

Red Stack

Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

Die Cloud ist da



Aus der Praxis

Oracle Datenbanken
in Docker Containern

Im Interview

Dr. Jana Jentzsch, Jentzsch IT
Rechtsanwalts-gesellschaft



Terraform

Infrastructure as
Code in der OCI

ORACLE CLOUD **KOSTENLOS** TESTEN

BIS ZU 3.500
GRATIS STUNDEN

cloud.oracle.com/de_DE/tryit

Erstellen Sie einsatzbereite Workloads mit einer Vielzahl von Cloud-Services im Wert von 300\$!
Zum Beispiel für Datenbanken, Compute, Container, IoT, Big Data, API-Management, Integration, Chatbots und vieles mehr

ORACLE®



Robert Marz
Themenverantwortlicher
Cloud, DOAG DB Community

Liebe Mitglieder, liebe Leserinnen und Leser,

der Gang in die Cloud ist keine „Entweder-oder-Option“, sondern eine „Sowohl-als-auch-Entscheidung“. Immer mehr Firmen nutzen die Ressourcen aus der Wolke mittlerweile als eine flexible Erweiterung des eigenen Rechenzentrums.

Die DSGVO hat ihren Teil dazu beigetragen, dass die vorhandenen Sicherheitsbedenken durch pragmatische Einstellungen ersetzt wurden. Schließlich unterliegen auch die Cloud-Anbieter diesen Regeln. Die Cloud ist mehr als nur „die Rechner von jemand anderem“. Dieses „mehr“ zu erreichen, erfordert allerdings Umdenken und bringt Veränderung für die Aufgabengebiete vieler IT-Mitarbeiter: Bisherige Regelaufgaben entfallen, weil sie automatisiert vom Cloud-Anbieter erledigt werden. Dafür kommen neue, andere Aufgaben hinzu. Bislang klar abgegrenzte Aufgabengebiete vermischen sich und werden zusammengefasst. In jedem Fall wird neues Wissen gebraucht.

Bei der DOAG macht sich das wachsende Interesse an der Cloud auf verschiedene Arten bemerkbar: Die eingereichten Vorträge zu unseren Veranstaltungen behandeln immer öfter primäre Cloud-Themen. Die DOAG 2018 Konferenz und Ausstellung, die im November zum 31. Mal in Nürnberg stattfand, bot an allen Tagen zu jeder Uhrzeit gut besuchte Vorträge zum Thema Cloud.

Veranstaltungen mit Praxisbezug, die sich komplett um die Cloud drehen, wie zuletzt unser „Noon-2Noon - Die Datenbank in der Oracle Cloud“, sind in kurzer Zeit komplett ausgebucht. Das Finden von Autoren fiel uns selten so leicht wie für dieses Heft. Wir sind mit hochwertigen Inhalten geradezu überhäuft worden. Die Cloud ist da.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Lesen dieser Ausgabe.

MUNIQSOFT
TRAINING



Training

Training

Unter neuem Namen, aber in bewährter, hoher Schulungsqualität firmieren wir nun als

Munisoft Training GmbH

Wir haben unser Schulungsportfolio erweitert und würden uns freuen, Sie wieder bei uns zu begrüßen.

Telefon: 089 679090-40

Webseite: www.munisoft-training.de

Tipps: www.munisoft-training.de/tipps



Unser Kundenservice:

- Öffentliche Datenbankschulungen:

z. B. Neuerungen 12c	20.-24.05.2019	1.990,-€ netto
• SQL Grundlagen	11.-13.06.2019	990,-€ netto
• PL/SQL I	24.-28.06.2019	1.990,-€ netto
• DBA Sommerspecial	22.-26.07.2019	1.490,-€ netto
- Inhouse Schulungen individuell nach Ihren Wünschen, viele weitere Termine auf unserer Webseite.



Die Datenbankadministration ist deutlich einfacher als bei einer herkömmlichen Oracle-Datenbank



Wie sieht das „neue“ Cloud-Lizenzierungsmodell von Oracle aus?



Wo liegen die Vorteile der Cloud, welche Dienste und Abrechnungsmodelle von Oracle gibt es?

Einleitung

- 3 Editorial
- 5 Timeline
- 8 Nach Inkrafttreten der DSGVO „haben Anwender jetzt sowohl mehr Rechte als auch Pflichten“
Interview mit Dr. Jana Jentzsch
- 60 Spurensuche in Microservices
Matthias Fuchs und
Andreas Chatziantoniou
- 64 DOAG noon2noon 2019 –
Mit Begeisterung der Oracle Cloud
auf den Zahn gefühlt
Martin Klier und Robert Marz

Cloud

- 11 Wie autonom ist das Oracle Autonomous Data Warehouse?
Dani Schnider und Christian Antognini
- 17 Terraform: Infrastructure as Code in der Oracle Cloud Infrastructure
Marcus Schröder
- 26 Universal Credits: Einfacher Kauf und Verbrauch von Oracle IaaS und PaaS Cloud Services
Gertrud Abram-Retschitzegger
- 30 Oracle-Datenbanken in Docker-Containern
Stefan Oehrli
- 36 Security im Oracle-Cloud-Rechenzentrum der Generation 2 in Frankfurt
Michael Fischer
- 42 Datenbank in der Wolke – Teil 1: Abrechnungsmodelle in der Oracle Cloud
Borys Neselovskyi
- 48 Cloud Connectivity – Herausforderungen und Lösungen
Daniel Steiger
- 54 Autonomous Data Warehouse im Vergleich mit Konkurrenten
Dr.-Ing. Holger Friedrich

Entwicklung

- 59 DOAG Forms Day 2019 feiert 40 Jahre Oracle Forms
Dr. Jürgen Menge

Intern

- 65 Termine
- 65 Neue Mitglieder
- 66 Impressum
- 66 Inserenten

✦ Timeline

9. Januar 2019

Anlässlich der kommenden Data Analytics 2019 am 26. und 27. März stimmt sich die Leitung des Konferenzteams mit dem Programmkomitee von Oracle und DOAG in einer Telefonkonferenz über die Programmgestaltung ab. Alle Beteiligten sind zuversichtlich, wieder ein Top-Programm zusammenzustellen zu können.



15./16. Januar

In Frankfurt am Main nahmen 35 hoch motivierte Techniker und Entscheider an einem Event der besonderen Art teil: Im Rahmen des DOAG noon2noon 2019 konnten sie – meist zum ersten Mal – eigenhändig der Oracle Cloud auf den Zahn fühlen. Von zwölf Uhr mittags bis zum nächsten Tag um „High Noon“ war „Ärmel hochkrempeln und selbst machen“ angesagt. Fachlicher Schwerpunkt war, die Struktur der Oracle Cloud zu verstehen, mit der Oracle Cloud Infrastructure (OCI) auf grafischem Wege oder per Skript umzugehen, die Servicetypen kennenzulernen, zu nutzen und eine eigene kleine Lösung für sich zu finden und zu bauen. Das engagierte Team um die Organisatoren Robert Marz und Martin Klier bestand aus Katharina Büchse, Axel vom Stein und Sabine Heimsath. Hervorragende technische Unterstützung leistete die Oracle Business Unit Technology + Cloud, insbesondere Tobias Deml, Marcus Schröder und Manuel Hossfeld.

22. Januar 2019

In München trifft sich Fried Saacke mit dem Geschäftsführer und Gesellschafter René Lehrieder der gleichnamigen Catering-Firma, die uns jährlich zur Konferenz + Ausstellung bewirtet. Im Fokus steht bei dem Treffen ein Ideenaustausch und die Frage, wie man die Konferenzteilnehmer zukünftig noch besser bewirten kann.

23. Januar 2019

Im Phantasialand in Brühl, dem Veranstaltungsort der JavaLand 2019, findet ein Abstimmungs-Treffen zur Vorbereitung der JavaLand 2019 statt. Die große Herausforderung ist es, in den Gegebenheiten des Freizeitparks Phantasialand, bei einer auf über 2000 Teilnehmer angewachsenen Konferenz hinsichtlich der Raumkapazitäten Warteschlangen und überfüllte Konferenzräume zu ver-

meiden. Die Nutzung von mehr Restaurants zur Optimierung der Logistik und die Ausnutzung des Bus-Shuttles sind Maßnahmen, um die Konferenz attraktiv zu halten.

31. Januar 2019

Die DOAG-Geschäftsstelle trifft sich zum jährlichen Kick-off-Meeting im Schloss Britz. Fried Saacke stimmt die Mitarbeiter auf die umfangreichen Herausforderungen ein. Großen Wert legt er darauf, dass diese gemeinsam bewältigt werden.

19. Februar 2019

Praxisnahe Vorträge und Erfahrungsberichte über den Stand der Technik sowie die Herausforderungen und Chancen, die sich aus einer Migration ergeben, stehen im Mittelpunkt des DOAG Day "Datenbank Migration" am 19. Februar in Mannheim. Die Oracle Datenbank Architektur steht vor einem grundlegenden Umbruch. Vor einigen Jahren wurde „Multitenant“ als Architektur der Zukunft eingeführt. Nun ist diese gereift, erste Kinderkrankheiten sind beseitigt und viele Funktionen und Möglichkeiten ergänzt worden. Anwender beschäftigen sich in zahllosen Projekten mit dem Umstieg auf diese neue Architektur.

21. Februar 2019

In München findet in der Oracle-Niederlassung München zum dritten Mal mit 54 Teilnehmern der Forms Day der DOAG statt. Diesmal galt es, ein besonderes Jubiläum zu feiern. Die Ursprünge der Technologie Oracle Forms reichen bis in das Jahr 1979 zurück. Doch Oracle Forms behauptet sich weiterhin als leistungsfähige Technologie für datenbank-getriebene Anwendungen in zahlreichen Unternehmen. Da jedoch viele Forms-Anwendungen in die Jahre gekommen sind, besteht ein dringender Bedarf an Modernisierung, sowohl was das Look & Feel als auch die Funktionalität betrifft. Der Höhepunkt der Veranstaltung ist zweifellos die Videokonferenz mit Bill Friend und Michael Ferrante in den USA.



22. Februar 2019

Der DOAG-Vorstand trifft sich in Koblenz zu seiner ersten Sitzung im neuen Jahr, die im Zeichen von Führungskräften und Delegiertenversammlung steht.

