte oder im Speicher der DMZ gespeichert sein.

Dies wird mithilfe der Integrations-Plattform gelöst. Deren Hauptaufgabe liegt in der Transformation komplexer Payloads und der Orchestrierung von Service-Aufrufen. Durch die Verwendung eines robusten Caching-Systems können Skalierbarkeit und hohe Leistung erreicht werden,

während die Daten in einer ordnungsgemäß gesicherten Zone gehalten werden. Darüber hinaus stellt die Integrationsplattform die Konnektivität zu inkompatiblen Backend-Systemen (etwa als On-Premises) bereit. Die Verwendung von Adaptern in einem API-Gateway würde zu gesteigerter Komplexität auf dem API-Layer führen. Bedeutet dies aber, dass ein API-Gateway diese Funktionen niemals anbieten sollte? Nicht unbedingt. Es gibt Fälle, in denen es sinnvoll ist, Daten in einem API-Gateway zwischenspeichern oder sogar einfache Transformationen durchzuführen. Hier ist es schwer, ein „one size fits all“ zu finden. Die Auswirkungen und Risiken sind demnach von Fall zu Fall zu entscheiden.

Betrieb in hybriden Infrastrukturen

Dieser Artikel behandelt insbesondere die Herausforderungen im Bereich des Datenschutzes, der Kosten und der System-Integration. Der Betrieb von hybriden Infrastrukturen verfügt über zusätzliche umfassende Anforderungen; dazu zählen insbesondere ein End-to-End-Infrastruktur- und ein Applikations-Monitoring, das sowohl Multi-Cloud als auch On-Premises-Systeme berücksichtigt. Weitere wichtige Aspekte sind umfassende Log-Auswertungen, Alerting, Aufgaben-Automatisierung sowie die proaktive Überwachung von Sicherheitsbedrohungen.

Die Oracle Management Cloud adressiert die meisten dieser Anforderungen [9]. Dabei handelt es sich um eine Suite integrierter Überwachungs-, Verwaltungs- und Analyse-Cloud-Angebote. Die Lösung wurde speziell für hybride Umgebungen entwickelt: On-Premises, Oracle Cloud und Cloud-Services von Drittanbietern. Betriebsdaten in IT-Organisationen, einschließlich maschinengenerierter Daten, können unterschiedlicher Art und Größe sein und werden häufig in mehreren Systemen gespeichert. Mit der Oracle Management Cloud lassen sich hochgeladene Daten in einer einzigen einheitlichen Plattform speichern und auswerten. Die Plattform analysiert Daten automatisch auf Basis von Machine-Learning-Algorithmen und Korrelation. Die Daten der zu überwachenden Entitäten (wie Datenbanken, Host Server, Compute-Ressourcen und Applikationsserver) werden von Agenten gesammelt. Zur

Vertiefung empfiehlt sich einen Blick in das umfassende Dokumentationsangebot [10].

Fazit

Hybride Architekturen erhöhen häufig die Komplexität, sind allerdings essenziell, um im Zeitalter der Digitalisierung Schritt halten zu können. Dafür ist es wichtig, dass das Modell der geteilten Verantwortung verinnerlicht wird. Integrations-Plattformen stellen die Konnektivität sicher und ermöglichen die Zusammenführung neuer Services mit vorhandenen Systemen (horizontale Integration). Für die Absicherung, Zugriffsbeschränkung und Monetarisierung von nutzerzentrischen APIs bietet sich die Etablierung einer API-Management-Plattform an. Nur wer in hybriden Infrastrukturen die volle Kostentransparenz besitzt, spart Geld und fördert Innovation.

Quellen

- [1] Artikel-29-Datenschutzgruppe, Stellungnahme 5/2012 zum Cloud Computing; Arbeitskreis Technik und Medien der Konferenz der Datenschutzbeauftragten des Bundes und der Länder, Orientierungshilfe Cloud Computing, 01.07.2012, https://www.lida.bayern.de/media/wp196_de.pdf
- [2] Oracle Corporation: Oracle Cloud Infrastructure Security, https://docs.us-phoenix-1.oraclecloud.com/Content/Resources/Assets/oci_security.pdf
- [3] <https://cloudcheckr.com>
- [4] <https://www.cloudability.com>
- [5] <https://www.rightscale.com>
- [6] Jim Harris, The Cloud is shifting our Center of Gravity, 19. Juli 2012: <http://www.ocdqblog.com/home/the-cloud-is-shifting-our-center-of-gravity.html>
- [7] <https://cloud.oracle.com/api-platform>
- [8] <https://apiary.io>
- [9] <https://cloud.oracle.com/management>
- [10] <https://docs.oracle.com/en/cloud/paas/management-cloud/index.html>



Danilo Schmiedel
danilo.schmiedel@opitz-consulting.com



Serverless mit Fn Project

Dr. Frank Munz, munz & more

Das vorherrschende Cloud-Thema im Jahr 2017 war „serverlose Architekturen“. So gab es auf der DevOxx in Belgien, einer der besten Entwickler-Konferenzen in Europa, gleich sieben verschiedene Präsentationen zu diesem Thema.

Dieser Artikel, eine Übersetzung des auf Oracle Developer im März publizierten Originalartikels von Dr. Frank Munz [1], ist eine Einführung in Fn Project. Es unterscheidet sich von den meisten anderen serverlosen Lösungen: Fn ist Cloud-agnostisch, polyglott, Open Source und hat nur Docker als einzige Abhängigkeit. Es ist außerdem brandneu; Fn wurde auf der JavaOne 2017 als Open Source freigegeben.

Dieser Artikel behandelt folgende Punkte: Zunächst werden die Grundlagen geklärt und dem Entwickler eine kurze Einführung in die serverlose Welt gegeben. Dabei kommen die einzigartigen Vorteile

zur Sprache, die oft nicht ganz eindeutig definiert sind. Zweitens, und am wichtigsten, wird der schnelle Einstieg in die Programmierung mit dem neuen Fn Project gezeigt. Für eine praxisnahe Entwicklung werden Go und Java, Monitoring, Tests, lokale Entwicklung, JSON-Parameterübergabe, Verwendung von Docker Hub und der Einsatz von Fn in der Cloud behandelt. Drittens gibt es eine Übersicht der aktuellen Ankündigungen und darüber, was als Nächstes zu erwarten ist. Fn ist erst der Anfang einer Reise in die serverlose Welt.

„Serverlos“ ist offensichtlich kein besonders kluger Name. Ganz ehrlich: Die

IT-Branche ist schlecht bei der korrekten Namensgebung und präzisen Definition von neuen Konzepten. Cloud Computing läuft nicht irgendwo in den Wolken, Data Lakes sind nicht nass und serverlos braucht Server. Deshalb zunächst eine Definition von „serverlos“ und der damit verwandten Konzepte.

Function as a Service (FaaS)

Function as a Service (FaaS) startete als Cloud-Service im Jahr 2014 mit AWS Lambda. Die Idee dahinter war einfach

und revolutionär zugleich: Es wird Quellcode ausgeführt, aber man muss sich keine Gedanken über die darunter liegenden Laufzeit-Umgebungen der Programmiersprachen, die Container, die virtuelle Maschine oder die Server machen. Im einfachsten Falle kopiert der Entwickler seinen Quellcode, fügt ihn in eine Maske des FaaS-Service in der Cloud ein und der Code wird ausgeführt.

Technisch gesehen sind Cloud-basierte FaaS-Lösungen auf der Basis von Containern implementiert (also ähnlich wie Docker, aber nicht unbedingt mit Docker selbst). Dieser Container ist für den Endanwender üblicherweise nicht sichtbar. Eine serverlose Funktion wird nur ausgeführt, wenn sie durch ein Ereignis ausgelöst wird; deshalb heißt es auch „ephemeral compute“. Bei FaaS gibt es also keinen permanent laufenden Server für einen Benutzer; es existiert auch keine Laufzeit-Umgebung wie WebLogic oder Tomcat mit einer permanent geöffneten IP-Adresse und Port. Ereignisse, die eine solche serverlose Funktion auslösen können, hängen vom Cloud Provider ab. Übliche Beispiele für Ereignisquellen sind: ein Datei-Upload, eine REST-Anfrage oder eine Nachricht, die von einem Nachrichtensystem aufgenommen wird.

Was FaaS in Public Clouds besonders interessant macht, ist der Umstand, dass nur pro Aufruf der Funktion bezahlt werden muss. Außerdem ist die Skalierung automatisiert, eine Konfiguration für die Anzahl der Funktions-Instanzen ist also nicht erforderlich. Das Konzept „nicht zahlen für Leerlauf“ ist verlockend. Einige Anwendungsfälle berichten von Kosteneinsparungen von ein bis zwei Größenordnungen, wenn eine traditionelle, serverbasierte Anwendung ersetzt wurde [2].

Man sollte allerdings fair sein und auch die Kehrseite betrachten. Die wesentlichen Bedenken bei heutigen FaaS-Implementierungen betreffen die Abhängigkeit von einem bestimmten Cloud Anbieter, den sogenannten „Vendor Lock-in“:

- Die Funktion selbst wird über ein typisiertes, Cloud-Provider-spezifisches Ereignis ausgelöst.
- Die Verdrahtung verschiedener Ereignisquellen mit serverlosen Funktionen erzeugt eine harte Abhängigkeit zwischen der serverlosen Funktion und weiteren Diensten des Cloud-Providers.

Das Konzept von FaaS entwickelt sich jedoch weiter. So gibt es Diskussionen unter Software-Architekten, ob Funktionen besser gleich als Container implementiert werden sollten [3]. Derzeit legt jedoch keiner der großen Public-Cloud-Provider den Container offen, der intern für die Ausführung serverloser Funktionen verwendet wird.

Microservices vs. FaaS

Eine Microservices-Architektur versucht, eine Anwendung als eine Menge unabhängiger Services zu implementieren. Jeder Service läuft in seinem eigenen Prozess und hat seine eigenen Daten; untereinander kommunizieren die Services über ein leichtgewichtiges Protokoll [4]. Es stellt sich daher die Frage, ob FaaS nicht auch nur eine Art von Microservice ist. Kurz: FaaS erfüllt theoretisch die Definition eines Microservice. Allerdings müssen serverlose Funktionen in der Praxis erst zu einem Microservice zusammengefasst werden. Aber wie?

„Serverlos“ ist ein Architektur-Trend, der darauf ausgelegt ist, „alle Beziehungen zur Infrastruktur“ zu reduzieren [5]. FaaS ist also serverlos. Ein serverloser Cloud-Service ist ein PaaS mit echter Bezahlung nach Benutzung und automatisierter Skalierbarkeit. Als Beispiel für einen serverlosen Cloud-Service kann man sich einen Nachrichtendienst vorstellen. Wenn nur für die Anzahl der erzeugten und verarbeiteten Nachrichten bezahlt wird und der Service automatisch skaliert, können wir ihn berechtigterweise „serverlos“ nennen. Wenn Server mit den Message-Brokern sichtbar sind und pro Stunde für diese Server bezahlt wird, unabhängig davon, ob Nachrichten erzeugt oder verarbeitet werden, dann ist der Service nicht serverlos.

Derzeit gibt es mehr als ein Dutzend FaaS-Frameworks oder -Plattformen (eine Übersicht über die Projekte siehe [6]). Diese Projekte lassen sich je nach Zielrichtung in drei unterschiedliche Kategorien einteilen, wobei jede Kategorie typischerweise die Eigenschaften der vorherigen enthält [7]:

- **Komplexität**
Die Komplexität einer Cloud-basierten FaaS-Implementierung eines bestimmten Anbieters wird reduziert, etwa durch die Konfiguration eines

API-Gateways und Zugriffsverwaltung, die für eine REST-basierte Funktion benötigt wird. Ein typisches Beispiel für diese Kategorie ist AWS Chalice.

- **Portierbarkeit**
Das Framework dient zur Abstraktion zwecks Portierbarkeit und zur zusätzlichen Vereinfachung der FaaS-Implementierung verschiedener Public-Cloud-Anbieter. Ein beliebtes Beispiel ist das „serverless.com“-Framework.
- **Standards**
Es handelt sich um eine Standard-basierte, serverlose Plattform oder ein Framework, um laufende Funktionen vom Betrieb der Server zu abstrahieren. Diese Frameworks werden üblicherweise ohne einen Bezug zu einem bestimmten Cloud-Provider entwickelt. Wenn man solch ein Framework auf IaaS betreibt, werden Server abstrahiert, automatische Skalierung ist möglich, aber durch das IaaS-Preismodell wird keine echte Pro-Aufruf-Basis geschaffen. Beispiele für diese Kategorie sind Open FaaS und Fn Project.