27./28. Februar 2019

Die Primavera Community trifft sich im Hotel Pullmann Stuttgart Fontana in Stuttgart zum 12. Primavera Community Day. Am Vorabend steht als besondere Veranstaltung eine Talquerung durch das Bauprojekt "Stuttgart 21" auf der Agenda. Ein Erfahrungsaustausch zum Thema Oracle-Lizenzen und Support unter Betrachtung ihrer Rechtmäßigkeit unter der Leitung von Dr. Dietmar Neugebauer und Michael Paege vom DOAG Legal Council ist auch ein Bestandteil des Programms. Thomas Koytek von der Visbo GmbH stellt zudem das Konzept der „smart Powerpoints“ vor, die die Kommunikation von Projekt- und Multiprojekt-Informationen auch in weltweit verteilten Projekten stark vereinfachen können. Zoran Trajkovski & Ana Atanasova von der Seavus Group präsentieren den ScheduleReader zum sicheren Teilen von vertraulichen Daten. Sascha Umbach von Oracle Deutschland B.V. Co. berichtet über die Anwendung von Aconex. Jürgen Schachner von Oracle Deutschland stellt die Oracle Roadmap vor.

19. bis 21. März 2019

Die JavaLand 2019 öffnet ihre Pforten im Phantasialand in Brühl. Das große Wiedersehen der Community mit Freizeitpark-Flair bietet auch in diesem Jahr wieder die ideale Umgebung für Wissensaustausch und Networking. Die über 2000 Teilnehmer erwartet ein erstklassiges Konferenzprogramm mit zahlreichen Vorträgen von Top-Referenten, interaktiven Community-Aktivitäten und jede Menge Spaß! Ed Burns berichtet in seiner Eröffnungsk

note von seinen weitreichenden IT-Erfahrungen der letzten 25 Jahre. Die beliebte Abendveranstaltung "OpenPark" am Ende des ersten Konferenztages lockt mit verschiedenen Themenbuffets zum Abendessen und ermöglicht die Nutzung einer ganzen Reihe von Fahrgeschäften im Phantasialand.



26. und 27. März 2019

Ebenfalls im Phantasialand in Brühl findet mit der Data Analytics 2019 die gemeinsame Konferenz von DOAG und Oracle unter dem Motto "die Daten-Explosion meistern" zu den Themen Advanced Analytics, Big Data, Cloud und Data Warehousing statt. Mit über 30 attraktiven, praxisrelevanten Vorträgen erhalten die Besucher wertvollen Input für Ihre tägliche Unternehmenspraxis und Projektarbeit sowie neue Impulse für die Umsetzung moderner Digitalisierungs-Strategien. Wie im vergangenen Jahr bietet Oracle als Auftakt der Veranstaltung drei exklusive, kostenfreie Knowledge Transfer Sessions zu ausgesuchten Themen im Analytics-Umfeld in Workshops an.



Dr. Dietmar Neugebauer
Ehemaliger DOAG-Vorstands-
vorsitzender

Aus der Ferne betrachtet: Auch zu viel Schnee wird mal Schnee von gestern ...

Noch bringen die Wolken den Schnee meterhoch hier in den Alpenraum. Ich höre gerade, dass die Jachenau und unser DOAG Botschafter Günter Unbescheid schon über mehrere Tage komplett eingeschneit sind. Wen stört es? Mit unserer Infrastruktur ist unsere Versorgung über mehrere Tage gesichert. Wir können sogar von zu Hause aus ohne Problem weiterarbeiten. Unsere Daten sind weltweit verfügbar. Die Cloud hat Fahrt aufgenommen. Und gerade bezüglich der Verfügbarkeit der Daten ist Oracle hier führend mit dabei. Die technischen Entwicklungen

der letzten Jahre in Richtung der autonomen Datenbank von Oracle sind wichtige Meilensteine im Datenbankbetrieb eines Cloud Data Centers gewesen.

Doch der Markt der führenden Anbieter von Cloud Services ist umkämpft. Technologievorsprung ist nicht immer das entscheidende Merkmal. Security und Einhaltung der DSGVO spielen eine entscheidende Rolle. Da vor allem auch hybride Lösungen nachgefragt werden, erwartet der Kunde eine hohe Flexibilität, um von einem Anbieter zum anderen zu wechseln, und möchte auch das Vertrauen haben, dass der Cloud-Betreiber ihm das gewährleisten kann.

Ob die Oracle-Strategie richtig ist, mit dem Fokus, die technisch weit entwickelten, hochverfügbaren Oracle-Produkte nur in der eigenen Cloud laufen zu lassen, wage ich zu bezweifeln. Mit dem Wechsel von Thomas Kurian, bisher zuständig für die Produktentwicklung bei Oracle, zu Google als CEO von Google Cloud werden auch Differenzen in der Oracle-Führung zu dieser Ausrichtung in Verbindung gebracht. Da Cloud für die Kunden eine neue Strategie ist, werden die Kunden hier vor einer Migration die Angebote genau überprüfen. Wichtig für den Kunden ist dabei besonders die Garantie, nicht von einem Cloud-Anbieter abhängig zu werden.

Abschottung ist da – sage ich mal – Schnee von gestern!



DOAG 2019 Datenbank

3. und 4. Juni 2019 in Düsseldorf

DOAG

datenbank.doag.org





Dr. Jana Jentzsch

Nach Inkrafttreten der DSGVO „haben Anwender jetzt sowohl mehr Rechte als auch Pflichten“

Das Thema „Cloud“ und „DSGVO“ hat bei vielen Unternehmen zu Unsicherheit und Fragen geführt. Martin Meyer, Redaktionsleiter des Red Stack Magazin, sprach darüber mit Dr. Jana Jentzsch, JENTZSCH IT Rechtsanwaltsgesellschaft

Woran erkennt man 6 Monate nach Inkrafttreten die Auswirkungen der Europäischen Datenschutzgrundverordnung?

Dr. Jana Jentzsch: Man bemerkt es vor allem an den vielen deutlichen Hinweisen auf Datenschutzbestimmungen, die man inzwischen deutlich vermehrt antrifft. Das bezieht sich sowohl auf das tägliche Leben, wie zum Beispiel bei den Hinweisen auf Videoüberwachung in Ladengeschäften, als auch auf Informationen zur Datenverarbeitung bei einer Registrierung auf Webseiten oder in Emails.

Bringt die neue DSGVO bedeutende rechtliche Verbesserungen für den Anwender von Oracle-Cloud-Produkten und -Dienstleistungen? Welche praxisrelevanten Neuerungen beurteilen Sie als besonders wichtig?

Dr. Jana Jentzsch: Grundsätzlich haben die Anwender jetzt sowohl mehr Rechte als auch Pflichten in bestimmten Konstellationen erhalten. Hierbei darf aber nicht übersehen werden, dass die Anwender für die Verarbeitung von personenbezogenen Daten verantwortlich sind und bleiben und Oracle der Auftragsverarbeiter ist.

Der Auftragsverarbeiter Oracle als Anbieter von Cloud Services ist jetzt durch die DSGVO zu mehr Transparenz, insbesondere hinsichtlich der Verarbeitung personenbezogener Daten, gezwungen. Ein Auftragsverarbeiter muss jetzt viel genauer angeben, welche Kategorien von Daten und zu welchen Zwecken verarbeitet werden. Der Einsatz von Subunternehmen muss offengelegt werden und der Anwender hat nun auch ein Recht zu widersprechen, wenn Subunternehmer einfach ausgetauscht werden.

Diese Tatsache ist insbesondere deshalb von Bedeutung, weil große Cloud-Anbieter oft mit zahlreichen Subunternehmen weltweit zusammenarbeiten. Für den Anwender ist die Transparenz bei der Datenverarbeitung in der Cloud gegeben, da er genau den Datenfluss nachvollziehen und auch kontrollieren kann.

Ist die DSGVO Ihrer Meinung nach gelungen bzw. schon ausreichend, um bei Einhaltung die Interessen der Oracle-Cloud-Nutzer wirkungsvoll zu schützen? Wo sehen Sie gegebenenfalls Verbesserungsbedarf?

Dr. Jana Jentzsch: Hier möchte ich zunächst nochmal klarstellen, dass die DSGVO personenbezogene Daten schützt und nicht generell alle Daten oder auch nicht alle Daten der Oracle-Anwender. Hier muss man schon trennen zwischen der DSGVO – also dem Schutz von personenbezogenen Daten – und dem Schutz von beispielsweise Geschäfts- und Betriebsgeheimnissen. Diese Tatsache wird manchmal vermengt oder nicht richtig auseinandergehalten. Die Oracle-Anwender sind nicht grundsätzlich Begünstigte der DSGVO.

Geschützt nach der DSGVO sind Personen, also Menschen, deren personenbezogene Daten verarbeitet werden. Der Oracle-Anwender ist Verantwortlicher der Datenverarbeitung, da in der Regel seine Mitarbeiter die Cloud-Produkte bedienen oder gegebenenfalls Dritte und diesbezüglich legt die DSGVO den Anwendern auch zusätzliche Pflichten auf. Die Schutzrichtung geht immer dahin, den Menschen zu schützen und nicht den Oracle-Anwender als Unternehmen. Die DSGVO schießt vermutlich sogar – Stichwort Verbesserungsbedarf – grundsätzlich über das Ziel hinaus, weil in der Praxis schon Situationen entstehen, die kaum noch handhabbar sind. Das wird beispielsweise deutlich an den extensiven Informationspflichten. Dies sind Pflichten für die Anwender, die Daten von Personen verarbeiten. Für diese bestehen grundsätzlich sehr weitgehende Pflichten, die Betroffenen über den Umfang und den Zweck der Datenverarbeitung zu informieren.

In der unternehmerischen Praxis ist es schon ein Thema, wie dies im Einzelfall interessengerecht umgesetzt werden kann. Ich bin aber nicht der Ansicht, dass hier Interessen der Oracle-Cloud-Nutzer durch die DSGVO nicht genügend berücksichtigt werden oder dass die DSGVO noch schärfer werden müsste. Die sich stellende Frage ist eher: Wie bekommt man die Rechte und Pflichten so sinnvoll in Einklang, dass alles praktisch und handhabbar bleibt und trotzdem ein wirkungsvoller Schutz gewährleistet ist sowie eine wirkungsvolle Kontrolle ermöglicht wird.

Wie gut hat Oracle die neue Rechtslage in seine Lizenzverträge integriert? Welche neuen Oracle-Angebote zur Datenbank-Sicherheit nach DSGVO (DSGVO Compliance) gibt es?

Dr. Jana Jentzsch: Oracle hat auf die DSGVO reagiert und das Data Processing Agreement for Oracle Cloud Services angepasst. Oracle hat zudem zahlreiche Sicherheitsprodukte für Cloud Services im Angebot, die die Anwender dabei unterstützen sollen, die Compliance-Anforderungen der DSGVO einzuhalten. Das sind z.B. Sicherheitsfunktionen für Datenbanken und Sicherheitslösungen für Anwendungen und Personen.

Das Legal Council

Das DOAG Legal Council ist eine im April 2018 gegründete Runde mit spezialisierten IT-Rechtsanwältinnen und Rechtsanwälten, die die fachliche Kompetenz der DOAG ergänzt und somit ihren Mitgliedern einen zusätzlichen Mehrwert bietet.