Fn Project

Fn Project ist eine serverlose Plattform mit einer Vielzahl einzigartiger Vorteile: Es ist Container-basiert, polyglott, Cloud-agnostisch und hat Docker als einzige Abhängigkeit. Zum derzeitigen Zeitpunkt ist Fn Project eine Software-Plattform und es existiert noch kein FaaS-Cloud-Service. Die Entwicklung eines solchen Cloud-basierten Service wurde allerdings von Oracle angedeutet. Um alle Eigenschaften von Fn zu verstehen, ist es am einfachsten, Fn in Aktion zu erleben und selbst einige Funktionen laufen zu lassen.

Fn lässt sich leicht auf Windows- und Unix-Systemen mit dem einzeiligen Kommando „\$ curl -LSs https://raw.githubusercontent.com/fnproject/cli/master/install | sh“ installieren. Auf Mac OS wird Fn mit „brew“ installiert. Einzelheiten zur Installation siehe [8].

Grundlagen mit Go

Um die Fn-Eigenschaften zu verstehen, zu Beginn eine einfache Go-Funktion.

```
# create oradev and with boilerplate for go
$ fn init --runtime go oradev
$ cd oradev
```

Listing 1

```
$ fn run

Building image oradev:0.0.1 ..
{"message":"Hello World"}
```

Listing 2

```
$ tree
.
├── func.go
├── func.yaml
└── test.json
```

Listing 3

```
$ fn deploy --app mygo --local

Deploying oradev to app: mygo at path: /oradev
Bumped to version 0.0.2
Building image oradev:0.0.2 ..
Updating route /oradev using image oradev:0.0.2...
```

Listing 4

```
# check deployed applications
$ fn apps list
mygo

# check existing routes
$ fn routes list mygo

path image endpoint
/oradev oradev:0.0.2 localhost:8080/r/mygo/oradev
```

Listing 5

```
$ # invoke function via fn server
$ fn call mygo /oradev
{"message":"Hello World"}
```

Listing 6

```
$ # invoke function with UNIX curl
$ curl localhost:8080/r/mygo/oradev
{"message":"Hello World"}
```

Listing 7

```
$ # run the docker image
$ docker run oradev:0.0.2
{"message":"Hello World"}
```

Listing 8

```
$ docker run --rm -it --link fnserver:api -p 4000:4000 -e "FN_API_
URL=http://api:8080" fnproject/ui

> FunctionsUI@0.0.21 start /app
> node server

Using API url: api:8080
Server running on port 4000
```

Listing 9

Listing 1 zeigt das Erstellen eines neuen Verzeichnisses, in dem mit der Go-Sprache eine lokale Fn-Funktion initialisiert wird. Anschließend lässt sich die Funktion sofort mit dem Run-Kommando ausführen und die Ausgabe der HelloWorld-Anwendung betrachten (siehe Listing 2).

Die erzeugte Datei zeigt, warum das möglich war. Es gibt eine Konfigurationsdatei namens „func.yaml“, die die Versionsnummer und die Laufzeit von Go spezifiziert. Außerdem wird eine Standard-Go-Routine („func.go“) mit einer Testdatei „test.json“ erzeugt (siehe Listing 3). Wenn man die Ausgabe beim Ausführen der Funktion genau betrachtet, ist zu sehen, dass ein Docker-Image „oradev:0.0.1“ erstellt wurde.

Das Kommando „fn run“ ruft diese Funktion direkt auf. Um einen Endpunkt für die Funktion zu erstellen, ist als Erstes der Fn Server mit „\$ fn start“ in einem anderen Terminal zu starten. Wenn der Server läuft, lässt sich die Funktion mit dem Kommando in Listing 4 installieren. Der Name der Funktion wird aus dem Ordernamen entnommen. Optional kann man den Namen auch in der Datei „func.yaml“ angeben.

Bei der Installation der Funktion erhöht sich die Version des Docker-Image auf 0.0.2. Durch das „Deploy“-Kommando wurde eine neue Anwendung registriert. Darüber hinaus ist ein Endpunkt für die Funktion entstanden. Um die Erstellung des Endpunkts zu prüfen, führt man die zwei Fn-Kommandos zum Auflisten der Anwendungen und der neuen Route aus (siehe Listing 5).

Die Funktion ist jetzt am Fn Server registriert, der wie ein Mikro-API-Gateway agiert. Er nimmt Anfragen an die angeführten Endpunkte entgegen und ruft die installierten Funktionen auf. Listing 6 zeigt, wie man das selbst ausprobieren kann. Wahlweise lässt sich die Funktion auch mit einem einfachen „curl“-Kommando von der Unix-Kommandozeile aufrufen, da Fn Server eine URL für die Funktion bereitstellt (siehe Listing 7). Eine weitere Möglichkeit ist das direkte Ausführen des Docker-Image (siehe Listing 8). Alle drei Ansätze liefern identische Ergebnisse.

Container/Funktionsdualität

Es ist zu beachten, dass ein Ausführen der Funktion über den URL-Endpunkt und das Ausführen des Docker-Containers das glei-

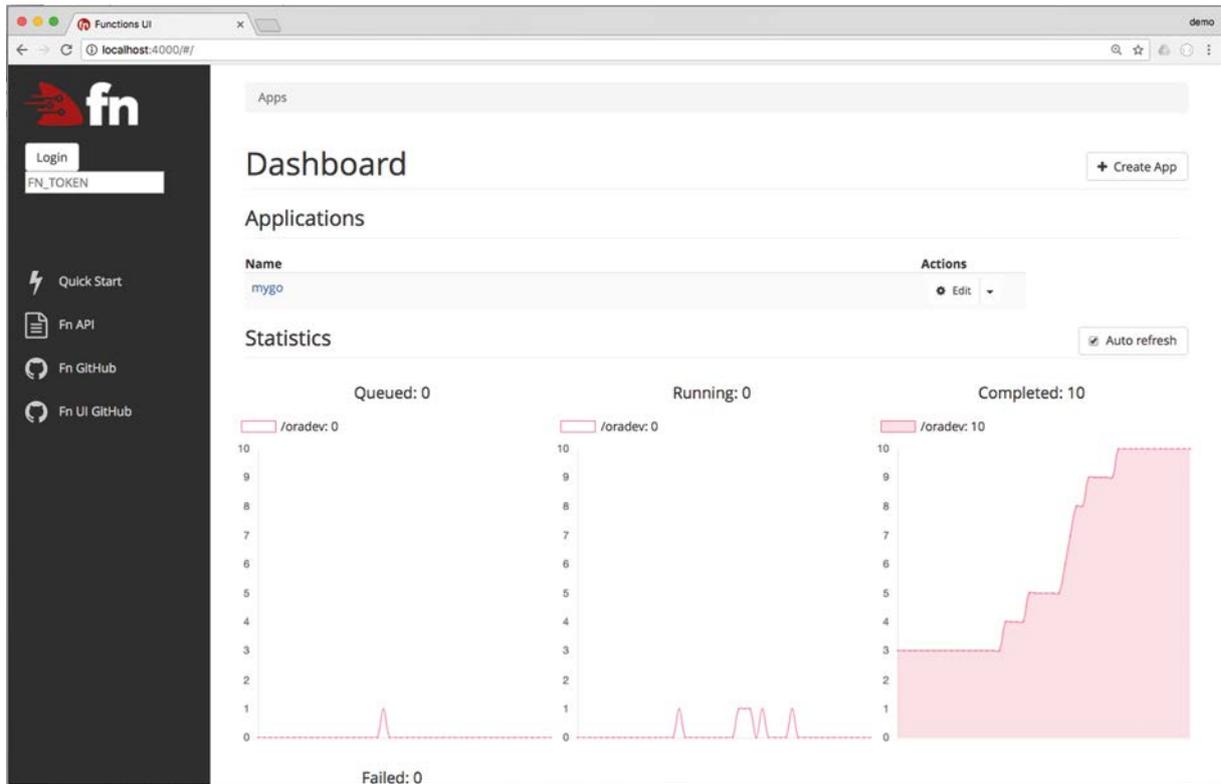


Abbildung 1: Die Überwachungskonsole

che Ergebnis liefern! Das Docker-Image, das die Funktion enthält, wurde automatisch erstellt, ohne irgendeine zusätzliche Konfiguration oder ein Kommando. Fn Project gibt so gesehen das Docker-Image kostenlos dazu. Zwei der vielen Vorzüge bei der Verwendung von Docker sind:

- Docker-Images lassen sich in jeder Public Cloud verwenden.
- Docker Hub kann Docker-Images speichern. Funktionen können somit im Docker Hub gespeichert werden.

Diese beiden Konzepte werden nachfolgend genauer betrachtet. Mit Fn lässt sich also einfach eine Funktion schreiben, ohne dabei auf Docker zu achten. Dennoch besteht der Vorteil, dass die Funktion in einem Container läuft.

Fn Monitoring

Fn Project bietet außerdem ein einfaches Werkzeug zur Überwachung, das als Docker-Container gestartet werden kann (siehe Listing 9). Um auf die Konsole zuzugreifen, öffnet man einen Browser und verbindet sich mit Port 4000. Um eine Veränderung in den Kurven der Überwachungskonsole zu

```
$ curl localhost:8080/metrics | head

# HELP fn_api_completed Completed requests by path
# TYPE fn_api_completed counter
fn_api_completed{app="mygo",path="/oradev"} 11
# HELP fn_api_queued Queued requests by path
# TYPE fn_api_queued gauge
fn_api_queued{app="mygo",path="/oradev"} 0
# HELP fn_api_running Running requests by path
# TYPE fn_api_running gauge
fn_api_running{app="mygo",path="/oradev"} 0
# HELP fn_docker_stats_cpu_kernel docker_stats metric cpu_kernel
...
```

Listing 10

```
$ cd ~ && mkdir javatest && cd javatest
$ fn init --runtime java
Runtime: java
Function boilerplate generated.
func.yaml created.
```

Listing 11

sehen, startet man die Go-Funktion einige Male [9] (siehe Abbildung 1).

Prometheus Monitoring

Für eine etwas anspruchsvollere Überwachungslösung ist Prometheus von

der Cloud Native Computing Foundation (CNCF) zusammen mit CNCF Grafana eine gute Wahl. Fn exportiert Metriken, die eine Überwachung mit Prometheus ohne zusätzliche Konfiguration zulassen. Auch ohne die Installation von Prometheus lassen sich die Metriken, die für Prometheus exportiert werden, mit

der URL „/metrics“ betrachten (siehe Listing 10).

Weitere Einzelheiten zu Prometheus und Fn sind unter [10] beschrieben. Man kann auch mit dem Go-Beispiel ein HelloWorld-Beispiel in Java erstellen, indem der Laufzeitschalter auf „Java“ gestellt wird (siehe Listing 11). Java 9 ist die Standard-Java-Version. Es ist zu beachten, dass für ein Java-Projekt auch eine Maven-Datei „pom.xml“ und ein Unit-Test „HelloFunctionTest.java“ erstellt werden (siehe Listing 12).

JSON-Parameter-Bereitstellung und Funktionslogik

Um einige erweiterte Eigenschaften von Fn zu zeigen, verlassen wir das HelloWorld-Beispiel in Java und betrachten ein Beispiel für eine Recommendation-Engine. Man kann das Beispiel mit dem Kommando in Listing 13 aus GitHub holen. Das API der Funktion simuliert die Empfehlungslogik. Sie verwendet ein POJO als Eingabe-Parameter. Dieser definiert das Alter des Reisenden, das Ziel und den Monat der Reise (siehe Listing 14).

Dieses Beispiel wird automatisch im Docker Hub eingechekkt – im Gegensatz zum vorherigen Go-Beispiel, das nur lokal vorgehalten wurde. Zu diesem Zweck wird die Umgebungsvariable „FN_REGISTRY“ auf die „DOCKER_ID“ gesetzt und außerdem im Docker Hub eingeloggt. Man ersetzt also im Beispiel „DOCKER_ID“ durch sein eigenes Docker-Log-in (siehe Listing 15) und installiert anschließend die Funktion. Die Funktion wird für eine Abenteuerreise-Anwendung verwendet, daher auch der Name (siehe Listing 16).

Aus der Ausgabe ist zu erkennen, dass das Docker-Image erstellt und im Docker Hub unter „DOCKER_ID/fn-recommend:0.0.2“ eingechekkt wurde. Man kann wie gewohnt die neue Anwendung und die neue Route im Fn Server prüfen (siehe Listing 17) und die Funktion mit einem „POST“-Request über ein „Curl“-Kommando ausführen, wenn man die erforderliche JSON-Datenstruktur für den Request liefert. Standardmäßig verwendet Fn das Jackson-Java-Framework, um automatisch JSON-Eingabeparameter für den korrekten Java-Typ bereitzustellen. Man kann aber auch jedes beliebige

```
$ tree
.
├── func.yaml
├── pom.xml
└── src
    ├── main
    │   ├── java
    │   │   ├── com
    │   │   │   ├── example
    │   │   │   │   └── fn
    │   │   │   │       └── HelloFunction.java
    └── test
        ├── java
        │   ├── com
        │   │   ├── example
        │   │   │   └── fn
        │   │   │       └── HelloFunctionTest.java
```

Listing 12

```
$ git clone https://github.com/fmunz/fn-recommend.git
$ cd fn-recommend
```

Listing 13

```
# check the API of the handler function
$ grep handle src/main/java/com/munzandmore/fn/RecommendFunction.java
public String handleRequest(Traveller t) {
# examine the Traveller POJO
$ cat src/main/java/com/munzandmore/fn/Traveller.java
package com.munzandmore.fn;
public class Traveller {
    public Integer age ;
    public String destination ;
    public String month;
}
```

Listing 14

```
# set environment for Docker hub
$ export FN_REGISTRY=YOUR_DOCKER_ID
$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If
you don't have a Docker ID, head over to https://hub.docker.com to cre-
ate one.
Username: DOCKER_ID
Password:
Login Succeeded
```

Listing 15

```
$ fn deploy --app advtravel
Deploying fn-recommend to app: advtravel at path: /fn-recommend
Bumped to version 0.0.2
Building image DOCKER_ID/fn-recommend:0.0.2
Pushing DOCKER_ID/fn-recommend:0.0.2 to docker registry...The push re-
fers to repository [docker.io/DOCKER_ID/fn-recommend]
7e2c18073a13: Layer already exists
...
0.0.2: digest: sha256:549e492a08d924dcfeef5f0354dc7d2df57cba820bcfa7ec5
50a1779a173983c size: 1997
Updating route /fn-recommend using image DOCKER_ID/fn-recom-
mend:0.0.2...umped to version 0.0.2
```

Listing 16

Framework zur Bereitstellung von JSON oder anderen Formaten wie XML etc. verwenden (siehe Listing 18).

Für die Evaluierung unterschiedlicher Eingabe-Parameter ist ein grafisches Tool wie Postman geeigneter. *Abbildung 2* zeigt, was die Fn-basierte Beispiel-Anwendung für eine Reise nach Sydney empfiehlt. Die Ausgabe sollte wie in *Abbildung 3* aussehen.

Fn in Public Clouds (IaaS)

Eine häufige Frage ist, wie Fn Project als Cloud-agnostisches Framework in Public Clouds verwendet werden kann. Ähnlich wie bei der lokalen Installation, die in den Beispielen gezeigt wurde, kann es auf jeder Public Cloud IaaS installiert werden. Bei den meisten IaaS-Clouds reicht es aus, das Installations-Kommando direkt bei der Erstellung einer Rechner-Instanz als sogenannte „user data“ zu übergeben, also als Kommandos, die ausgeführt werden, wenn die Instanz bereitgestellt wird. Beim Betrieb in einer Public Cloud ist außerdem darauf zu achten, Zugriffsregeln für den Fn Server auf Port 8080 einzurichten, entweder von der eigenen IP oder von allen öffentlichen IP-Adressen ausgehend.

Wer Fn Project auf einem IaaS betreibt, bekommt allerdings nicht den Vorteil des echten „Bezahlt pro Aufruf“. Dies wäre

nur mit einem echten FaaS-Service eines Cloud-Providers der Fall. Die Funktionen selbst laufen dennoch aus Anwendersicht serverlos in einer standardisierten, portablen und skalierbaren Art und Weise. Wenn der Fn Server beim Cloud-Provider seiner Wahl läuft, gibt es zwei Möglichkeiten, die Recommendation-Engine aus

dem Beispiel zu installieren (siehe Listing 19).

Es ist zu beachten, dass mit den beiden Kommandos nie die Funktion oder das Container-Image auf die Cloud-Instanz zu kopieren war. Wenn die Funktion das erste Mal aufgerufen wird, checkt Fn den Docker-Container, speichert ihn lokal

```
$ fn apps list
advtravel
mygo

$ fn routes list advtravel
path      image      endpoint
/fn-recommend DOCKER_ID/fn-recommend:0.0.2 localhost:8080/r/advtravel/
fn-recommend
```

Listing 17

```
$ cat testdata/muc.json
{
  "age": 41,
  "destination": "Munich",
  "month": "Oct"
}
# get a recommendation for Munich in October
$ curl -X POST --data @testdata/syd.json localhost:8080/r/advtravel/fn-recommend

Visit the Octoberfest!
# there is more test data under testdata/Casablanca.json
# see what is recommended for that city!
```

Listing 18

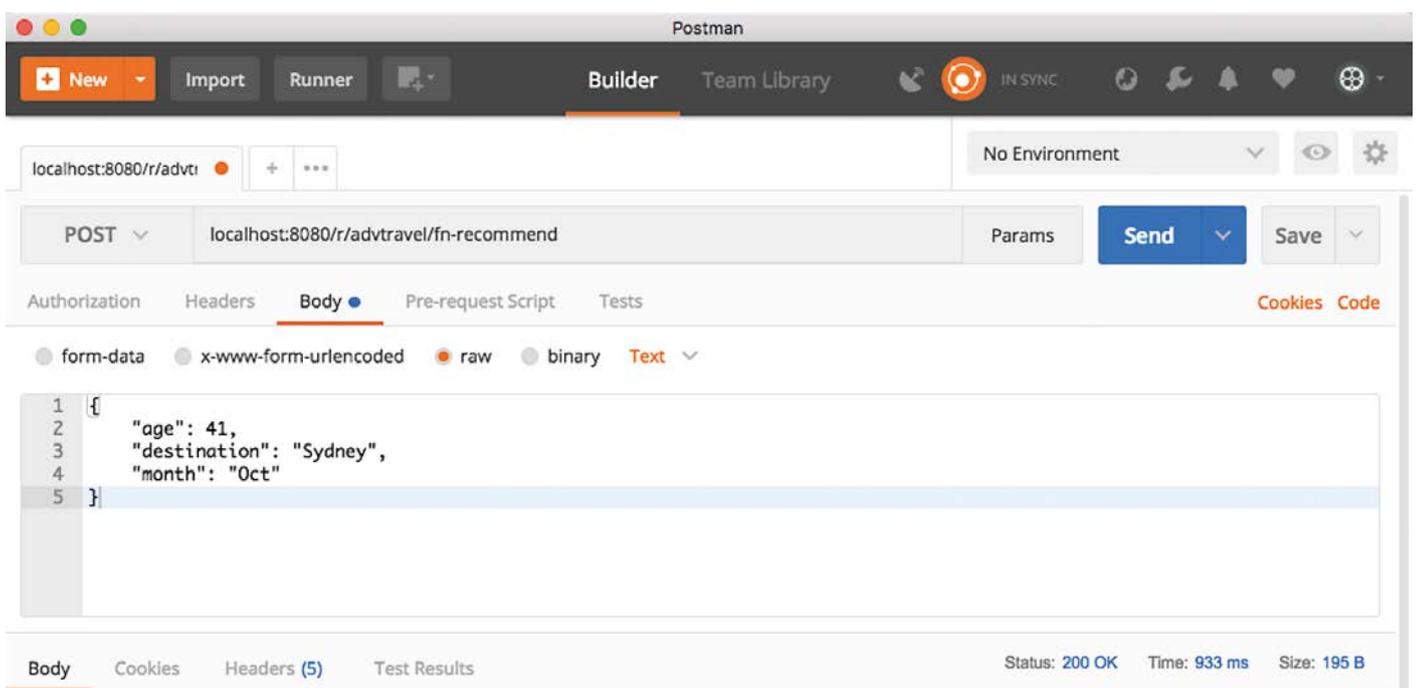


Abbildung 2: Eine Fn-basierte Beispiel-Anwendung



Abbildung 3: Die Ausgabe mit Postman

```
# example 1 (for demo purpose only, in production use approach below)
# note: run these commands on the cloud instance
$ fn apps create advtravel
$ fn routes create advtravel /fn-recommend DOCKER_ID/fn-recommend:0.0.2

# check for the created route
$ fn routes list advtravel
```

Listing 19

```
# example 2
# run these commands on cloud instance

$ export FN_API_URL=URLCloudInstance
$ fn deploy --app advtravel
$ fn routes list advtravel
```

Listing 20

```
$ curl -X POST --data @testdata/syd.json PUBLIC_IP:8080/r/advtravel/
fn-recommend
```

Listing 21

und führt anschließend die Funktion aus. Eine bessere Methode zur Installation der Funktion ist, die „FN_API_URL“-Umgebungsvariable lokal zu konfigurieren, sie auf die Remote-Cloud-Instanz zu setzen und anschließend das lokale Fn-Installationskommando gegen die Remote-Cloud-Instanz auszuführen (siehe Listing 20).

Wenn Fn in der Cloud läuft und die Anwendung installiert ist, kann man von jedem lokalen Rechner aus über die Kommandozeile oder mittels Postman auf die Anwendung zugreifen. Der Aufruf erfolgt genauso wie im lokalen Beispiel, man ersetzt einfach „localhost“ durch die öffentliche IP-Adresse der Cloud-Instanz (siehe Listing 21). Eine aufgezeichnete Demo von der DevOxx-Konferenz zur Installation einer Fn-basierten Anwendung mit der Recommendation-Engine auf IaaS steht unter [11].

JAX-RS, Spring Cloud und mehr

Da Fn Project nur Docker als Abhängigkeit hat und die Maven-Datei „pom.xml“ auch generiert wird, kann die Funktionsentwicklung einfach um weitere Java-Frameworks erweitert werden. Bisher wurde im Fn-Team an der Unterstützung von JAX-RS mit Fn Projects gearbeitet [12]. Spring unterstützt ebenfalls die Implementierung von Anwendungslogik als Funktionen mit ihrem Konvention-vor-Konfiguration-Ansatz. Spring-Cloud-Funktionen lassen sich mit Fn verwenden [13, 14].

Fn LB, eine separate Komponente, ist für Lastverteilung und intelligentes Routing zuständig. Werden Funktionen als „hot functions“ installiert, bleibt ein Container für 30 Sekunden bestehen und wird nicht für jeden Aufruf neu gestartet.

Fn LB leitet dann Aufrufe dorthin, um optimale Performance zu sichern [15].

Am Anfang dieses Artikels haben wir uns mit den Unterschieden zwischen Microservices und FaaS befasst und erklärt, dass ein Microservice typischerweise mehr als eine einzige Methode oder Funktion enthält. Heutzutage werden häufig grafische Tools oder übergeordnete PaaS-Dienste verwendet, um FaaS in sinnvolle, größere Services zu integrieren. Diese grafischen Tools bieten jedoch meistens keinen wirklich guten Einblick in die Einzelheiten des übergeordneten Service. Erkenntnisse aus Projekten mit ESB und BPEL zeigen, dass diese Details nicht gleichzeitig grafisch angezeigt werden können und deswegen häufig unter einem Eigenschaften-Tab des grafischen Modells versteckt sind. Einen solchen „Flow“ in einem grafischen Modell darzustellen, ist daher häufig eingeschränkt.

Fn Flow behandelt dieses Problem für Fn Project. Es verfolgt einen interessanten „Code first“-Ansatz, indem das Java-8-CompletableFuture-API mit Methoden wie „thenApply()“ oder „thenCompose()“ etc. verwendet wird. Grafische Tools oder große YAML-Dateien sind nicht erforderlich; die Verkettung der Funktionen erfolgt ausschließlich mit Java-8-Konstrukten und ist dadurch leicht lesbar. Eine interessante Anwendung dieses Konzepts wird bei der Verwendung von SAGAs anstelle einer ACID-Transaktion für eine Reisebuchungsanwendung [16] basierend auf Microservices gezeigt (siehe Abbildung 4).