Juristische Aspekte und vielfältige rechtliche Herausforderungen spielen auch in der Arbeit der DOAG und der Vertretung der Mitgliederinteressen eine immer größere Rolle.

Das DOAG Legal Council steht den DOAG-Gremien und den DOAG Competence Centern beratend zur Seite, berät sich gebündelt über Anliegen der Mitglieder und tauscht mit dem gesamten DOAG-Netzwerk fundiertes Fachwissen aus.

Ob Vertragsgestaltung, Lizenzierung oder Datenschutz – das DOAG Legal Council widmet sich der Aufgabe, rechtliche Themen für IT-Spezialisten zugänglicher zu machen und der DOAG bei der Vertretung der Interessen der Mitglieder zur Seite zu stehen.

Fundiertes Wissen wird zum Beispiel in Form von Fachartikeln oder Fachvorträgen in den eigenen Medien bzw. Veranstaltungen im DOAG-Netzwerk zur Verfügung gestellt.

Mit den zahlreichen Oracle-Database-Security-Produkten gibt es ziemlich umfangreiche Lösungen, wie zum Beispiel Oracle Database Key Vault oder Oracle Advanced Security. Diese Produkte zielen darauf ab, die technisch-organisatorischen Maßnahmen, zu denen die Kunden und Anwender bei einem bestehenden Verhältnis der Auftragsverarbeitung verpflichtet sind, angemessen umzusetzen. Insbesondere dem Anwender werden Werkzeuge an die Hand gegeben, um sich sicherheitstechnisch insbesondere bei Verschlüsselungen auf ein hohes Niveau zu bringen.

Wird das von Oracle vorgelegte neue „Data Processing Agreement for Oracle Cloud Services“ (DPA) den Datenschutzanforderungen in Deutschland und der Europäischen Union schon gerecht?

Dr. Jana Jentzsch: Oracle hat sich bemüht, die neuen rechtlichen Rahmenbedingungen umzusetzen. Im Vergleich zur Vorversion gab es viele Anpassungen im Data Processing Agreement (DPA). Natürlich gibt es für viele Unternehmen bei einigen Vorschriften immer noch Unklarheiten, weil die DSGVO auch nicht für jeden auf den ersten Blick leicht verständlich ist. Bei einigen Punkten, wie zum Beispiel der Höhe des Detaillierungsgrades, dem Zweck der Verarbeitung personenbezogener Daten oder den Datenkategorien, ist es schwer, ein abschließendes Urteil zu treffen. Es gibt noch keine Rechtsprechung zu solchen Einzelfragen. Ich würde jedoch insgesamt sagen, dass das Oracle Data Processing Agreement den gesetzlichen Vorgaben schon sehr nahekommt. Es gibt aber durchaus noch ein paar Einzelpunkte, über die die Anwender mit Oracle nachverhandeln sollten, auch um ihre eigenen Pflichten nach der DSGVO zu erfüllen.

Welche Kontrollrechte räumt Oracle seinen Cloud-Nutzern aktuell ein?

Dr. Jana Jentzsch: Grundsätzlich möchte ich nochmal darauf hinweisen: Der Oracle-Anwender ist nach der DSGVO für die Datenverarbeitung verantwortlich. Oracle ist der beauftragte Dienstleister. Der Cloud-Anwender als Verantwortlicher ist grundsätzlich weisungsberechtigt und bestimmt deshalb auch Inhalt und Zweck der Datenverarbeitung – nicht Oracle! Der Anwender darf daher kontrollieren, ob diese Weisungen durch Oracle ordentlich umgesetzt werden. Der Anwender darf auch einmal pro Jahr Oracle auditieren und überprüfen, ob Oracle seine Pflichten aus dem Data Processing Agreement einhält. Dies ist recht extensiv in Ziffer 10 des DPA geregelt und entspricht auch grundsätzlich den gesetzlichen Vorgaben. Dies ist für die Cloud-Nutzer eine recht kuriose Situation, da sie ja aus dem Oracle-Lizenzbereich gewohnt sind, selbst auditiert zu werden. Hier ist es jetzt umgekehrt. Der Anwender hat ein Auditrecht gegenüber Oracle.

Welche zusätzlichen Pflichten kommen auf die Anwender auf Basis der neuen Verträge zu?

Dr. Jana Jentzsch: Von großer Wichtigkeit ist die Notwendigkeit, dass die Anwender die Ausübung ihrer Rechte nicht vernachlässigen dürfen!

Häufig wird es beispielsweise nur als ein Recht interpretiert, neuerdings auch kontrollieren zu dürfen, es aber nicht zwingend zu müssen. Dies ist aber falsch! Es sind tatsächlich Pflichten, die hier dem Anwender neu auferlegt werden. Die Anwender sind verantwortlich für die Datenverarbeitung und sollten diese Kontrollrechte auch wirklich in der Praxis wahrnehmen.

Darüber hinaus gibt es noch weitere Pflichten, die durch die DSGVO etabliert werden. Anfragen von Betroffenen (z.B. Mitarbeitern, Privatkunden, Nutzern) zur Datenverarbeitung müssen beantwortet werden. Grundsätzlich ist es nicht die Pflicht von Oracle, sondern die Pflicht des Anwenders, solche Anfragen zu beantworten.

Von großer Bedeutung ist auch die neue Pflicht, dass unter bestimmten, oft schnell erreichten Voraussetzungen Datenvorfälle an die zuständige Datenschutz-Aufsichtsbehörde gemeldet werden müssen und gegebenenfalls auch an die Betroffenen, je nachdem wie hoch die Risikostufe ist, die diesem Datenvorfall entspricht. Dies kann beispielsweise auch bei einem reinen Datenverlust der Fall sein. Im Vergleich zu früher wurden die Meldepflichten stark verschärft. In der Regel müssen die zuständigen Behörden mit einbezogen werden, was sicherlich aus Sicht vieler Unternehmen keine besonders beliebte Vorschrift ist.

Wie sehen Sie die Datensicherheit berücksichtigt?

Dr. Jana Jentzsch: Die Datensicherheit wird in Zusatzdokumenten zum Data Processing Agreement (DPA) geregelt. Grundsätzlich ist davon auszugehen, dass Oracle angemessene Datensicherheitsvorschriften hat. Anwender sollten sich aber in Abhängigkeit der von ihnen genutzten Produkte auch individuell mit ihrer eigenen Situation beschäftigen. Bezüglich der Datensicherheit lässt sich nicht pauschal sagen, welche Sicherheitsmaßnahmen genau erforderlich sind. Dies orientiert sich immer an der konkreten Risikostufe.

Bei sensiblen Bewerberdaten, Arbeitnehmerdaten oder Gesundheitsdaten von Patienten, die in der Cloud verwaltet wer-

den, wären viel schärfere und höhere Maßnahmen zur Datensicherheit erforderlich, als wenn man nur eine Urlaubsliste oder eher unkritische Daten verarbeitet.

Deshalb sollte man diese Frage nicht pauschal beantworten, sondern kann feststellen, dass Oracle die Möglichkeiten und technischen Funktionen hat, Datensicherheit auf der höchsten Stufe zu berücksichtigen. Der Anwender muss aber schon im Einzelfall genau prüfen, ob das in seinem Vertrag für die von ihm in der Cloud verarbeiteten Daten auch konkret angemessen ist.

Haben Sie schon Erfahrungen aus ersten Fällen aus der Rechtspraxis?

Dr. Jana Jentzsch: In den Fällen, die ich betreue, geht es um konkrete praktische Umsetzungsfragen und nicht um Streitfragen.

Die Anwender fragen beispielsweise, wie ein Verzeichnisverzeichnis angelegt werden soll, wie Oracle als Auftragsverarbeiter unterstützend einbezogen werden soll oder wie generell die Kommunikation mit Oracle gehandhabt werden sollte. Es gibt in unserer Kanzlei auch Anfragen von Anwendern, die Unterstützung von Oracle benötigen und Fragen zur Etablierung von Prozessen haben.

Weitere Informationen

Webinar-Aufzeichnung vom 15. Februar 2019 – „Was hat sich nach der DSGVO für Oracle-Anwender geändert?“
<https://www.doag.org/de/home/news/aufgezeichnet-was-hat-sich-nach-der-dsgvo-fuer-oracle-anwender-geändert/detail/>

Zur Person: Dr. Jana Jentzsch

Dr. Jentzsch studierte Rechtswissenschaften an den Universitäten Bonn, Mailand und Köln und arbeitete während des Studiums als freie journalistische Mitarbeiterin für einen Verlag im Bundespressehaus. Nach dem ersten Staatsexamen erwarb sie einen „Master of Laws“ (LL.M.) im internationalen Luft- und Weltraumrecht an der Universität Leiden (Niederlande). Im Rahmen des LL.M.-Studiums absolvierte sie Stationen beim Internationalen Schiedsgerichtshof in Den Haag und bei der Europäischen Weltraumagentur in Paris.

Nach dem zweiten juristischen Staatsexamen 2005 erfolgte die Zulassung zur Anwaltschaft. Dr. Jentzsch arbeitete zunächst als Rechtsanwältin in einer internationalen Wirtschaftssozietät in Frankfurt am Main in der Praxisgruppe „Technologie Medien Telekommunikation“.

Anschließend war sie für zweieinhalb Jahre in einer renommierten Hamburger Medienrechtskanzlei tätig, bevor sie im Jahr 2009 die eigene IT-Rechtsberatung gründete. Seit Dezember 2011 ist Dr. Jentzsch von der Rechtsanwaltskammer Hamburg als Fachanwältin für Informationstechnologierecht zugelassen. Dr. Jentzsch berät insbesondere im Bereich Softwarelizenzen, IT-Projekte und neue Geschäftsmodelle.



Wie autonom ist das Oracle Autonomous Data Warehouse?

Dani Schnider, Christian Antognini, Trivadis AG

Mit dem Autonomous Data Warehouse (ADW) verspricht Oracle eine selbstkonfigurierende, schnelle, sichere und skalierbare Cloud-Plattform für Data Warehouses. Die Installation erfolgt auf Knopfdruck, Entwicklung und Betrieb des Data Warehouse werden viel einfacher. Um Dinge wie Upgrades und Patches, Backups, Performanceoptimierung, Skalierung, Indexierung und Partitionierung, Materialized Views, Optimizer-Statistiken und andere „Nebensächlichkeiten“ braucht sich der Kunde nicht mehr zu kümmern.

Die Marketingaussagen von Oracle tön-
 en verlockend – oder beängstigend, je
 nach Ausgangslage. Doch was steckt wirk-
 lich dahinter? Was ändert sich, wenn wir
 unsere Data Warehouses mit ADW be-
 treiben? Welche Aufgaben werden ver-
 einfacht oder überflüssig? Welche Aus-
 wirkungen hat die selbstkonfigurierende
 Cloud-Lösung auf Datenbankdesign, La-
 deprozesse und analytische Abfragen?