Was auf den ersten Blick wie ein gewöhnliches Java-8-Programm aussieht, erinnert bei der Ausführung eher an Apache Spark. Die Ausführung findet parallel statt, die Eingabe-Parameter der Funktion werden übergeben und Rückgabe-Werte entsprechend zurückgegeben. Jede Funktion wird in seinem eigenen Container unter Verwendung von Verbindung, Fehler-

```

public void book1(TripReq input) {
    Flow f = Flows.currentFlow();

    FlowFuture<BookingRes> flightFuture =
        f.invokeFunction("./flight/book", input.flight, BookingRes.class);

    FlowFuture<BookingRes> hotelFuture =
        f.invokeFunction("./hotel/book", input.hotel, BookingRes.class);

    FlowFuture<BookingRes> carFuture =
        f.invokeFunction("./car/book", input.carRental, BookingRes.class);

    flightFuture.thenCompose(
        (flightRes) -> hotelFuture.thenCompose(
            (hotelRes) -> carFuture.whenComplete(
                (carRes, e) -> EmailReq.sendSuccessMail(flightRes, hotelRes, carRes)
            )
        )
    );
}

```

Abbildung 4: Der Einsatz von SAGAs anstelle einer ACID-Transaktion

behandlung und Fan In/Out ausgeführt. Fn Flow kann die Aufrufkurven nachverfolgen und sie visualisieren.

Fn on Kubernetes

Auf der KubeCon wurde im Dezember 2017 offiziell Support für Fn on Kubernetes angekündigt. Fn kann mit Helm, einem Paketmanager für Kubernetes, installiert werden. Vorkonfigurierte Pakete wie Fn Service, Fn UI, Flow Service und Flow UI werden als Helm-Chart bereitgestellt. Mit diesem lässt sich Fn auf jedem beliebigen Kubernetes-Cluster installieren [18]. Es ermöglicht außerdem den Betrieb von Fn auf dem brandneuen Oracle-managed Kubernetes Service, Oracle Container Engine (OCE) oder einer lokalen Installation von Minikube auf dem Laptop [19, 20].

Fazit

Fn Project ist ein interessanter neuer Ansatz für die serverlose Welt. Es ist Cloud-agnostisch und verhindert dadurch den Anbieter-Lock-in. Darüber hinaus sind Entwickler bei der Verwendung von Fn nicht an bestimmte Programmiersprachen gebunden. Funktionen werden automatisch als Docker-Image gebaut, ohne

dass der Entwickler dafür zusätzlich etwas tun muss. Die Funktionen können überall laufen und werden vom Docker Hub geladen.

Fn ist in die Projekte der Cloud Native Computing Foundation eingebunden und bietet Support für Kubernetes und Prometheus; hoffentlich kommen noch weitere dazu. Zu guter Letzt wird es interessant sein zu sehen, ob Oracle oder ein anderer Cloud-Provider einen Fn-basierten FaaS als PaaS in naher Zukunft mit Bezahlung pro Aufruf und voll automatisierter Skalierung anbieten werden [21].

Referenzen

- [1] Oracle Developer: https://developer.oracle.com/en_US/opensource/serverless-with-fn-project
- [2] Referenz Expedia: www.youtube.com/watch?v=gT9x9LnU_rE, Referenz Postlight: <https://goo.gl/uAMGYF>
- [3] Funktionen vs. Container: <https://medium.com/oracledevs/containers-vs-functions-51c879216b97>
- [4] Microservices, M. Fowler: <https://martinfowler.com/articles/microservices.html>
- [5] Mark Cavage, Serverless, Java One Keynote 2017: <https://www.youtube.com/watch?v=UNg9Imk60sg>
- [6] Overview FaaS Frameworks: <https://github.com/faas-lane/FaaS-Lane/tree/master/candidates>
- [7] Serverless Classification: <http://www.munzandmore.com/2018/cc/serverless-faas-frameworks-and-platform-classification>

- [8] Fn Project Installation: <https://github.com/fnproject/fn>
- [9] Fn UI: <https://github.com/fnproject/ui>
- [10] Fn Monitoring mit Prometheus: <https://medium.com/fnproject/announcing-prometheus-metrics-from-fn-2d0f9ddf0f09>
- [11] Fn Project on Public Clouds (IaaS): <http://www.munzandmore.com/2017/aws/fn-project-on-public-clouds>
- [12] Fn mit JAX-RS: <https://github.com/fnproject/fn-jrestless-example>
- [13] Spring-Cloud-Funktionen: <https://spring.io/blog/2017/07/05/introducing-spring-cloud-function>
- [14] Fn with Spring Cloud Functions: <https://medium.com/fnproject/announcing-spring-cloud-function-support-for-fn-project-921e54f49d99>
- [15] Fn LB: <https://github.com/fnproject/fn/tree/master/fnlb>
- [16] Serverless Sagas mit Fn Flow: <https://medium.com/fnproject/serverless-sagas-with-fn-flow-d8199b608b12>
- [17] Acht Gründe, warum wir Fn Project entwickelt haben: <https://medium.com/fnproject/8-reasons-why-we-built-the-fn-project-bcfe45c5ae63>
- [18] Fn Project mit Helm Chart für Kubernetes: <https://medium.com/fnproject/fn-project-helm-chart-for-kubernetes-e97ded6f4f0c>
- [19] Fn Project on Kubernetes: <https://blogs.oracle.com/developers/kubernetes-serverless-and-federation-oracle-at-kubecon-2017>
- [20] Oracle Container Engine: <https://blogs.oracle.com/developers/announcing-oracle-container-engine-and-oracle-container-registry-service>
- [21] Serverlos und Fn Präsentationen: <http://www.munzandmore.com/2017/cc/devoxx-2017-serverless>



Dr. Frank Munz
fm@munzandmore.com



Container Native Development Platform

Marcel Amende, ORACLE Deutschland B.V. & Co. KG

Beim Entwickeln in der Cloud und für die Cloud („Cloud Native Development“) stehen hohe Flexibilität und Interoperabilität im Fokus – bei einfacher Nutzbarkeit und kostengünstigem Betrieb. Derzeitige Top-Themen aus Sicht der Anwender sind die Docker-Containerverwaltung und das sogenannte „Serverless Computing“. Beide Technologien halten Einzug in die Oracle-Cloud-Infrastruktur und sind ein deutliches Zeichen dafür, dass Oracle als Mitglied der „Cloud Native Computing Foundation (CNCF)“ [1] Open-Source-Projekten eine sehr hohe Wertschätzung entgegenbringt.

Neue Technologien krempeln Märkte um, verändern klassische Geschäftsfelder oder eliminieren diese sogar. Unternehmen sind gezwungen, schnell zu handeln. Ein Beispiel: Heute ersetzen Apps wie Uber und Mytaxi die Taxizentralen. Morgen ersetzen autonome Fahrzeuge die Taxifahrer. Man muss sich ständig neu erfinden, digital transformieren, um mit der Veränderungsgeschwindigkeit der Konkurrenz und aufkommenden Start-ups mithalten zu können. Neue Funktionen und neue Geschäftsmodelle müssen entwickelt und

in der Praxis erprobt werden. Software ist der Treibstoff dieser Innovation, agile Software-Entwicklung ist gefragt. Ganz wie in der realen Welt bringen Container die neue Ware zum Konsumenten.

Agile Software-Entwicklung und Microservices

Die klassische Methodik der Software-Entwicklung ist das Wasserfall-Modell. Man beginnt mit dem Schreiben eines

Lastenhefts und der Definition der System-Architektur in der Entwurfsphase, gefolgt von der Implementierung, den Abnahmetests und final dem Ausrollen der Software. Bei einfachen Projekten mit klaren Anforderungen ergibt sich daraus eine große Planungssicherheit. In einem herausfordernden Projekt werden Probleme jedoch erst sehr spät sichtbar, da man erst zum Ende der Implementierung ein lauffähiges und testbares Produkt erhält. Anpassungen sind zu diesem Zeitpunkt schwierig und teuer, da

der gesamte Planungsaufwand bereits zu Projektbeginn erfolgt ist. In der Praxis benötigt man nach der Wasserfall-Methode mehrere Monate oder sogar Jahre, um ein neues Software-Release zu erstellen.

In Innovationsprojekten sind die Anforderungen zu Projektbeginn oft unklar. Man weiß beispielsweise noch nicht, wie hoch die Akzeptanz der Anwender ist oder ob sich das neue Geschäftsmodell in der Praxis bewährt. Gleiches gilt für die Leistungsfähigkeit neuer Technologien. Erst nach mehrmonatiger Entwicklung grundsätzliche Probleme, beispielsweise mit der Performance, zu erkennen, wäre fatal.

Ein alternativer Ansatz ist eine modulare Entwicklung, die Implementierung sogenannter „Microservices“. Die Einzelmodule sind weniger komplex, schneller implementiert und leichter testbar. Bei agilen Projektmethoden wie SCRUM entwickelt man die einzelnen Module in Sprints von wenigen Tagen bis wenigen Wochen Dauer, in kleinen Teams, parallel und inkrementell. Jedes Inkrement führt zu einem isoliert lauffähigen und testbaren Produkt. Man überlässt den Teams die Auswahl der geeigneten Werkzeuge und Technologien, die sich von Modul zu Modul unterscheiden können. In täglichen Meetings und regelmäßigen Reviews werden Erfahrungen geteilt und Probleme schnell sichtbar. Umso wichtiger wird es aber, Routine-Aufgaben wie Kompilieren und Ausbringen von Software-Modulen vollständig zu automatisieren.

DevOps als neue Unternehmenskultur

Schnell und kontinuierlich neue Software-Versionen auszubringen, ist eine sehr verantwortungsvolle Aufgabe, die nur funktionieren kann, wenn Entwickler und Administratoren gemeinsam Verantwortung übernehmen. Der Entwickler verantwortet seinen Code und seine Werkzeuge bis in die Produktion. Der Administrator arbeitet eng mit den Entwicklungsteams zusammen, um eine vollautomatisierte, nahtlose Ausbringung zu gewährleisten. Man braucht eine gemeinsame, gesunde Fehlerkultur.

Bei moderner Microservice-Entwicklung konzentriert man sich auf die zu implementierende Funktionalität und nicht auf Technologie-Schichten wie bei einer serviceorientierten Architektur oder ei-

ner Mehrschicht-Implementierung. Es gilt der Grundsatz: Mache genau eine Sache, aber mache sie richtig. Der Entwickler genießt große Freiheitsgrade. Er wählt für jedes Modul die seiner Meinung nach am besten geeigneten Werkzeuge, Programmiersprachen, Datenbanken und Laufzeit-Umgebungen. Die Kunst besteht darin, diese Freiheiten nicht zu einem nicht mehr handhabbaren Wildwuchs werden zu lassen.

Spätestens beim Datenschutz und bei der Datensicherheit hört der Spaß mit der Freiheit auf. Hier kommen neue Rollen auf Mitarbeiter mit Betriebserfahrung und Administratoren zu, die bei der agilen Entwicklung ansonsten oft als Blockierer gesehen werden. Sie können die agilen Teams bei der Automatisierung und auch in Sachen Daten-Sicherheit, -Wiederherstellung und -Konsistenz ideal beratend unterstützen. Vor allem in der Umsetzung der EU-Datenschutz-Grundverordnung ist ein umsichtiger und vorausschauender Umgang mit Daten zwingend, um hohe Aufwände bei der Lösch- und Auskunftspflicht und schlimmstenfalls Strafen zu vermeiden.

Ein ideales agiles Team wäre also im Sinne der neuen Kultur eine heterogene Mischung aus Entwicklern und aus den Entwicklungsprozess beratend begleitenden Administratoren. Viele sich wiederholende Routinetätigkeiten aus dem Bereich der Datenbank- und System-Administration werden in Zeiten der Cloud vom Betreiber oder von autonomen Systemen erledigt. Der moderne Administrator wird sich in Zukunft qualitativen Aufgaben zuwenden können und die Rolle des vertrauenswürdigen Beraters in den Entwicklungsteams übernehmen.

Klassische DevOps-Werkzeuge

Es gibt eine Vielzahl von Werkzeugen, die bei der Automatisierung des kompletten DevOps-Zyklus unterstützen; zu meist sind diese quelloffen. Der Oracle Developer Cloud Service [2] umfasst eine Sammlung der verbreitetsten Werkzeuge und deckt den kompletten Zyklus in Form eines Dienstes in der Oracle-Cloud ab. Dieser ist kostenfrei für Nutzer der Oracle-Universal-Credit-PaaS- und -IaaS-Dienste.

Die entsprechenden Werkzeuge lassen sich jedoch auch von Hand zusammenstellen. Ein DevOps-Zyklus startet mit der Projekt-Planung und -Verwaltung. „Jira“ [3] ist ein gutes Beispiel für die Unterstützung agiler Entwicklung. Für das Code- und Versions-Management setzt man heute meist auf das von Linus Torwalds initiierte und verteilt nutzbare „Git“. Der automatische Bau des Kompilats und die Zusammenstellung der Bibliotheken erfolgt mit „Maven“, „Ant“, „Grunt“ oder „Gradle“. Für die Ablaufkontrolle und das Ausbringen der Softwarepakete verwendet man „Hudson“ oder „Jenkins“. Eine Test-Automatisierung kann mit „JUnit“ oder „Selenium“ erfolgen.

Bei aller Automatisierung kommt dieses Prinzip bei der Microservice-Architektur an seine Grenzen. Jeder Microservice benötigt neben dem funktionalen Code auch eine Sammlung von Laufzeit-Umgebungen und Bibliotheken. Er sollte isoliert betreibbar sein, was das parallele Ausbringen auf geteilte Laufzeitinstanzen praktisch ausschließt. Man benötigt mehr als nur die automatische Provisionierung einer Applikation. Es geht auch um die massenhafte und vollautomatische Provisionierung von mehr oder minder komplexen Laufzeitumgebungen und Infrastrukturen.

Container-basiertes DevOps für Microservices

Die Lösung für das Problem der isolierten Provisionierung von Microservices liegt in der Nutzung von Containern. Docker hat sich mittlerweile als Quasi-Standard für Container etabliert. Man packt alle benötigten Komponenten eines Microservice in einen gemeinsamen Docker-Container und betrachtet diesen als Ausbringungseinheit.

Für ein Container-basiertes DevOps sind drei Basis-Komponenten erforderlich: Erstens ein Werkzeug für die Erstellung und Verwaltung von Containergruppen; man spricht in diesem Zusammenhang auch von der Orchestrierung von Containern. Hier hat sich das ursprünglich von Google entwickelte und mittlerweile unter dem Dach der CNCF als Open-Source-Projekt weitergeführte Kubernetes (K8s) [4] gegenüber alternativen Ansätzen wie Docker Swarm oder Mesosphere durch-

gesetzt. Für die Ablage und Verteilung von Docker-Containern nutzt man private oder öffentliche Container Registries. Eine mit diesen integrierte Automatisierung der Erstellung und Ausbringung der Container ist die dritte Komponente. In Summe erhält man eine „Continuous Integration/Continuous Delivery“-Pipeline (CI/CD) für Microservices auf Basis von Docker-Containern.

Oracle Container Native Application Development Platform

Die Oracle Container Native Application Platform [5] bietet DevOps-Teams eine durchgängige Produktsammlung auf Open-Source-Basis, um Container-Cluster automatisiert aufsetzen und verwalten zu können. Die Container laufen virtualisiert, „Bare Metal“, also ohne Hypervisor, oder in Kombination auf Basis der derzeit modernsten Oracle-Cloud-Infrastruktur (OCI). Diese ist etwa in der Oracle-Rechenzentrumsregion Frankfurt am Main verfügbar. Sie bietet mit ihren dediziert zuweisbaren Ressourcen höchste Performance, Verfügbarkeit und Ausfallsicherheit für den professionellen Unternehmenseinsatz. Die Oracle-Container-Plattform (siehe Abbildung 1) deckt den kompletten Lebenszyklus von Microservices ab:

- **Oracle Container Pipelines**
Eine elegante, flexible und programmatische CI/CD Lösung, um Container-basierte Microservices schnell und wiederholbar auszuliefern

- **Oracle Container Registry (in Kürze verfügbar)**
Eine hochverfügbare, private Container Registry, um Container-Images zu speichern und zu verteilen
- **Oracle Container Engine (in Kürze verfügbar)**
Ein vollständig verwalteter Kubernetes-Dienst, der den Betrieb hochverfügbarer Cluster ermöglicht, ohne sich selbst um die Wartung und Administration von Kubernetes kümmern zu müssen

Oracle Container Pipelines

„Bringe deine Software frühzeitig aus und bringe sie oft aus“ ist das Motto der Microservice-Entwicklung. Die Oracle-Container-Plattform kann in diesem Sinne mit beliebigen CI/CD-Lösungen wie Jenkins oder Hudson genutzt werden. Container Pipelines sind die Container-zentrische CI/CD-Lösung von Oracle, basierend auf der Akquise von Wercker. Die Modellierung der Software-Auslieferung erfolgt hier in sogenannten „Workflow Pipelines“. Ein Workflow startet automatisch, etwa mit einem „Git Push“. Wercker kompiliert daraufhin die benötigten Code-Basen und führt alle notwendigen Schritte aus, das komplette Microservice-Projekt zu bauen. Es nutzt isolierte Container für die Code-Ausführung und generiert als Ergebnis des Workflows ebenfalls einen Docker-Container. Dieser enthält alle benötigten Konfigurationen und Umgebungen, um eigenständig und isoliert lauffähig zu sein. Er lässt sich in Container Registries ablegen und auf ein Ziel der

Wahl, etwa die Oracle Container Engine, ausbringen.

Oracle Container Registry

Bei der Container Registry (in Kürze verfügbar) handelt es sich um einen privaten, Docker-Registry-V2-konformen Ablageort für Images in der Oracle-Cloud. Sie ist vollständig mit der Container Pipeline und der Container Engine der Plattform integriert. So lassen sich mit der Container Pipeline generierte Images als Arbeitsschritt direkt und mit sehr geringer Latenz, hochverfügbar und sicher ablegen. Man kann die Registry durch Nutzung der Docker-CLI auch eigenständig vom Entwickler-Laptop, dem eigenem Rechenzentrum oder Cloud-Diensten von Dritt-Anbietern aus nutzen. Als Anwender bezahlt man übrigens nicht die Registry selbst, sondern nur die genutzten IaaS-Ressourcen der Oracle-Cloud.

Oracle Container Engine

Das Open-Source-Werkzeug Kubernetes [4] automatisiert den Umgang mit Docker-Clustern vollständig. Dazu gehört die automatisierte Ausbringung, Skalierung, Lastverteilung und Speichernutzung. Das Werkzeug ist allerdings entsprechend seinen Aufgaben komplex in der Handhabung und bei einer Installation von Hand selbst zu betreiben und zu warten. Aus diesem Grund bietet Oracle ein über das CNCF-Konformitätsprogramm zertifiziertes Kubernetes als vollständig verwalteten und hochverfügbaren Dienst integ-

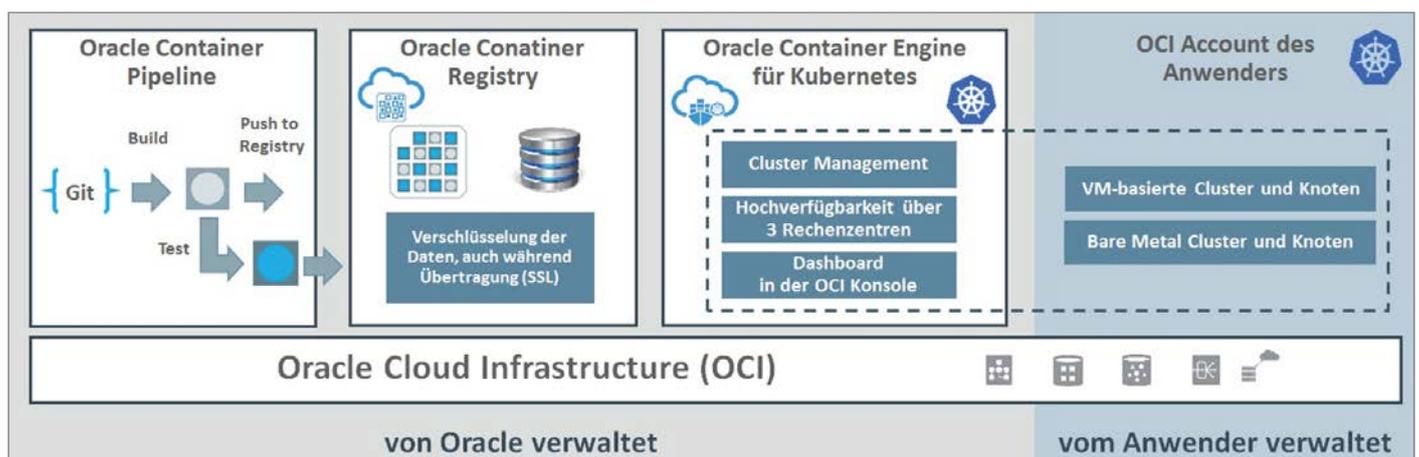


Abbildung 1: Die Oracle-Container-Plattform

riert mit der Oracle-Cloud-Infrastruktur (OCI). Man muss sich also nicht selbst um die Wartung der Kubernetes-Infrastruktur kümmern. Auch hier zahlt man als Anwender nicht den Kubernetes-Dienst selbst, sondern wiederum nur die genutzten IaaS-Ressourcen (wie Compute, Storage und Load Balancer).

Die von der Oracle Container Engine (in Kürze verfügbar) verwalteten Kubernetes-Cluster lassen sich flexibel erstellen und orchestrieren; entweder direkt aus der Container Pipeline heraus, über ein REST-API oder über das Benutzer-Interface der Oracle-Cloud. Die einzelnen Knoten sind per SSH erreichbar, damit man bei Bedarf weiterhin die gewohnten Docker-Werkzeuge nutzen kann. Die Zugriffsrechte des ganzen Teams lassen sich von der Entwicklung bis in die Produktion zentral verwalten. Die Vorteile der zugrunde liegenden Oracle-Cloud-Infrastruktur kommen voll zum Tragen: Man kann auf Knopfdruck Bare-Metal-Cluster mit voller vorhersagbarer Leistung über mehrere Verfügbarkeits-Domänen, also drei unabhängige Rechenzentren in einer Region, spannen.

Serverless Computing mit Fn Project

Als die nächste Evolutionsstufe der modernen Applikationsentwicklung wird das Serverless Computing betrachtet. Man macht sich keine Gedanken mehr über Server, Storage, Netzwerk, Virtualisierung und andere Infrastruktur-Komponenten, sondern konzentriert sich ganz auf den eigenen Code, also die Umsetzung der Geschäftsanforderung in Form eines kleinen Funktionspakets (Function-as-a-Service, FaaS). Server gibt es beim Serverless Computing natürlich dennoch. Sie brauchen allerdings keinerlei Administration und bleiben für den Entwickler unsichtbar.

Mit AWS Lambda [6], Cloud Functions [7] und anderen gibt es schon einige kommerzielle, weitgehend geschlossene Serverless-Computing-Angebote am Markt. Mit Fn Project [8] hat Oracle eine Open-Source-Serverless-Plattform als Alternative initiiert. Im Gegensatz zu den vorgenannten Plattformen nutzt man bei Fn Project nicht nur ein API oder eine Web-Oberfläche, um Code hochzuladen.

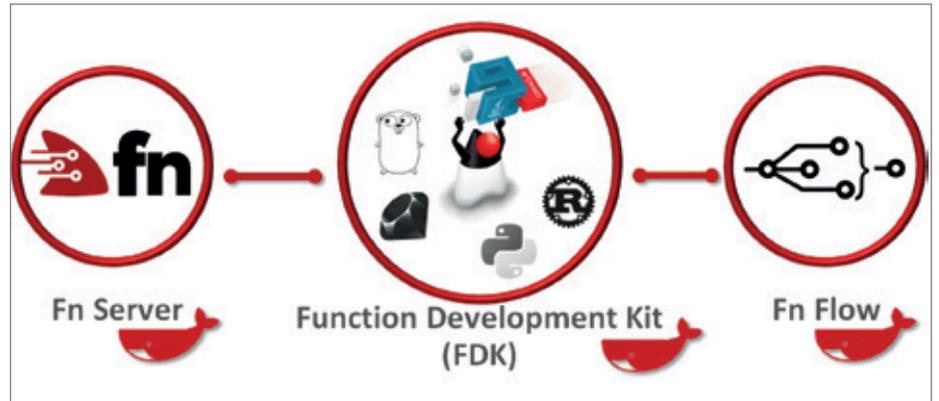


Abbildung 2: Der Aufbau von Fn Project

Fn Project ist komplett Container-basiert (siehe Abbildung 2). Die Funktion ist hier der Container und der Container die Funktion, Docker die einzige Abhängigkeit. Selbst der Fn Server, die Laufzeit-Umgebung von Fn Project, ist ein Docker-Container. Dadurch ist er unverändert überall lauffähig: auf dem Laptop des Entwicklers, auf jeder beliebigen Container-Plattform im eigenen Rechenzentrum oder als Kubernetes-Cluster in der Oracle-Cloud-Infrastruktur (OCI).