Wir wollten genauer wissen, was sich
 hinter der neuen DWH-Plattform ver-
 birgt und wie die „selbstkonfigurierende“
 Lösung in der Praxis funktioniert. Inter-
 essiert haben uns vor allem Performan-
 ceaspekte sowie die Fragen, wo wir noch
 Einfluss nehmen müssen und können
 und welche Aufgaben beim Aufbau und
 Betrieb eines Data Warehouse in Zukunft
 wegfallen oder vereinfacht werden.

Einfaches Setup, Administration über Browser

Die gute Nachricht vorweg: Das ADW ist
 eine gut konfigurierte Datenbankumge-
 bung, die für effiziente Ladeprozesse und
 analytische Abfragen im Data Warehouse
 optimiert ist. Das Setup ist sehr einfach
 und kann über ein paar wenige Eingaben
 in einem Webbrowser durchgeführt wer-
 den. Nach wenigen Minuten steht eine
 funktionierende DWH-Plattform in der
 Oracle Cloud zur Verfügung.

Die Administration der Datenbank er-
 folgt ebenfalls über den Browser. Neben
 dem Starten und Stoppen der Datenbank
 gehört hierzu die Skalierung der Umge-
 bung. Die Anzahl der CPU Cores und der
 Diskplatz in Terabyte kann bei laufendem
 Betrieb angepasst werden. Backups wer-

den automatisch erstellt. Bei Bedarf kön-
 nen manuelle Backups und – wenn nötig –
 Restores auf einen bestimmten Zeitpunkt
 durchgeführt werden. Über die „Service
 Console“ (siehe Abbildung 1) können die
 Aktivitäten der Datenbank und die ausge-
 führten SQL-Befehle überwacht werden.

Auf den ersten Blick fällt die einfache
 Administration auf. Auf den zweiten Blick
 wird jedoch klar, dass diese auch eine
 Kehrseite hat. Der erfahrene Datenbank-
 administrator wird schnell feststellen,
 dass es im ADW zahlreiche Einschränkun-
 gen gibt. Die meisten Konfigurationspa-
 rameter können nicht verändert werden,
 einige SQL-Befehle sind nicht erlaubt, Zu-
 satzparameter werden ignoriert.

So wird zum Beispiel der Versuch, ei-
 nen eigenen Tablespace anzulegen, mit
 der Fehlermeldung „ORA-01031: insuf-
 ficient privileges“ bestraft. Eine TABLE-
 SPACE-Angabe beim Erstellen von Ta-
 bellen wird einfach ignoriert. Ebenfalls
 verboten ist das Erstellen von Database
 Links. Partitionierte Tabellen, Indizes und
 Materialized Views können zwar manu-
 ell erstellt werden. Oracle weist jedoch
 in der ADW-Dokumentation darauf hin,
 dass dies nur von „highly skilled Oracle
 Database tuning experts“ gemacht wer-
 den soll und dass danach die Auswirkun-
 gen gut getestet werden sollen. Optimizer-
 Hints können nur verwendet werden,
 wenn der Konfigurationsparameter OP-
 TIMIZER_IGNORE_HINTS zuvor auf FALSE
 gesetzt wird.

Dies alles klingt nach einer sehr limi-
 tierten Datenbankumgebung mit zahl-
 reichen Einschränkungen. Die Absicht
 dahinter ist klar: Je weniger an der Kon-
 figuration geändert werden kann, desto
 einfacher kann das System betrieben und
 bei Bedarf skaliert werden. In vielen Fäl-
 len ist dies auch sinnvoll. Dass zum Bei-
 spiel in den ersten Versionen des ADW
 keine Indizes (außer Primary Key und
 Unique Constraints) angelegt werden
 konnten, war sogar ein Vorteil. Viele Data
 Warehouses sind „überindexiert“, was
 nicht nur zu lang laufenden ETL-Prozes-
 sen führt, sondern oft auch zu ineffizien-
 ten Abfragen. Für analytische Abfragen
 sind in der Regel Full Table Scans viel ef-
 fizienter. Dieser Effekt wird hier noch ver-
 stärkt durch effiziente Smart Scans (das
 ADW wird auf Exadata betrieben) und
 spaltenweise HCC-Komprimierung (Hyb-
 rid Columnar Compression) aller Tabel-

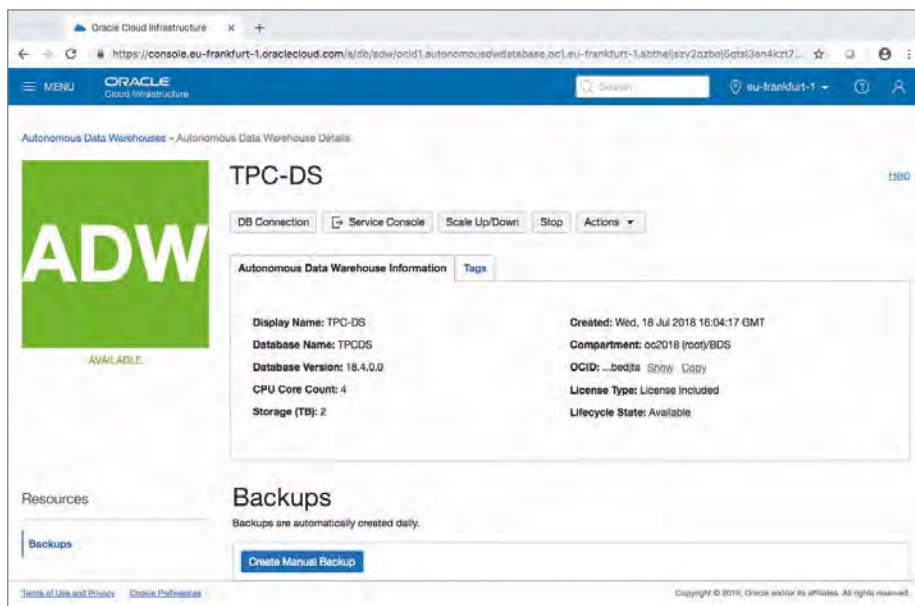


Abbildung 1: Administration des Autonomous Data Warehouse über Browser

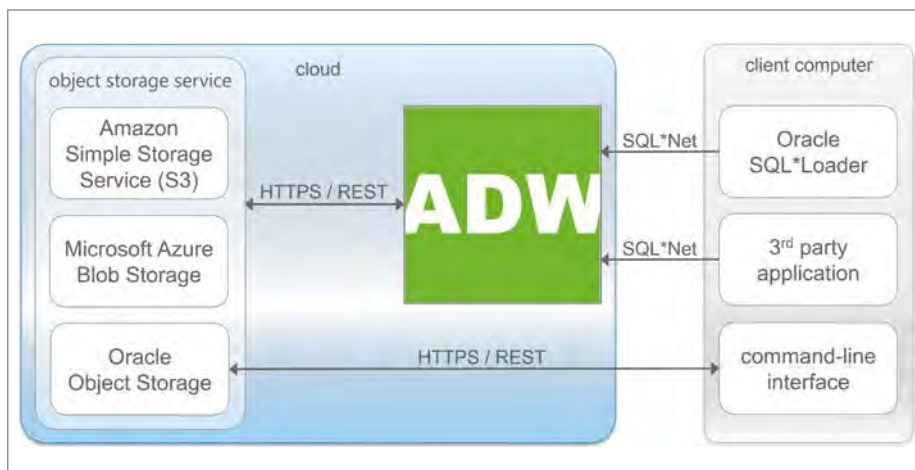


Abbildung 2: Daten können über unterschiedliche Services in ADW geladen werden

Id	Operation	Name	TQ	IN-OUT	PQ Distrib
0	INSERT STATEMENT				
1	LOAD AS SELECT	TARGET			
2	PX COORDINATOR				
3	PX SEND QC (RANDOM)	:TQ10000	Q1,00	P->S	QC (RAND)
4	OPTIMIZER STATISTICS GATHERING		Q1,00	PCWC	
5	PX BLOCK ITERATOR		Q1,00	PCWC	
6	TABLE ACCESS STORAGE FULL	SOURCE	Q1,00	PCWP	

- automatic DOP: Computed Degree of Parallelism is 8

Listing 1: Execution Plan eines Ladeprozesses mit Parallel DML und Direct Path Load

len. Die Komprimierung kann bei Bedarf ausgeschaltet oder geändert werden, beispielsweise für Tabellen mit sehr vielen UPDATE-Operationen.

Wie kommen die Daten in die Cloud?

Ein Data Warehouse wird mit Daten aus unterschiedlichen Quellsystemen geladen. Wie kann dies nun im ADW implementiert werden, wenn keine Database Links zur Verfügung stehen? Die effizienteste und von Oracle empfohlene Möglichkeit besteht darin, Datenfiles in einen speziellen Object Storage in der Cloud zu stellen und von dort mittels External Tables ins ADW zu laden. Der Object Storage liegt typischerweise auch in der Oracle Cloud, aber auch andere Cloud-Anbieter wie Amazon oder Microsoft werden unterstützt. Das Bereitstellen und Hochladen der Files kann ebenfalls über den Browser erfolgen. Sinnvoller für einen regelmäßigen Betrieb ist jedoch die Verwendung des Command Line Interface

der Oracle Cloud Infrastructure (OCI).

Stehen die Daten in einem Object Storage bereit, kann mit dem im ADW vorhandenen PL/SQL-Package DBMS_CLOUD eine External Table auf jedes File erstellt werden, über welches die Daten in die Datenbank geladen werden, üblicherweise in eine Staging Area.

Auch das Importieren von Daten aus einem Object Store mit Data Pump wird unterstützt. Dazu müssen zusätzlich zu den Verbindungsinformationen zum ADW die Credentials und die URL des Object Store als Parameter von IMPDB angegeben werden.

Ebenfalls möglich ist die Verwendung von SQL*Loader, falls Files von einem lokalen Server direkt ins ADW geladen werden sollen. In diesem Fall werden die Daten über eine SQL*Net-Verbindung direkt in die Zieltabelle der Staging Area geschrieben.

Selbstverständlich ist es auch möglich, Daten aus Files, Datenbanken oder anderen Datenquellen mithilfe von ETL-Tools (z.B. Oracle Data Integrator oder ETL-Tools von Fremdherstellern) ins ADW zu laden.

Der Verbindungsaufbau kann dabei via Oracle Call Interface (OCI), JDBC OCI, JDBC Thin Client oder ODBC erfolgen.

Performance von ETL-Prozessen

Das Laden von Daten aus unterschiedlichen Quellsystemen in ein Data Warehouse ist oftmals nur der erste Schritt. Typischerweise werden danach die Daten in verschiedenen Schritten bereinigt, integriert und transformiert und dabei in die nächste DWH-Schicht geladen. Dies entspricht dem üblichen Vorgehen in einem mehrschichtigen Data Warehouse. Für die Transformation von einer Schicht zur nächsten sollten aus Performancegründen mengenbasierte Operationen durchgeführt werden. Dies erfolgt entweder direkt mit SQL-Statements in der Datenbank oder über ein ETL-Tool, das die Möglichkeit bietet, mengenbasierte SQL-Befehle auszuführen.