Der Fokus von Fn Project liegt auf der einfachen und flexiblen Nutzbarkeit. Es ist polyglott, Funktionen können also in jeder beliebigen Programmiersprache erstellt werden. Es gibt bereits vorgefertigte Entwicklungsbaukästen (FDK, Function Development Kit) für Java, Go, JavaScript, Ruby, Rust und Python. Man entwickelt von der Kommandozeile aus mit dem Kommando „fn“. Funktionen werden als Docker-Container gepackt, die auch die Ausbringungseinheit sind. Mit einem vorgefertigten Helm-Chart lässt sich Fn Project als Kubernetes-Cluster automatisiert in der Oracle-Cloud aufsetzen und passt sich damit bestens in die Oracle-Container-Plattform ein.

Mit Fn Flow lassen sich einzelne Funktionen zu komplexen Diensten zusammensetzen. Dabei werden Aspekte wie Parallelisierung, Ausführungs-Reihenfolge, Wiederanläufe und Fehlerbehandlung adressiert. Es ist im Vergleich zu einem klassischen Service-Bus oder einer SOA-Orchestrierung ein Leichtgewicht. Abläufe werden über ein asynchrones und verteiltes Programmiermodell im Code definiert. Ausgeführt wird Fn Flow ebenfalls als Funktion.

Hinweis: Oracle Container Registry und Oracle Container Engine sind in Kürze verfügbar, Stand 04/18

Weitere Informationen

- [1] <https://cloud.oracle.com/tryit>
- [2] <https://cloud.oracle.com/developer-service>
- [3] <https://www.atlassian.com/software/jira>
- [4] <https://kubernetes.io>
- [5] <https://cloud.oracle.com/containers>
- [6] <https://aws.amazon.com/lambda>
- [7] <https://cloud.google.com/serverless/functions>
- [8] <https://fnproject.io>



Marcel Amende
marcel.amende@oracle.com

Objekte machen das Leben leichter

Jürgen Sieben

In dieser Folge möchte der Autor eine Lanze für die recht selten genutzte Objektorientierung in PL/SQL brechen.

Ein erstes Beispiel wurde bereits verwendet, um mithilfe einer „NESTED TABLE“ beliebig viele Parameter an eine „BULK REPLACE“-Prozedur zu übergeben (siehe Red Stack Magazin, Ausgabe 06/2017). Schon dort war die Tatsache angenehm aufgefallen, dass Objekte gleichermaßen in SQL und PL/SQL verwendet werden können. Diesen Ansatz soll ein wenig weitergedacht werden.

Oracle unterstützt drei Typen von Objekten: das eigentliche „OBJECT“, das einem Record ähnelt, und die beiden Kollektionstypen „VARRAY“ (variabel langes Array, dem eine maximale Anzahl von Elementen mitgegeben wird und die eine garantierte Sortierung hat) und die unbegrenzte, aber auch nicht sicher sortierte „NESTED TABLE“.

Ein „OBJECT“ kann immer auch als Ersatz für einen Record verwendet werden, wenn man ausschließlich Attribute im Objekt festlegt und einen Standard-Konstruktor verwendet, um ein Objekt zu erstellen: Ähnlich wie die Funktion „TO_DATE“ aus einer Zeichenkette eine Instanz vom Typ „DATE“ erstellt, erzeugt ein Konstruktor eine Instanz des Objekts. Der Standard-Konstruktor heißt genauso wie der Typ, den er erstellen soll, und erwartet, dass alle Attribute des Objekts mit Werten belegt werden. Hiervon kann man jedoch auch abweichen und eigene Konstruktor-Methoden erzeugen, etwa um nur einige Parameter zu übergeben und die Attributwerte aus diesen Parametern zu ermitteln.

Ein Beispiel: Ein Logging-Package soll in der Lage sein, Meldungen durch eine beliebige Anzahl von Parametern zu spezialisieren; zudem soll die Meldung in der Sprach-Einstellung der Session erzeugt werden. Dies kann natürlich auch über

herkömmlichen Code erfolgen, alternativ wäre es allerdings auch möglich, einem Objekt vom Typ „MESSAGE_TYPE“ dieses Verhalten beizubringen. *Listing 1* zeigt ein solches Objekt, man erkennt die Konstruktor-Methode, die für die Instanziierung dieses Objekts verwendet werden soll.

Beim genauen Betrachten ist ein Attribut „P_ARG_LIST“ vom Typ „MSG_ARGS“ zu erkennen. „MSG_ARGS“ ist ein Kollektionstyp, allerdings keine „NESTED TABLE“, sondern ein variables Array („VARRAY“) mit bis zu zwanzig Einträgen. In einem Objekt können also weitere Objekt-Typen als Attribut hinterlegt sein. Die Konstruktor-Methode übernimmt nun die Aufbereitung der Nachricht in der benötigten Sprache und liefert eine Instanz der Meldung zurück, die nicht nur aus dem Meldungstext, sondern auch aus flankierenden Informationen besteht, die in der weiteren Verarbeitung benötigt werden. Umgesetzt sind diese Konstruktor-Methoden als Me-

thoden mit der Klausel „CONSTRUCTOR FUNCTION“, die „SELF“ als Ein- und Ausgabe-Parameter besitzen. Das ist die Objekt-Instanz, die über die Methode mit der Anweisung „RETURN“ später zurückgegeben werden wird.

Bevor wir den Code zur Erzeugung eines Objekts betrachten, zunächst die Fragen: „Wozu das alles?“, „Warum erstellen wir eine Meldung nicht mit normalem Code?“. Der wichtigste Aspekt ist die Kapselung der Logik. Wie eine Meldung konkret gebaut wird, weiß der Datentyp selbst. Sobald das Objekt kompiliert ist, lässt es sich wie ein eingebauter Datentyp verwenden, die zugehörige Logik gibt es unmittelbar dazu. Alternativ wäre die Erstellung einer Meldung nur über ein Package möglich, doch dann hätte man noch einen Plan entwickeln müssen, wie die weiteren Attribute bei der Meldung gespeichert sind. Diese Alternativlösung hätte einen Record deklariert und diesen in einem Hilfspackage mit Werten gefüllt. Das kann ein Objekt

```
create or replace type message_type force is object(  
  id number,  
  message_name varchar2(30 char),  
  message_text clob,  
  severity number(2,0),  
  stack varchar2(2000),  
  backtrace varchar2(2000),  
  error_number number(5,0),  
  message_args msg_args,  
  -- Konstruktor  
  constructor function message_type(  
    self in out nocopy message_type,  
    p_message_name in varchar2,  
    p_arg_list msg_args)  
    return self as result)  
;
```

Listing 1

```

create or replace type body message_type
as
  constructor function message_type(
    self in out nocopy message_type,
    p_message_name in varchar2,
    p_arg_list msg_args)
    return self as result
  as
    l_errm varchar2(2000);
begin
  -- Meldungstext aus Meldungstabelle lesen und Sprache festlegen
  select pms_pml_name,
         pms_text,
         pms_pse_id,
         coalesce(pms_active_error, pms_custom_error) error_number
    from pit_message m
   join pit_message_language_v l -- View liefert die Sessionsprache
     on m.pms_pml_name = l.pml_name
   where m.pms_name = p_message_name
   order by l.pml_default_order desc
   fetch first 1 row only;

  -- Objektattribute belegen
  self.id := pit_log_seq.nextval;
  self.message_name := p_message_name;
  self.message_args := p_arg_list;

  -- Im Fehlerfall: Fehlerdetails speichern
  if sqlcode > 0 and self.severity <= 30 then
    self.stack := dbms_utility.format_error_stack;
    self.backtrace := dbms_utility.format_error_backtrace;
  end if;

  -- Platzhalter der Meldung durch Variablen ersetzen (bulk replace)
  if p_arg_list is not null then
    for i in p_arg_list.first..p_arg_list.last loop
      self.message_text :=
        replace(self.message_text, '#' || i || '#', p_arg_list(i));
    end loop;
  end if;
  return;
end;
end;
Nun kann das Objekt sehr einfach verwendet werden:
declare
  l_msg message_type;
begin
  l_msg := message_type('MY_MESSAGE', msg_args('FOO', 'ACME'));
end;

```

Listing 2

direkt und vor allem: auch in SQL. Selbst wenn man einen Record erstellt hätte, wäre dieser nicht in SQL verwendbar und könnte auch nicht in einer Tabelle gespeichert werden. Dies wäre für Meldungen aber durchaus praktisch, zum Beispiel weil nun ein Log-Eintrag auch im Nachhinein in unterschiedlichen Sprachen aufbereitet werden kann, und das geht eben nur mit Objekten.

Listing 2 zeigt die Implementierung des Objekt-Typs, die ähnlich einem Package

im „TYPE BODY“ erfolgt. Über den Parameter „SELF“ werden die Objekt-Attribute belegt, und dieses Objekt liefert zum Ende der Methode über die Anweisung „RETURN“ zurück. In unserem Fall wird die Meldung aus einer Meldungstabelle ermittelt und die Spracheinstellung dynamisch aus einer Voreinstellung und der Session-Sprache abgeleitet. Bei aufmerksamem Studium des Codes ist die „BULK REPLACE“-Methode wiederzuerkennen, die hier alle übergebenen Parameter im

Meldungstext auf Meldungsanker verteilt.

Fazit

Objekte können ein Ersatz für Records sein, haben diesen aber einen Standard-Konstruktor voraus, der das Belegen der Attribute einfacher macht als bei einem Record, zudem können Objekte in SQL und PL/SQL genutzt werden. Wird mehr Logik benötigt, um eine Instanz des Objekts zu erstellen, können eigene (und auch beliebig viele) Konstruktor-Methoden erzeugt werden, um die für die Erstellung benötigte Logik beim Objekt zu halten und dort zu kapseln. Ein Package ist im Regelfall nicht erforderlich.

Die Möglichkeit, Logik in Objekten zu abstrahieren, ist erfahrungsgemäß für viele PL/SQL-Entwickler eine ungewohnte Lösung, doch der Autor hofft, durch dieses einfache Beispiel einige der Vorteile dieser Lösung aufgezeigt zu haben. Es lohnt sich, sich an Objekte heranzutrauen.



Jürgen Sieben
j.sieben@condes.de



Verarbeitung von Streamingdaten – ein Erfahrungsbericht der Testphase mit Oracle NoSQL

Marcus Bender, Rainer Marekwia, Thomas Robert und Dr. Nadine Schöne,
ORACLE Deutschland B.V. & Co. KG

Streamingdaten sind Daten, die als nahezu kontinuierlicher Datenstrom vorliegen und aus einer Vielzahl von Datenquellen stammen können. In diesem Artikel stellen wir zwei Möglichkeiten vor, diese in der Oracle-NoSQL-Datenbank zur schnellen Verarbeitung zu speichern. Die verwendeten Komponenten (Kafka, Flume und Oracle-NoSQL-Datenbank) wurden dazu auf dem Oracle Big Data Cloud Service installiert. Die Autoren gehen auch kurz auf die Eigenschaften der verschiedenen Komponenten ein und verlinken den verwendeten Code.

Streamingdaten stammen meist aus einer Vielzahl von Datenquellen. Sie liegen als nahezu kontinuierlicher Datenstrom vor. In der Regel handelt es sich um viele kleine Datenpakete; Beispiele sind Tweets, Log- oder Sensordaten. Die Auswertung solcher Daten wird für viele Unternehmen immer wichtiger; eine Analyse in nahezu Real-Time erfolgt durch direkte Beobachtung des Datenstroms. Dagegen werden bei einer klassischen Batch-

Analyse die Daten erst abgespeichert – entweder in einer relationalen Datenbank nach Anwendung eines Datenschemas (Schema-on-Read) oder in HDFS (gewöhnlich Schema-on-Write) – und danach zeitversetzt analysiert.

Wie geht man aber am besten vor, wenn man aktuelle Daten möglichst schnell analysieren will, zum Beispiel, wenn man einen Überblick über den letzten Fertigungslauf erhalten oder Kundereaktionen auf

Twitter direkt nach Launch einer neuen Produktlinie untersuchen möchte?

Für solche Batchanalysen von aktuellen Daten bieten sich NoSQL-Datenbanken an (NoSQL = „Not Only SQL“). Beliebige Daten lassen sich schnell ablegen. Zugriffe und Abfragen sind performant und einfach, und sie sind kostengünstig skalierbar. Verschiedene NoSQL-Datenbanken sind spezialisiert auf unterschiedliche Zwecke und Daten wie Dokumente,

Messdaten in XML-Format

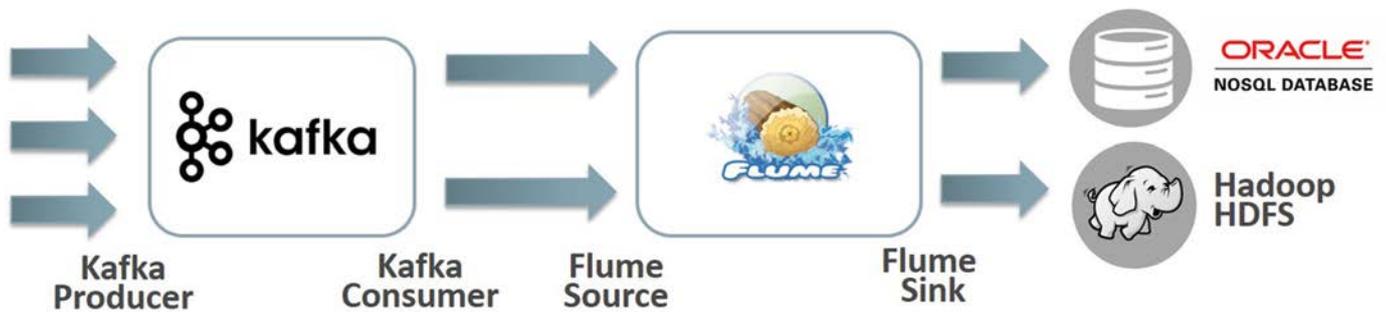


Abbildung 1: Aufgabenstellung ist die Transformation und das Ablegen von Sensordaten in einer Oracle-NoSQL-Datenbank (und gegebenenfalls auch in HDFS) unter Nutzung von Kafka und Flume

XML- oder JSON-Daten, Key-Value-Inhalte oder spaltenbasierte Daten.

Relationale Datenbanken schöpfen hingegen ihr Potenzial am besten aus, wenn bereits ein Datenschema vorliegt. Für den Zugriff auf strukturierte Daten bei gleichzeitigen Schreib- und Lesevorgängen, hoher Concurrency sowie Anforderungen an Transaktionssicherheit und Security sind relationale Datenbanken in der Regel ausgereifter als NoSQL-Datenbanken.

Das Speichern auf HDFS ist einfach umsetzbar. Nahezu alle Big-Data-Komponenten unterstützen HDFS nativ – auch mit optimierter Abspeicherung wie beispielsweise Parquet oder Avro. Abfragen sowie Batch- und Massen-Datenverarbeitung sind in der Regel sehr performant. HDFS ist aber nicht gut für die Verarbeitung vieler kleiner Dateien, sondern eher für größere Dateien geeignet. Auch eine Veränderung bereits gespeicherter Daten („Update“) ist nicht vorgesehen: Das Lesen einzelner Sätze aus großen Datenbeständen ist nicht performant und Veränderungen sind in der Regel nur durch Löschen und Einfügen möglich, was zu langen Antwortzeiten führen kann. Damit ist HDFS nicht ausgelegt für eine direkte und schnelle Verarbeitung. Um die Daten im Rohzustand für spätere Batchanalysen vorzuhalten, bietet sich aber das parallele Abspeichern der Daten in HDFS an.

Aufgabenstellung

Ziel ist die Transformation und das Ablegen von Sensordaten in einer Oracle-NoSQL-Datenbank (und gegebenenfalls auch in HDFS, siehe Abbildung 1). Hierbei sollen die Daten aus dem Datenstrom

über Flume in das NoSQL-Zielsystem übertragen werden. Der Zugriff auf die gespeicherten Daten soll möglichst wahlfrei über beliebige Schlüsselwerte möglich sein.

Datenformate

Streamingdaten können in verschiedenen Formaten vorliegen, etwa als Message-, Text-, XML- oder JSON-Format. Die in diesen Tests verwendeten Daten liegen als XML-Dateien vor. Um die Daten von ihrer Quelle zum Zielort zu bewegen, ist ein Message Broker erforderlich. Die Autoren haben sich hier für Apache Kafka entschieden.

Als Plattform wurde der Oracle Big Data Cloud Service genutzt. Dieser fertig installierte und vorkonfigurierte Cloud Service stellt auf Basis der Cloudera-Distribution bereits viele Komponenten zur Verfügung. Vorhandene Komponenten wie HDFS, Kafka und Flume sind direkt nutzbar, die Oracle-NoSQL-Datenbank lässt sich über eine einfache Installationsprozedur nachinstallieren. *Abbildung 2* zeigt eine Übersicht über die genutzten Komponenten.

Apache Kafka ist ein Broker, also ein verteiltes („distributed“) Messaging-System, das für zuverlässige Message-Verwaltung, hohen Datendurchsatz, geringe Latenz und hohe Skalierbarkeit ausgelegt ist. Message-Feeds werden in Topics vorgehalten: Daten werden von Produzern in Topics geschrieben und von Consumern aus Topics konsumiert. Weil Kafka ein verteiltes System ist, werden diese Topics partitioniert und auf mehrere Knoten repliziert. Die redundante Speicherung stellt Hochverfügbarkeit und Skalierbarkeit sicher. Hier kommt ein Replikations-

faktor von drei zum Einsatz, es können also bis zu zwei Knoten ausfallen, ohne dass Daten verloren gehen.

Kafka ist ein Multi-Purpose-System: Viele Producer und viele Consumer können viele Topics gleichzeitig nutzen. Damit ist Kafka sehr gut geeignet, wenn Daten von verschiedenen Applikationen verarbeitet werden sollen. Messages sind einfache Byte-Arrays, in denen Objekte in beliebigem Format gespeichert werden. Sie können mit einem Key versehen sein, sodass alle Messages mit demselben Key in dieselbe Partition geschrieben werden. Jede Topic-Partition wird als Log (geordnete Menge von Messages) behandelt. Messages werden für eine bestimmte Zeit vorgehalten, unabhängig davon, ob sie gelesen wurden oder nicht.

Kafka ist unabhängig von anderer Software, mit der Ausnahme von Apache ZooKeeper, das als übergeordnete Instanz für verteilte Ressourcen genutzt wird.

Flume ist ein Framework für das zuverlässige Sammeln, Aggregieren und Bewegen einer großen Anzahl verteilter Dateien. Ein Flume-Agent hat je mindestens eine Source, einen Channel und einen Sink. Die Source empfängt Daten und sendet diese an einen oder mehrere Channels. Ein Channel ist eine Datenqueue, die als Buffer für die Flüsse von Input und Output eines oder mehrerer Sources fungiert. Der Sink liefert die Daten eines einzelnen Channels an ein Ziel wie HDFS oder einen anderen Flume-Agenten. Daten verbleiben solange im Channel, bis ein Sink sie verarbeitet.

Es gibt fertige Schnittstellen, sodass unterschiedliche Sources und Sinks einfach angebunden werden können. Flume repliziert keine Events – wenn der Knoten mit



Abbildung 2: Übersicht der genutzten Komponenten – Apache Flume, Apache Kafka, Oracle NoSQL und Oracle Big Data Cloud Service

dem Flume-Agenten abstürzt, verliert man erst einmal den Zugriff auf die Events im Channel. Durch die Verteilung von mehreren Agenten über einen Cluster erreicht man horizontale Skalierbarkeit.

Flume enthält bereits Schnittstellen für Kafka und kann sowohl als Consumer als auch als Producer dafür fungieren. Flume liefert in dieser Kombination vorgefertigte Schnittstellen, Kafka garantiert Stabilität: Sobald die Daten in Kafka residieren, können sie nicht mehr verloren gehen. Die Nutzung eines Flume-Agenten zusammen mit Kafka wird auch als „Flafka“ bezeichnet.

Der erste Schritt beim Aufbau einer Streamingdaten-Pipeline ist die Überlegung, wie die Daten am besten abgespeichert werden sollten, damit eine optimierte, schnelle Analyse erfolgen kann. Um einzelne Sätze aus großen Datenbeständen nachträglich zu verändern oder schnell zu lesen, eignen sich bestimmte NoSQL-Datenbanken wie Oracle NoSQL oder Apache HBase hervorragend. Die Autoren haben sich für die Transformation und das Ablegen von Sensordaten in einer Oracle-NoSQL-Datenbank entschieden. Oracle NoSQL ist ein verteilter Key-Value Store, mit Unterstützung von transaktionaler Verarbeitung (Create, Read, Update & Delete) und schnellem Zugriff auf einzelne Sätze über Keys (siehe Abbildung 3).

Der Artikel beschränkt sich auf eine Speicherung in der NoSQL-Datenbank, einmal als Key-Value-Store und einmal über das Table-API. Daten sind in Oracle NoSQL typischerweise mit einem Repli-

kationsfaktor von drei vorgehalten. Die Replikation auch über Datacenter hinweg liefert Hochverfügbarkeit. Es wird auch von den Replikaten gelesen (Load Balancing) und Treiber berücksichtigen Netzwerk-Latenzen. Durch die dynamische Partitionierung und Verteilung sind Hochverfügbarkeit und Skalierbarkeit garantiert. Konsistenz und Persistenz sind konfigurierbar (ACID <-> BASE).

Das Datenmodell als Key-Value-Store besteht aus einem obligatorischen Major (primären) Key und einem fakultativen Minor (sekundären) Key sowie den zugeordneten Werten (Values). Diese Ansammlung von Key-Value-Paaren wird auf Partitionen verteilt gespeichert. Gleiche Major Keys werden gemeinsam abgelegt (Daten-Kollokation). Einfache Zugriffe sind über GET, PUT und DELETE möglich. Die Identifizierung der Values erfolgt über Teile des Keys, der Value wird ausgelesen.

Oracle NoSQL bietet auch die Möglichkeit, sogenannte „Key-Paths“ zu definieren, in denen der Schlüssel aus einem Pfad von Teilschlüsseln zusammengesetzt ist. In unserem Fall sollte dieser Pfad aus den drei Attributen „Ort“, „Maschinen-ID“ und „Material“ zusammengesetzt

werden, die aus den XML-Dateien zu extrahieren sind. Ein Beispiel für einen solchen Key-Path ist „/HH/4711/Blech“. Es gibt keine Standard-Abfragesprache, aber eine Verwendung von SQL ist durch Nutzung von Oracle-Big-Data-SQL möglich.

Die Oracle-NoSQL-Datenbank erlaubt über das Table-API den Zugriff auf das Table-Interface des Datenstores. Tables sind Tabellen-Abstraktionen mit Zeilen und Spalten und ähneln damit Tabellen in relationalen Datenbanken. Tables haben einen wohldefinierten Primary Key. Das Table-Interface ermöglicht darüber hinaus die Nutzung von sekundären Indizes, die über separate Strukturen den Indexwerten die entsprechenden Primary Keys zuordnen und so schnellen Zugriff auf die Values ermöglichen. Unterstützte Datentypen sind zum Beispiel Integer, String, Date, Array, Map und Record.

Nutzung und Installation der Komponenten

Alle Komponenten sind Bestandteil des Oracle Big Data Cloud Service mit Ausnahme der NoSQL-Datenbank, die nach-

```
<cfg>.channels.channel-kafka.producer.max.request.size=18000000
<cfg>.channels.channel-kafka.producer.max.partition.fetch.bytes=18000000
<cfg>.sources.src-spooldir.deserializer.maxLineLength=23000000
```

Listing 1

installiert wurde. Der Oracle Big Data Cloud Service ist in diesem Fall ein Cluster mit neun Knoten und insgesamt 306 Cores, 2,3 TB Memory und 864 TB Plattenplatz. Dahinter steckt eine Oracle Big Data Appliance, also ein Engineered System mit Oracle Linux, der Cloudera-Distribution, die Apache-Hadoop-Komponenten beinhaltet, und Oracle-Software wie Advanced Analytics For Hadoop (ORAAH).

Flume enthält fertige Schnittstellen zur Datenspeicherung im Hadoop-Ökosystem, jedoch nicht für Oracle NoSQL. Um die NoSQL-Datenbank als Sink für Flume-Events nutzen zu können, wurde ein Code-Beispiel auf GitHub (siehe „<https://github.com/gvenzl/FlumeKVStoreIntegration>“) genutzt. Es erlaubt jedoch keine komplexeren Key-Strukturen, da es alle Sonderzeichen aus dem Schlüssel eliminiert. Deshalb wurde das Git-Projekt dupliziert und für die benötigten Zwecke angepasst.

Als Value sollte die komplette XML-Datei verwendet werden. Da einige der Dateien in diesem Fall bis zu 11 MB groß waren, war die Konfiguration des Kafka-Channels anpassen, um eine fehlerfreie Übertragung der Daten zu gewährleisten. Hier war insbesondere die Anpassung der folgenden Parameter auf die hier angegebenen Werte notwendig (siehe Listing 1).