Das ADW ist auf mengenbasierte Ladeprozesse optimiert. Die Datenbank ist so

Id	Operation	Name
0	INSERT STATEMENT	
1	LOAD TABLE CONVENTIONAL	TARGET
2	TABLE ACCESS STORAGE FULL	SOURCE

- automatic DOP: Computed Degree of Parallelism is 1 because of no expensive parallel operation
 - PDML disabled because object is not decorated with parallel clause
 - Direct Load disabled because no append hint given and not executing in parallel

Listing 2: Execution Plan eines Ladeprozesses unter dem Service LOW

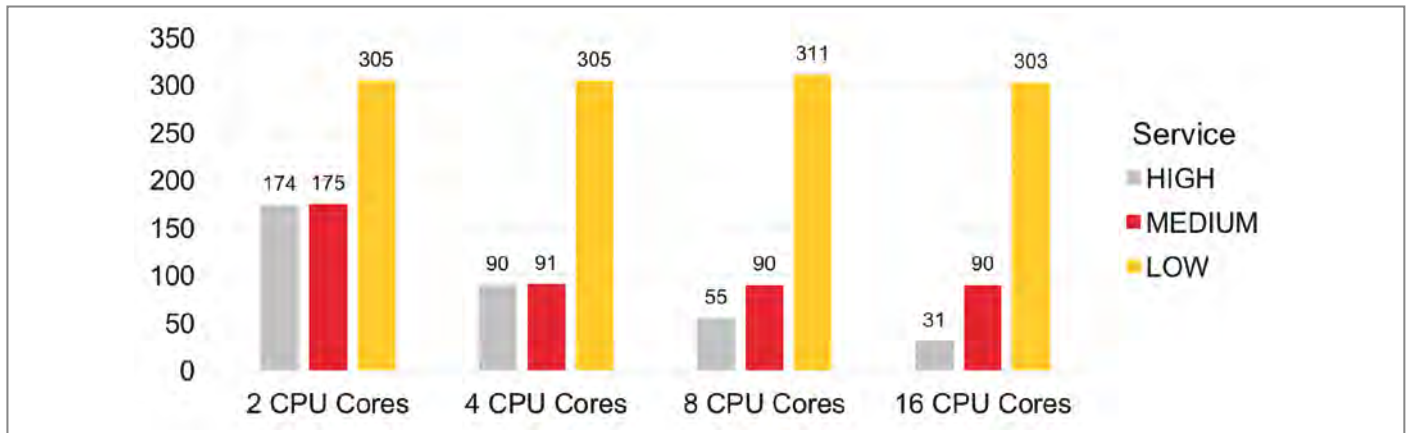


Abbildung 3: Skalierbarkeit von Abfragen (Antwortzeit in Sekunden)

konfiguriert, dass die Ladeprozesse mittels Parallel DML und Direct Path Loads durchgeführt werden (siehe Listing 1). Die Anzahl paralleler Prozesse (Degree of Parallelism, DOP) ist abhängig von der konfigurierten Anzahl der CPU Cores. Das mit Oracle Database 12c eingeführte Feature „Online Statistics Gathering“ wurde für ADW so erweitert, dass die Optimizer-Statistiken nicht nur beim Laden in eine leere Tabelle berechnet werden, sondern auch bei inkrementellen Ladeprozessen. Zusätzlich werden auch automatisch Histogramme berechnet. Das explizite Berechnen von Statistiken mit DBMS_STATS ist somit nicht mehr notwendig – zumindest nicht, wenn alle Voraussetzungen erfüllt sind. Denn es sind einige wichtige Punkte zu beachten, damit dies alles einwandfrei funktioniert.

Zwar ist Parallel DML für alle Sessions standardmäßig eingeschaltet, es wird aber nur verwendet, wenn mindestens 2 CPU Cores konfiguriert sind und wenn die Session unter der Consumer Group HIGH oder MEDIUM ausgeführt wird. Für jede erstellte Datenbank in ADW werden drei Services (HIGH, MEDIUM und LOW) zur Verfügung gestellt, die vorkonfigurierte (und nicht veränderbare) Consumer Groups vom Oracle Resource Manager verwenden. Nur mit dem Service HIGH kann die maximale Anzahl von CPUs verwendet werden. Mit MEDIUM wird immer DOP 4 verwendet, mit LOW werden alle Statements seriell ausgeführt (siehe Listing 2). Damit wird auch nicht mehr Direct Path Load verwendet, und ohne Direct Path Load funktioniert auch das „Online Statistics Gathering“ nicht. ETL-Jobs sollten somit immer unter dem Service HIGH oder MEDIUM ausgeführt werden.

Ein ähnlicher Effekt tritt auf, wenn referenzielle Integrität verwendet wird. Sind auf der Zieltabelle Foreign Key Constraints definiert und eingeschaltet, kann zwar Parallel DML verwendet werden, jedoch nicht Direct Path Load. In diesem Fall müssen die Optimizer-Statistiken unbedingt nach dem Laden mit DMBS_STATS berechnet werden. Der Default-Statistikjob, der fehlende Statistiken nachträglich berechnet, wird im ADW nicht ausgeführt. Deshalb wird empfohlen, entweder nach jedem Ladejob DBMS_STATS aufzurufen oder alle Constraints mittels RELY DISABLE NOVALIDATE anzulegen.

Also aufgepasst: Das ADW ist zwar so konfiguriert, dass effiziente Ladeprozesse möglich sind, bei falscher Verwendung können allerdings ungeahnte Nebenwirkungen auftreten.

Abfrageperformance

Für die effiziente Ausführung von Abfragen verwendet das ADW verschiedene Features der Oracle-Datenbank. Auch die Abfragen werden in den Services HIGH und MEDIUM parallel ausgeführt. Wie in Abbildung 3 ersichtlich, kann somit die Antwortzeit durch eine höhere Anzahl von CPU Cores reduziert werden. Dies gilt jedoch nur für Abfragen unter dem Service HIGH. Für Service MEDIUM werden maximal 4 parallele Prozesse verwendet, für LOW erfolgt die Ausführung seriell – die Antwortzeit ist somit unabhängig von der Anzahl der CPU Cores. Für die Tests wurde eine Query auf ein Star Schema des TPC-DS-Benchmarks verwendet.

Bei hoch-aggregierten Abfragen, die nur wenige Datensätze zurückgeben, fällt

auf, dass ab der zweiten Ausführung die Antwortzeit sehr kurz ist. Dies liegt am Result Cache, der im ADW immer eingeschaltet ist. Für mehrmals ausgeführte SQL-Abfragen muss somit die Resultatmenge nur einmal ermittelt werden. Ihre Wiederverwendung ist anschließend möglich, bis sich die Daten ändern oder neue Daten hinzugeladen werden. Vor allem für Standard-Reports kann dies ein wesentlicher Vorteil sein.

Interessant ist die Optimierung von Queries auf Star Schemas und Snowflake Schemas. Hier bietet Oracle seit Jahren die Möglichkeit der sogenannten „Star Transformation“ an, eine spezielle Optimierung für Abfragen auf eine Faktentabelle mit mehreren Dimensionen. Die Star Transformation wird im ADW aus zwei Gründen nicht verwendet: Erstens werden standardmäßig keine Bitmap-Indizes auf die Dimensionsschlüssel angelegt (dies ist eine wichtige Voraussetzung für die Star Transformation), zweitens ist der Parameter STAR_TRANSFORMATION_ENABLED auf FALSE gesetzt und kann auch nicht geändert werden.

Stattdessen kommt im ADW die „Vector Transformation“ zum Einsatz. Sie ist in den meisten Fällen effizienter als die Star Transformation und benötigt keine Indizes. Die Vector Transformation wurde mit Version 12.1.0.2 für die Oracle Database In-Memory Option eingeführt und wird dort für die In-Memory Aggregation verwendet. Im ADW wird sie verwendet, obwohl hier momentan der In-Memory Column Store auf dem Datenbankserver nicht zum Einsatz kommt.

In größeren Data Warehouses wird oft Partitionierung verwendet, um Abfragen mit Einschränkungen auf Datums-

intervalle durch „Partition Pruning“ zu beschleunigen. Die manuelle Partitionierung von Tabellen ist möglich; unterstützt werden wie gewohnt alle Partitionierungsmethoden. Somit liegt es im Ermessen der DWH-Entwickler oder Datenbankadministratoren, eine optimale Partitionierungsstrategie zu implementieren. Eine automatische (bzw. „autonome“) Partitionierung von Tabellen steht zumindest zum heutigen Zeitpunkt nicht zur Verfügung.

Auch hier gilt: Die Konfiguration von ADW ist auf eine hohe Effizienz von Abfragen optimiert, es braucht aber nach wie vor spezifisches Know-how, um eine optimale Abfrageperformance zu erreichen.

Performance Monitoring

Die ADW-Datenbank und die Performance der ausgeführten SQL-Statements können über die „Service Console“ in einem Webbrowser überwacht werden (siehe

Abbildung 4). Hier stehen verschiedene Übersichtsgrafiken über Speicherbedarf, CPU-Auslastung, Disk-I/O-Statistiken und Wait Events zur Verfügung. Die ausgeführten SQL-Befehle sowie ihre Execution Plans können ebenfalls im Browser angezeigt werden. Bei Bedarf kann ein SQL Monitoring Report für die einzelnen SQL-Befehle exportiert werden.

Wem diese Informationen nicht genügen, der hat auch die Möglichkeit, AWR und ASH Reports zu erstellen. Zu beachten ist dabei jedoch, dass diese ausschließlich auf der Pluggable Database (PDB) ausgeführt werden können, in der das ADW läuft. Der Zugriff auf die Container Database (CDB) ist nicht möglich.

Fazit

Mit dem Oracle ADW steht eine effiziente, skalierbare und gut konfigurierte DWH-Plattform in der Oracle Cloud zur Verfügung. Viele Aufgaben werden

vereinfacht oder fallen ganz weg. Die Datenbankadministration ist deutlich einfacher als bei einer herkömmlichen Oracle-Datenbank (On-Premises oder in der Cloud), hat aber auch gewisse Einschränkungen, da nicht alle Features zur Verfügung stehen. Automatisches Performance Tuning oder ähnliche in der Werbung für ADW erwähnte Features konnten wir bisher keine feststellen.