Ergebnisse

Kafka, Flume, Oracle NoSQL (und gegebenenfalls HDFS) bieten eine performante, skalierbare Möglichkeit, Messages in der Big-Data-Welt zu verarbeiten und später abzufragen. Kafka und Flume sind Software-Lösungen, mit denen man relativ schnell einfache Anforderungen umsetzen kann. Kafka ist eine sehr robuste, vielseitig einsetzbare Big-Data-Komponente, die über das Topic-Konzept einen Datenstrom auch in verschiedene Verarbeitungsstränge laufen lassen kann. So werden beispielsweise XML-Dateien original in HDFS-Parquet-Files archiviert und parallel können dieselben Daten in einem zweiten Strom transformiert und weiterverarbeitet werden. Dieses Konzept wird aus Sicht der Autoren viel Verbreitung finden.

Flume stellt fertige Schnittstellen mit bestimmten Funktionalitäten bereit. Aufwendigere Transformationen sind jedoch allein über Kafka und Flume (Flafka) nicht abbildbar. Dafür müssen wie in diesem Fall die

Schnittstellen angepasst beziehungsweise Transformationen in Spark, Python oder Java eingebunden werden. Für die Verarbeitung einzelner Datensätze oder geringer Datenmengen aus großen Datenbeständen eignen sich NoSQL-Datenbanken wie Oracle NoSQL sehr gut. In diesen Tests hat man sich auf Oracle NoSQL konzentriert.

Es ist einfach zu installieren, skalierbar, hochverfügbar und für selektive Anfragen sehr performant. Es bietet originär über Major Keys und Minor Keys guten Zugriff auf bestimmte selektive Datenbestände. Über das Oracle-NoSQL-Table-API in Kombination mit Sekundär-Indizes werden vielfältige Selektionen von einzelnen Datensätzen bis hin zu kleineren Datenmengen unterstützt. Für komplexere Anforderungen muss man jedoch relativ viel Zeit einplanen, da die Umsetzung in der Big-Data-Welt mit Programmieraufwand, Integration in Entwicklungsumgebungen wie Maven und der Kontrolle der Kompatibilität der Softwareversionen unterschiedlicher Komponenten verbunden ist.

Mit dem angepassten Code lassen sich dann jedoch die Dateien (hier: XML-Dateien) hochperformant in der NoSQL-Datenbank ablegen. Die definierte Key-Struktur ermöglicht eine einfache Suche in dem Key-Value Store auch nach Teilschlüsseln. Allerdings müssen bei dem Key-Value-API immer die führenden Schlüsselwerte bei der Suche spezifiziert werden. In diesem Beispiel, wenn der Key aus dem Pfad „/<Ort>/<Maschinen-ID>/<Material>“ besteht, kann zwar nach den Werten aller Maschinen eines Ortes ge-

sucht werden, aber nicht nach allen Maschinen für ein bestimmtes Material.

Bei Verwendung des Oracle-NoSQL-Table-API kann über beliebige Teilschlüssel gesucht werden. Hier wurden dann neben dem unveränderten Primary Key die beiden Secondary Keys „Material“ und „Maschinen-ID/Material“ angelegt. Diese drei Keys ermöglichen nun die Suche über eine beliebige Schlüssel-Kombination. Alle Suchoperationen sind dabei hochperformant und erlauben einen einfachen Zugriff auf beliebige Teilmengen der gespeicherten XML-Dateien, solange sich die gesuchte Teilmenge über die Schlüsselwerte eingrenzen lässt.

Fazit

Die vorgestellten Lösungsansätze sind gut geeignet für alle Anwendungen, die eine schnelle Auswertung von Daten aus einem begrenzten Zeitraum erfordern. Kafka, Flume und Oracle NoSQL (und gegebenenfalls HDFS) bieten in Kombination eine performante, skalierbare Möglichkeit, Messages in der Big-Data-Welt zu verarbeiten und später abzufragen.

Diese Kombination liefert bereits einen guten Funktionsumfang und deckt viele Bedarfe ab. Bei komplexeren Anforderungen wie in unserem Testbeispiel wird man aber um Programmieraufwand nicht herumkommen. Dies ist nach Erfahrung der Autoren mit hohem initialen Zeitaufwand verbunden.



Marcus Bender
marcus.bender@oracle.com

Thomas Robert
thomas.robert@oracle.com

Dr. Nadine Schöne
nadine.schoene@oracle.com

Rainer Marekwia
rainer.marekwia@oracle.com



Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- > Peter von Zimmermann
- > Sven Kostka
- > Deniz Yalman
- > Joachim Stahlberg
- > Martin Schrader
- > Olaf Nowatzki
- > Nikhil Wani
- > Hala Algendy
- > Nicole Schneiderreit
- > Daniel Schreiber
- > Joachim Jansen
- > Detlef Wildenhoff
- > Nebojsa Petrovic
- > Iradj Atchatchloui
- > Yoba Francis

Firmenmitglieder DOAG

- > Viktor Hinterleitner, HiSolutions AG



Termine



Juli

05.07.2018

Regionaltreffen NRW

Martin Schmitter
Dortmund

12.07.2018

Regionaltreffen Stuttgart

Jens-Uwe Petersen & Anja Stollberg
Stuttgart

13.07.2018

DOAG Datenbank Webinar: Pluggable Database

online

18.07.2018

Regionaltreffen München/Südbayern

Andreas Ströbel
München

19.07.2018

Regionaltreffen Nürnberg/Franken

Martin Klier & Thomas Köppel
Nürnberg



August

10.08.2018

DOAG Datenbank Webinar: Oracle Database 18c – New Performance Features

Christian Antognini
online

21.08.2018

Regionaltreffen Bremen

Ralf Kölling
Bremen

30.08.2018

Regionaltreffen Karlsruhe

Reiner Bünger
Karlsruhe



September

03.09.2018

Regionaltreffen Osnabrück/Bielefeld/Münster

Andreas Kother & Klaus Günther

06.09.2018

Regionaltreffen NRW

Martin Schmitter
Düsseldorf

10.09.2018

Regionaltreffen Thüringen

Jörg Hildebrandt
Weimar

11.09.2018

Regionaltreffen Hannover

Andreas Ellerhoff
Hannover

12.09.2018

Regionaltreffen Berlin/Brandenburg

Michael Keemers & Mylène Diacquenod

13.09.2018

HA Day Hochverfügbarkeit

Martin Klier & Jürgen Vitek
Hamburg

14.09.2018

DOAG Datenbank Webinar

online

17.09.2018

Regionaltreffen Halle/Leipzig

Matthias Reimann

20.09.2018

DOAG 2018 Big Data Days: Daten, der Treibstoff der digitalen Gesellschaft

Dresden

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Stefan Kinnen. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
Chefredakteur (ViSdP): Wolfgang Taschner
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Lisa Damerow, Mylène Diacquenod, Marina Fischer, Klaus-Michael Hatzinger, Sanela Lukavica, Yann Neuhaus, Fried Saacke

Titel, Gestaltung und Satz:

Alexander Kermas, DOAG Dienstleistungen GmbH (Anschrift s.o.)

Fotonachweis:

Titel: © jossdiim /123RF
S. 11: © everythingpossible/123RF
S. 13: © deniskot/123RF
S. 18: © Pavlo Vakhrushev/123RF
S. 23: © Jorgen McLeman/123RF
S. 29: © bluedesign/fotolia
S. 37: © microone/123RF
S. 41: © Natalia Merzlyakova/123RF
S. 47: © Alexander Kharchenko/123RF
S. 50: © Dzianis Rakhuba/123RF
S. 55: © macrovector/123RF
S. 63: © mertsalov/123RF
S. 69: © Kheng Ho Toh/123RF
S. 73: © apinan/123RF
S. 73: © Dzianis Kuryanovich/123RF
S. 73: Designed by Katemangostar/Freepik

Anzeigen:

Simone Fischer, DOAG Dienstleistungen GmbH (verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org
Mediadaten und Preise unter:
www.doag.org/go/mediadaten

Druck:

adame Advertising and Media GmbH,
www.adame.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

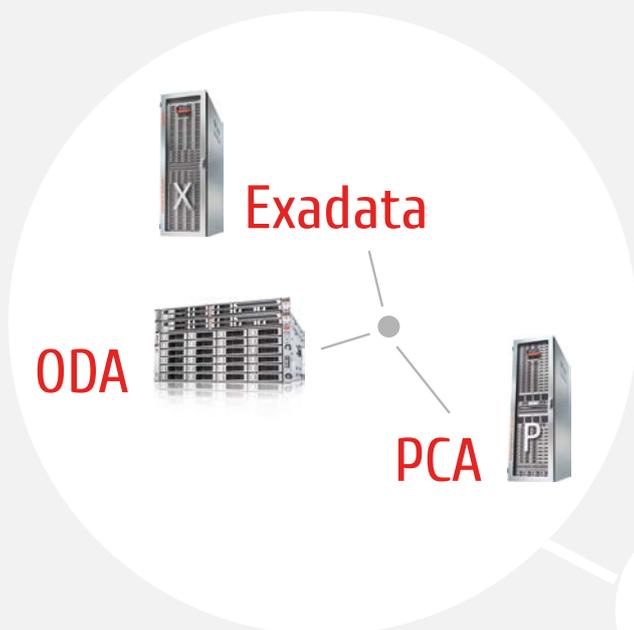
Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

dbi services sa www.dbi-services.com	S. 17	ORACLE Deutschland B.V. & Co. KG. www.oracle.com	U 2	Robotron Datenbank- Software GmbH www.promatis.de	U 3
DOAG e.V. www.doag.org	S. 39, 46	PROMATIS software GmbH www.promatis.de	S. 19	Trivadis AG www.trivadis.com	U 4
MuniQsoft GmbH www.muniqsoft.de	S. 3				

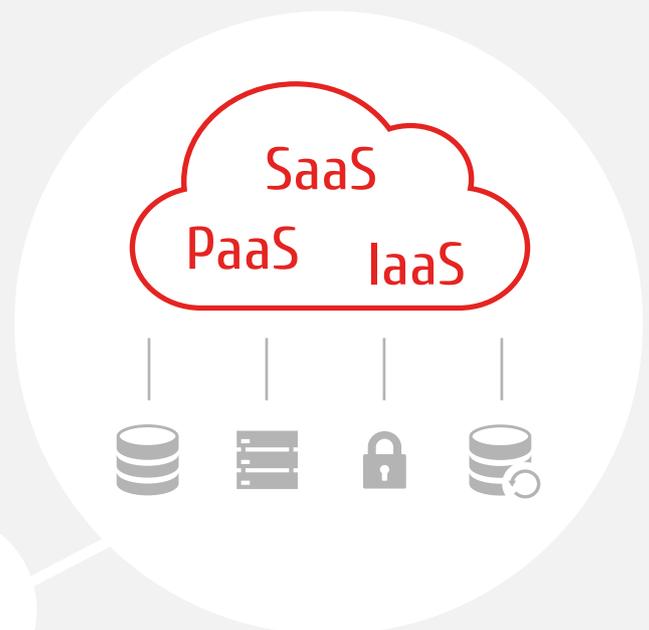
Ihr Partner für Engineered Systems & Cloud-Lösungen

Oracle Engineered Systems



- ▶ voll integrierte **Betriebsplattformen** für Datenbanken und Applikationen
- ▶ höchste **Verfügbarkeit, Performance und Skalierbarkeit** mit ODA, Exadata, PCA, ZFSSA und weiteren Systemen
- ▶ **professionelle und praxisorientierte Beratung** durch unsere Experten

Oracle Cloud



- ▶ PoC, Benchmarks und weitere **Evaluierungen** in der Private Cloud, Public Cloud und Cloud@Customer
- ▶ **Workshops, Assessments und TCO-Betrachtungen** für Ihre Anwendungen
- ▶ **Beratung, Unterstützung und Durchführung** von Migrationsprojekten
- ▶ Betriebsunterstützung und **Managed Services**



Wir leben Cloud.



- Bei den mittlerweile unüberschaubaren Cloud-Angeboten sind wir Ihr verlässlicher und vorausschauender Navigator für Ihren Weg in die Cloud. Für Sie haben wir sämtliche Aspekte im Blick und entscheiden gemeinsam mit Ihnen die richtige Strategie. Wir sind an Ihrer Seite: von der Beratung über die Planung und Umsetzung bis hin zu Training und Betrieb. Sprechen Sie mit uns.

www.trivadis.com/cloud-solutions | info@trivadis.com



BASEL ■ BERN ■ BRUGG ■ DÜSSELDORF ■ FRANKFURT A.M. ■ FREIBURG I.BR. ■ GENÈVE
HAMBURG ■ KOPENHAGEN ■ LAUSANNE ■ MÜNCHEN ■ STUTTGART ■ WIEN ■ ZÜRICH

trivadis
makes IT easier. ■ ■ ■