Ein Nachteil aus unserer Sicht ist, dass man als Anwender von ADW keine Kontrolle über geplante Patches und Upgrades hat. Obwohl es ein Vorteil ist, dass man sich nicht selber um die Installation von Patches kümmern muss, kann es für den produktiven Betrieb eines Data Warehouse ein Risiko darstellen, wenn Patches ohne Ankündigung und Tests installiert werden. Hinzu kommt, dass die Unterstützung durch den Oracle Support momentan noch nicht sehr gut eingespielt ist, da die Support-Mitarbeiter selber keinen Zugriff auf die ADW-Plattformen haben. Dies führt teil-

MUNIQSOFT
CONSULTING

Consulting

Hochverfügbarkeit mit IQ

Sicherheit vor teuren Ausfallzeiten:

Mit dem richtigen Konzept sind Ihre Daten und Server vor Systemausfällen optimal geschützt.

Nutzen Sie die Erfahrung der Muniqsoft Consulting GmbH
www.muniqsoft-consulting.de

ORACLE® Gold Partner

Specialized
Oracle Database



Jetzt Beratungstermin vereinbaren:
+49 89 62286789-39

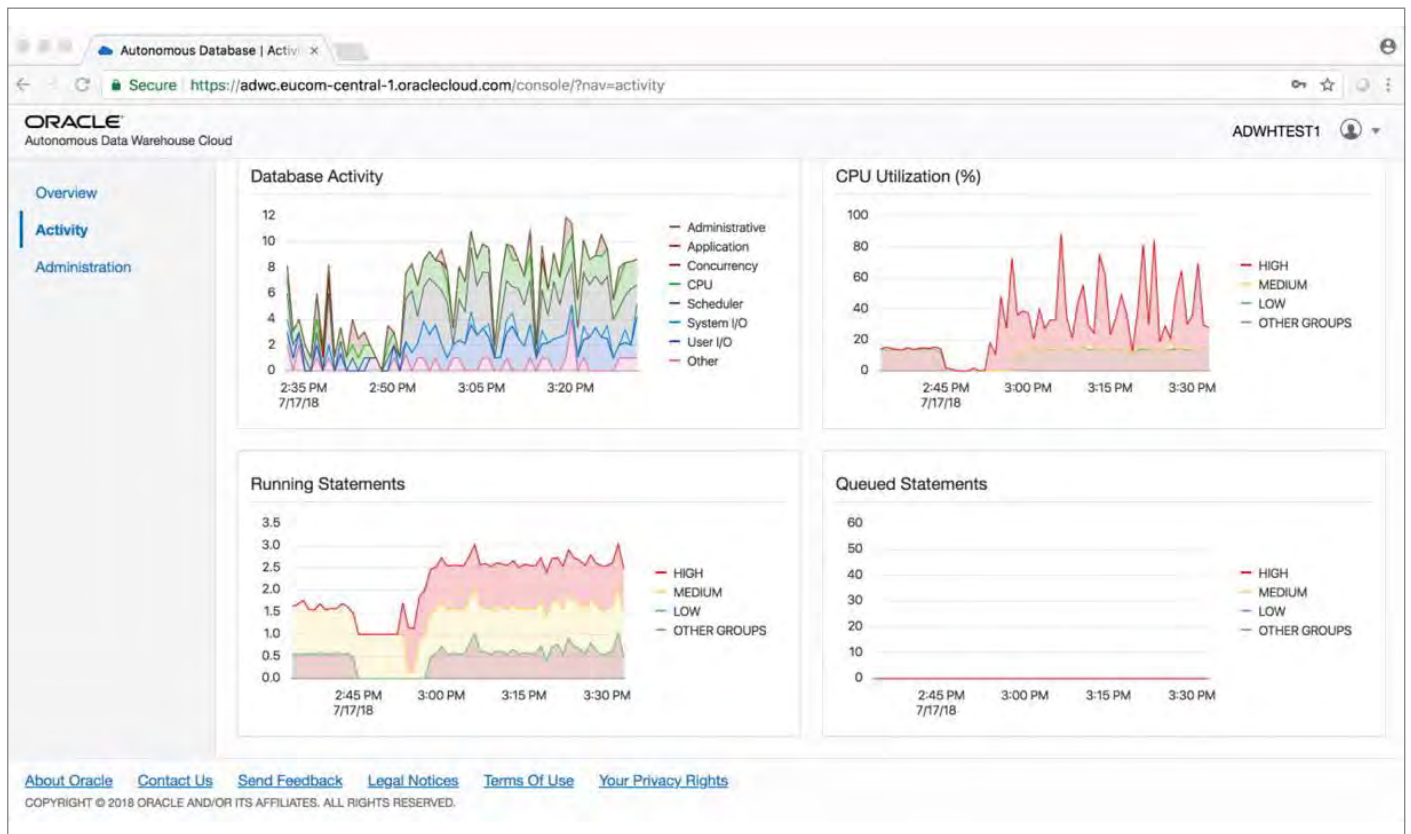


Abbildung 4: ADW Service Console mit einer Übersicht über die Datenbankaktivitäten

weise zu langen Reaktionszeiten, was bei einem produktiven Betrieb problematisch sein kann.

Abgesehen von kleinen Mängeln, die im Laufe der Zeit sicher behoben werden, lassen sich mit dem ADW effiziente und zuverlässige DWH-Systeme aufbauen und betreiben. Dazu ist allerdings nach wie vor Know-how über DWH-Architektur, physisches Datenbankdesign und Performance-Optimierung notwendig.

Weiterführende Informationen

Im Rahmen der intensiven Tests, die wir über drei Monate mit dem ADW durchgeführt haben, sind wir auf viele Detailaspekte gestoßen, die im vorliegenden Artikel und im Vortrag auf der DOAG-Konferenz nicht oder nur kurz erwähnt werden. Weitergehende Informationen können in folgenden Blogbeiträgen nachgelesen werden:

- DBMS_CLOUD Package – A Reference Guide: https://antognini.ch/2018/07/dbms_cloud-package-a-reference-guide/
- Which Privileges Are Required to Use

the ADWC Service Console?: <https://antognini.ch/2018/07/which-privileges-are-required-to-use-the-adwc-service-console/>

- Observations About the Scalability of Data Loads in ADWC: <https://antognini.ch/2018/07/observations-about-the-scalability-of-data-loads-in-adwc/>
- External Tables in Autonomous Data Warehouse Cloud: <https://danischnider.wordpress.com/2018/07/04/external-tables-in-autonomous-data-warehouse-cloud/>
- Gathering Statistics in the Autonomous Data Warehouse Cloud: <https://danischnider.wordpress.com/2018/07/11/gathering-statistics-in-the-autonomous-data-warehouse-cloud/>
- 10 Tips to Improve ETL Performance – Revised for ADWC: <https://danischnider.wordpress.com/2018/07/20/10-tips-to-improve-etl-performance-revised-for-adwc/>
- Star Schema Optimization in Autonomous Data Warehouse Cloud: <https://danischnider.wordpress.com/2018/09/13/star-schema-optimization-in-autonomous-data-warehouse-cloud/>



Dani Schnider
Dani.Schnider@trivadis.com



Christian Antognini
Christian.Antognini@trivadis.com



Terraform: Infrastructure as Code in der Oracle Cloud Infrastructure

Marcus Schröder, Oracle Deutschland B.V. & Co. KG

In einer Sache sind sich die Fachleute einig, einer der Hauptgründe für Cloud Computing ist die hohe Agilität. Umgebungsbereitstellungen erfolgen automatisch und standardisiert, die Konfiguration ist verglichen mit einer manuellen Installation stark vereinfacht. Allerdings müssen auch im Cloud-Infrastruktur-Bereich die Infrastruktur-Komponenten wie Compute, Netzwerk und Storage nacheinander und in der richtigen Reihenfolge angelegt werden. Beim Erzeugen von komplexeren Umgebungen steigt auch beim Cloud-Computing an dieser Stelle der Aufwand.

Die großen Cloud-Provider bieten mittlerweile ein REST-API an, in dem die Konfiguration mit Skripten über eine Schnittstelle durchgeführt werden kann.

Das hat neben den offensichtlichen Vorteilen auch eine Reihe von Nachteilen: Skripte sind relativ unflexibel und jeder Arbeitsschritt muss detailliert nachgestellt werden. Speziell beim Lö-

sen und Ändern der Umgebungen müssen für diese Arbeiten eigene REST-API-Skripte entwickelt werden. Das Benutzer-Management stellt eine große Herausforderung dar. Der technische Skript-Benutzer muss über ausreichende Rechte verfügen, um alle Arbeiten durchzuführen, darf aber nicht zu viele Rechte besitzen, da das Skript sonst ein

Sicherheitsrisiko darstellen könnte. Ein weiterer Punkt ist, dass Skripte meist auf eine spezielle Umgebung bzw. einen Anbieter beschränkt sind, d. h. der Aufwand des Skriptens ist auf eine proprietäre Umgebung beschränkt.

An dieser Stelle kommt Terraform ins Spiel. Terraform ermöglicht das Anlegen, Ändern und Löschen von komplexen Um-

gebungen in der Cloud. Terraform unterstützt die gängigen Cloud-Anbieter und lässt sich bei Bedarf für nahezu jede Umgebung anpassen. Dadurch ist es möglich, Cloud-Umgebungen bereitzustellen, die verschiedene Cloud-Anbieter beinhalten. Terraform fällt in die Kategorie „Infrastructure as Code“ (IaC) und wird durch einen unabhängigen Anbieter entwickelt und bereitgestellt.

Ein bisschen zur Geschichte von Terraform: Das Produkt wurde vom Entwickler und Videospieleer Mitchell Hashimoto erfunden. Angeblich kam ihm die Idee zu einer Automatisierungsplattform durch ein Projekt, in dem die Änderung an einem System einige Minuten gedauert hat, das Aufsetzen der Entwicklungsumgebung allerdings mehrere Stunden. Daraus entwickelte sich eine Reihe von Tools, die in der von Mitchell Hashimoto mitbegründeten Firma Hashicorp angeboten werden. Die Firma bietet in einem „Freemium“-Geschäftsmodell Open-Source- und kommerzielle Produkte an. Terraform wurde im Juli 2014 als fünftes Produkt der Firma Hashicorp „released“ (neben Tools wie Vagrant, Packer, Vault etc.).

Um verschiedene Szenarien abdecken zu können, bietet Terraform drei unterschiedliche Ressourcen-Typen: Provider, Provisioner und Modules. Provider agiert direkt mit den APIs der entsprechenden Cloud-Anbieter aus den Bereichen IaaS, PaaS und SaaS. Provider sind zum Beispiel Oracle OCI, Amazon AWS, Microsoft Azure, OpenStack etc. Provisioner werden dazu verwendet, um Skripte auf lokalen oder entfernten Maschinen auszuführen. Sie kommen ins Spiel, wenn die Maschinen bereitgestellt wurden, um zum Beispiel maschinenspezifische Installationen und Konfigurationen durchzuführen bzw. vor dem Löschen der Maschine bestimmte Software zu deinstallieren. Module sind sogenannte „Self-Contained-

Packages“ für Terraform, die als Gruppe administriert werden. Module agieren als wiederverwendbare Komponenten und werden zum Beispiel von den oben genannten Providern auf Source-Code-Plattformen wie GitHub, Bitbucket etc. bereitgestellt. Die Module werden in der Terraform-Registry zentral indiziert und können von dort lokalisiert werden. Auch die OCI bietet Module, um zum Beispiel Compute-Instanzen oder Virtual Cloud Networks bereitzustellen.

In der folgenden Beschreibung möchte ich Sie durch einen vollständigen Terraform-Zyklus führen. Anhand eines Praxisbeispiels werden die Punkte Installation, Konfiguration, Umgebungsbereitstellung, Änderungen und Löschen erläutert sowie viele Hintergrundinformationen vermittelt. Es gibt insgesamt vier Beispiele:

1. Auslesen der Availability-Domänen einer Region mit externer Authentifizierung
2. Auslesen der Availability-Domänen einer Region mit interner Authentifizierung
3. Erstellen einer Compute-Instanz
4. Erstellen einer Autonomous-Transaction-Processing-Instanz

Um das Listing möglichst klein zu halten, werden alle vier Beispiele in der Minimalkonfiguration erstellt. Es werden also nur die unbedingt benötigten Parameter übergeben, der Rest der Parameter wird automatisch mit den Standardwerten belegt. Diese Beispiele nutzen eine Compute-Instanz Oracle Linux 7.6 innerhalb der Oracle Cloud Infrastructure, dadurch müssen Sie keine Änderungen an Ihrer Workstation durchführen, die Beispiele funktionieren garantiert ohne OS-spezifische Eigenarten. Einen kostenlosen, 30-tägigen Trial kann man unter folgender URL beantra-

gen: <https://cloud.oracle.com/>. Klicken Sie „Try for Free“. Der Trial verlangt als Sicherheit und Authentifizierung eine Kreditkartennummer. Dieses Verfahren ist mittlerweile bei allen Cloud-Anbietern üblich und soll verhindern, dass mit den Trial-Accounts illegale Aktivitäten durchgeführt werden bzw. dass ein Benutzer eine Vielzahl von Trials mit falschen E-Mail-Adressen beantragt. Sie können für die Übungen natürlich auch Ihre eigene Hardware verwenden.

Terraform-Installation

Im ersten Schritt muss die Terraform-Software auf der Oracle-Linux-Instanz installiert und konfiguriert werden. Dazu verbindet man sich mit SSH (*siehe Listing 1*) auf die vorher erzeugte Instanz. Die nachfolgenden Befehle gelten für Unix-basierte Betriebssysteme; für Windows verwenden Sie bitte PUTTY.

Als Nächstes benötigt man das Terraform-Package für das verwendete Betriebssystem beziehungsweise die CPU-Architektur. Das Package ist die Laufzeitumgebung für die Nutzung von Terraform und für alle Cloud-Provider gleich. Die Terraform-Laufzeitumgebung ist in der Sprache Go geschrieben und nur wenige Megabyte groß. Um das passende Package auszuwählen, geht man auf die Homepage von Hashicorp (*siehe <https://www.hashicorp.com>*), wählt unter Products „Terraform“ und kopiert sich unter Download die URL für das benötigte Package. Das Package ist kostenlos und erfordert keine Benutzerregistrierung. Unter Linux kann man sich das Package mit folgendem Befehl (*siehe Listing 2*) direkt auf die Oracle-Linux-7.6-Instanz laden.

Um die Datei verwenden zu können, müssen die korrekten Rechte (*siehe Listing 3*) gesetzt werden. Nach dem Entpacken kann die Terraform-Laufzeitumgebung direkt aufgerufen werden (*siehe Listing 4*). Der Aufruf gibt eine Online-Hilfe zurück, die eine Übersicht über die Befehlsparameter gibt. Soll die Laufzeitumgebung

```
ssh -i <vollständigerPfad ssh key> opc@<public ip der instanz>
```

Listing 1

```
wget
https://releases.hashicorp.com/terraform/0.11.11/terraform_0.11.11_linux_amd64.zip
```

Listing 2

nicht nur aus dem Installationsverzeichnis aufrufbar sein, muss man auf Linux das Verzeichnis zur PATH-Variablen hinzufügen (siehe Listing 5).

Um diese Änderung dauerhaft zu machen, kann man auf Linux den Pfad auch in die „bash_profile“ einfügen.

1. Auslesen der Availability-Domänen einer Region mit externer Authentifizierung

Für einen ersten Test der korrekten Konfiguration lassen wir uns alle Availability-Domänen (ADs) in unserer Cloud-Umgebung ausgeben. Hierzu einige grundsätzliche Best-Practices: Die Konfigurationsbeschreibungen in Terraform werden in Dateien angegeben, diese haben die Endung „.tf“.

Umgebungsvariablen können in Umgebungsvariablen oder ebenfalls in einer Datei angegeben werden. Für das Abspeichern der Umgebungsvariablen in einer Datei muss diese die Bezeichnung „terraform.tfvars“ aufweisen. Wir erzeugen für dieses Beispiel drei Dateien. Die erste enthält Informationen zum Provider und Login-Daten, die zweite den Ablauf-Code für das Anzeigen der Ads und die dritte die Umgebungsvariablen.

Wir erzeugen uns im nächsten Schritt ein Übungsverzeichnis (siehe Listing 6), in dem die erste Terraform-Übung abgespeichert wird, und wechseln in dieses Verzeichnis.

Bis zu dem letzten Schritt ist Terraform noch Provider-unabhängig, jetzt müssen wir Terraform für die Nutzung auf der Oracle Cloud Infrastructure (OCI) konfigurieren. Das Herunterladen des Providers erfolgt normalerweise zur Laufzeit aus dem Internet, der Provider kann aber auch lokal installiert werden. Für die Initialisierung des Providers und der Ausführung zukünftiger Skripte auf den korrekten Host wird manuell eine „provider.tf“-Datei erzeugt. (siehe Listing 7)

In dieser Datei stehen folgende Informationen: Provider-Typ ist in diesem Fall OCI. Werden keine weiteren Informationen übergeben, lädt sich Terraform den aktuellsten OCI-Provider herunter. Die folgenden Informationen sind OCI-spezifisch und werden an verschiedenen Stellen in der OCI-Cloud gesammelt. Die OCIDs stellen sicher, dass der korrekte

```
chmod u+x ./terraform_0.11.11_linux_amd64.zip
```

Listing 3

```
unzip ./terraform_0.11.11_linux_amd64.zip
./terraform
```

Listing 4

```
export PATH=$PATH:<pfad-zur-terraform-datei>
```

Listing 5

```
mkdir list_ads
```

Listing 6

```
Inhalt Datei provider.tf: Provider-Informationen inklusive Login-Daten
# Configure the OCI provider with an API Key
# Using the variables in variables.tfvars
variable "tenancy_ocid"      {}
variable "user_ocid"        {}
variable "fingerprint"     {}
variable "private_key_path" {}
variable "region"           {}

# Configure the OCI provider with an API Key
provider "oci" {
  tenancy_ocid = "${var.tenancy_ocid}"
  user_ocid   = "${var.user_ocid}"
  fingerprint = "${var.fingerprint}"
  private_key_path = "${var.private_key_path}"
  region     = "${var.region}"
}
```

Listing 7

Tenant bzw. Benutzer verwendet wird. Die „tenancy_oid“ bekommt man in der OCI-Console im Bereich „Administration -> Tenancy Details“, die „user_ocid“ unter den Benutzerinformationen. Es wird eine ID verwendet, da es in der Oracle Cloud Infrastructure möglich ist, Anzeigenamen zu benutzen, und es daher zu Doppelungen kommen kann. Zur sicheren Verbindung wird ein RSA-Key-Paar im PEM-Format (Minimum 2048 Bits) benötigt, davon wird der Public-Teil unter den Benutzer-API-Keys gespeichert, der resultierende Fingerprint wird für die Variable „fingerprint“ verwendet. Die Regionsbezeichnung findet man im oberen rechten Teil der Console. Unter folgender URL ist beschrieben, wo Sie welche Werte finden und wie die Keys erzeugt werden können:

[„https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm“](https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm).

TIPP: Zur Erzeugung des PEM-RSA-Key-Paars erstellt man im ersten Schritt ein Verzeichnis und erzeugt dort die nötigen Schlüssel. In unserem Beispiel in Listing 8 werden der Private- und Public-Key ohne Passwort-Phrase erstellt.

Anschließend lässt man sich die Datei anzeigen und kopiert den Inhalt (siehe Listing 9) im Bereich „API-Keys“ unter dem eingeloggten Benutzer. Damit man nur selbst den Private-Key lesen kann, sollte man aus Sicherheitsgründen noch die Zugriffsrechte ändern.

Listing 10 zeigt den Inhalt der Datei „listADs.tf“. In der Datei wird die OCID für den Cloud-Tenant übergeben. Anschließend werden aus der Data Source für

```
mkdir ~/.oci

openssl genrsa -out ~/.oci/oci_api_key.pem 2048

openssl rsa -pubout -in ~/.oci/oci_api_key.pem -out ~/.oci/oci_api_key_public.pem
```

Listing 8

```
chmod go-rwx ~/.oci/oci_api_key.pem
```

Listing 9

```
# Get a list of Availability Domains
data "oci_identity_availability_domains" "ads" {
  compartment_id = "${var.tenancy_ocid}"
}

# Output the result
output "show-ads" {
  value = "${data.oci_identity_availability_domains.ads.availability_domains}"
}
```

Listing 10

den Tenant die benötigten Informationen ausgelesen und ausgegeben. An dieser Stelle noch ein wichtiger Hinweis: Alle Ressourcen sind auf der Hashicorp-Seite detailliert beschrieben, auch welche Parameter optional sind und welche nicht. Eine Beschreibung dazu befindet sich unter „<https://www.terraform.io/docs/providers/oci/index.html>“. Hier werden auf der linken Seite alle Ressourcen aufgelistet. An dieser Stelle ein sehr wichtiger Hinweis: Im linken oberen Teil werden erst die Data Sources beschrieben, also die Ressourcen-Objekte, die bestimmte Informationen beinhalten, wenn diese bereits instanziiert sind. Für die Erstellung von Ressourcen stehen weiter unten auf der linken Seite die entsprechenden Ressourcentypen (Audit, Core, IDM, DB etc.); hier wird beschrieben, wie und mit welchem Parameter die Ressourcen erzeugt werden können.

Als dritte Datei muss die Variablendatei „`terraform.tfvars`“ angelegt werden. Wie Sie zu den benötigten Informationen kommen, steht oben im Text. Ich habe die Informationen anonymisiert, da mit den Informationen und dem korrekten Schlüssel auf den jeweiligen Cloud-Tenant zugegriffen werden kann. Auch an dieser Stelle noch einmal der Hinweis: Die Datei für die Variablen muss „`terra-`

```
tenancy_ocid = "ocid1.tenancy.oc1..geheimeTenanyOCID"
user_ocid    = "ocid1.user.oc1..geheimeUserOCID"
fingerprint  = "9d:cb:a6:f8:32:geheimerFingerprint"
private_key_path = "~/oci/pfadUndNameZurPrivateKeyDatei"
region       = "eu-frankfurt-1-DieRegionFrankfurt"
```

Listing 11

form.tfvars“ heißen, ansonsten wird die Datei nicht erkannt und nicht verwendet.

Listing 11 zeigt den Inhalt der Datei „`terraform.tfvars`“. Hat man diese Dateien erzeugt und abgespeichert, geht es um den Start von Terraform. Im ersten Schritt wird der Service-Provider initialisiert, in diesem Fall ist das die Oracle Cloud Infrastructure (siehe Listing 12). Gibt man keine weiteren Informationen an, wird der Service-Provider zur Laufzeit geladen und verwendet die aktuelle Version. Der Provider liegt anschließend im

```
terraform init
```

Listing 12

```
terraform plan
```

Listing 13

```
terraform apply
```

Listing 14

```
# Using the variables in variables.tfvars
variable "region" {}

# Configure the OCI provider use Instance
provider "oci" {
  auth = "InstancePrincipal"
  region = "${var.region}"
}
```

Listing 15

```

[opc@terraform-client list_ads]$ terraform apply
data.oci_identity_availability_domains.ads: Refreshing state...

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

show-ads = [
  {
    compartment_id = ocid1.tenancy.oc1..aaaaaaaap7sg74i4ibhxz7ymm343ztoyytoi3sp5bwj4r6ydz6fscx6nuka,
    id = ocid1.availabilitydomain.oc1..aaaaaaaaiifj24st3w4j7cowuo3pmqcuqwjapjv435vtjmg5j7q3flguwna,
    name = mQqW:EU-FRANKFURT-1-AD-1
  },
  {
    compartment_id = ocid1.tenancy.oc1..aaaaaaaap7sg74i4ibhxz7ymm343ztoyytoi3sp5bwj4r6ydz6fscx6nuka,
    id = ocid1.availabilitydomain.oc1..aaaaaaaalcdcl7u6akbmkobjxhrozpj2v7yavqqydkj3ytyjbt47Lnoqnm2q,
    name = mQqW:EU-FRANKFURT-1-AD-2
  },
  {
    compartment_id = ocid1.tenancy.oc1..aaaaaaaap7sg74i4ibhxz7ymm343ztoyytoi3sp5bwj4r6ydz6fscx6nuka,
    id = ocid1.availabilitydomain.oc1..aaaaaaaalcdcl7u6akbmkobjxhrozpj2v7yavqqydkj3ytyjbt47Lnoqnm2q,
    name = mQqW:EU-FRANKFURT-1-AD-3
  }
]
[opc@terraform-client list_ads]$ █

```

Abbildung 1: Auflistung der Availability Domains in der Region Frankfurt

Verzeichnis „<user_home>/terraform_example/.terraform/plugins/linux_amd64“. Die Initialisierung und das Laden des Providers erfolgt im Übungsverzeichnis mit dem Befehl.

Die Ausgabe beschreibt die erfolgreiche Initialisierung und verwendete Provider-Version. Im nächsten Schritt können wir die Ausführung simulieren. Diese Simulation dient der Überprüfung der korrekten Syntax und ermöglicht einen Überblick über die geplanten Änderungen/Tätigkeiten von Terraform. Da in unserem Beispiel (siehe Listing 13) etwas ausgelesen wird, bekommen wir keine Änderungen angezeigt.

Im letzten Schritt lassen wir das Auflisten der Availability-Domänen laufen. Die Datei „listADs.tf“ wird ausgeführt und liefert das Ergebnis in den Terminal-Output. (siehe Listing 14). Da von Terraform keinerlei Ressourcen angelegt werden, ist auch hier keine Veränderung sichtbar. Es werden nur die vorhandenen Availability-Domänen in der Oracle Cloud Infrastructure Region Frankfurt angezeigt.

Diese Befehle können Sie beliebig oft wiederholen, da in der Umgebung nichts verändert wird und die Initialisierung bzw. Planung beliebig oft durchgeführt werden kann (siehe Abbildung 1).

Durch diesen Probelauf haben wir sichergestellt, dass die Konfigurationsparameter korrekt angelegt wurden und wir uns in die Oracle Cloud Infrastructure einloggen können.

2. Auslesen der Availability-Domänen einer Region mit interner Authentifizierung

Das obige Beispiel zeigt die Konfiguration und Run-Time von Terraform. Befinden Sie sich mit Ihrer Client-Instanz innerhalb der Oracle Cloud Infrastructure, können Sie die Authentifizierung wesentlich kürzer gestalten, mit der sogenannten Instance Principal Authentication. Legen Sie dazu ein neues Verzeichnis „list_ads_principals“ an und kopieren und editieren Sie die drei Dateien in das neue Verzeichnis. Es reichen die folgenden Zeilen in der „provider.tf“-Datei. Hier der Vollständigkeit halber die Listings der drei Dateien:

Listing 15 zeigt den Inhalt der Datei „provider.tf“: Provider-Informationen inklusive Login-Daten. Listing 16 zeigt den Inhalt der Datei „listADs.tf“: Ablauf-Code. Listing 17 zeigt den Inhalt der Datei „terraform.tfvars“.

Dieses Beispiel zeigt ein neues Feature der OCI, das Arbeiten mit Principal-Authentifizierung muss jedoch noch in der Oracle-Cloud-Infrastruktur aufgesetzt werden. Bei simplen Abfragen funktioniert es auch ohne spezielles Setup, allerdings sind zum Erstellen von

```

# Using the variables in variables.tfvars
variable "tenancy_ocid" {}

# Get a list of Availability Domains
data "oci_identity_availability_domains" "ads" {
  compartment_id = "${var.tenancy_ocid}"
}

# Output the result
output "show-ads" {
  value = "${data.oci_identity_availability_domains.ads.availability_domains}"
}

```

Listing 16

```

# Provider and Tenancy Details
region          = "eu-frankfurt-1"
tenancy_ocid    = "ocid1.tenancy.oc1.ocid1..TenancyOCID"

```

Listing 17

Ressourcen zusätzliche Schritte notwendig. Daher arbeiten wir in den weiteren Beispielen mit der Standard-Authentifizierung.

Noch ein Hinweis: In der „provider.tf“-Datei wird die „compartment_id“ anstatt der „tenant_id“ angegeben, da die „tenant_id“ eigentlich die Root-Compartment-Id ist und wir direkt in unser Arbeits-Compartment wollen. Daher können wir gleich unsere „compartment_id“ angeben.

3. Erstellen einer Compute-Instanz

In dem folgenden Beispiel wird eine Oracle-Cloud-Infrastructure-Compute-Instanz erzeugt. Voraussetzung ist, dass im ersten Schritt die Netzwerkkomponenten erstellt und konfiguriert werden. Die einzelnen Komponenten sind: Virtual

```
# Using the variables in variables.tfvars
variable "tenancy_ocid"      {}
variable "user_ocid"        {}
variable "fingerprint"      {}
variable "private_key_path" {}
variable "region"           {}
variable "compartment_ocid" {}

# Configure the OCI provider with an API Key
provider "oci" {
  tenancy_ocid = "${var.tenancy_ocid}"
  user_ocid    = "${var.user_ocid}"
  fingerprint  = "${var.fingerprint}"
  private_key_path = "${var.private_key_path}"
  region       = "${var.region}"
}
```

Listing 18

```
# Using the variables in variables.tfvars
variable "AD"          {}
variable "vcn-cidr"    {}

# ----- get the list of available ADs
data "oci_identity_availability_domains" "ADs" {
  compartment_id = "${var.tenancy_ocid}"
}

resource "oci_core_virtual_network" "terra-vcn" {
  cidr_block = "${var.vcn-cidr}"
  compartment_id = "${var.compartment_ocid}"
  display_name = "terraform-vcn"
}

# ----- Create a new Internet Gateway
resource "oci_core_internet_gateway" "terra-ig" {
  compartment_id = "${var.compartment_ocid}"
  display_name = "terraform-internet-gateway"
  vcn_id = "${oci_core_virtual_network.terra-vcn.id}"
}

# ----- Create a new Routing Table
resource "oci_core_default_route_table" "default_route_table" {
  manage_default_resource_id = "${oci_core_virtual_network.terra-vcn.default_route_table_id}"
  route_rules {
    network_entity_id = "${oci_core_internet_gateway.terra-ig.id}"
    destination        = "0.0.0.0/0"
    destination_type   = "CIDR_BLOCK"
  }
}

# ----- Create a public subnet in the new VCN
resource "oci_core_subnet" "terra-public-subnet" {
  availability_domain = "${lookup(data.oci_identity_availability_domains.ADs.availability_domains[var.AD - 1], "name")}"
  cidr_block          = "10.0.1.0/24"
  display_name        = "terraform-public-subnet"
  compartment_id      = "${var.compartment_ocid}"
  vcn_id              = "${oci_core_virtual_network.terra-vcn.id}"
}
```

Listing 19

Cloud Network – kurz VCN, Subnetze, Gateways, Security-Listen, DHCP-Optionen und Routing Table.

Einige dieser Komponenten werden bei Erzeugung eines Subnetzes automatisch mit einer Default-Instanz angelegt, dies betrifft die Security-Listen, DHCP-Optionen und Routing Table. Um das Listing möglichst „schlank“ zu halten, benötigen wir für diese Komponenten keine zusätzliche Konfigurationsbeschreibung, sie werden mit einer Default-Konfiguration angelegt.

Listing 18 zeigt den Inhalt der Datei „provider.tf“. Dieses Datei-Listing ist wie im obigen Beispiel „Auslesen der Availability-Domänen einer Region mit externer Authentifizierung“.

Anlegen des Virtual Cloud Network (VCN) inklusive aller Komponenten. Das VCN ist Voraussetzung für das Erzeugen einer Compute-Instanz und enthält alle Ressourcen, um sich mit der Compute-Instanz zu verbinden.

Listing 19 zeigt den Inhalt der Datei „vcn_subnet_ig.tf“.

Listing 19 im Detail:

Im oberen Teil werden die benötigten Variablen deklariert; die Werte für diese Variablen werden in der zentralen Variablen-datei (terraform.tfvars) definiert. Im nächsten Teil werden die in der Region verfügbaren Availability-Domänen abgefragt und in die Data-Source der ADs geschrieben. Bei der Erstellung des Virtual Cloud Network (VCN) werden der Netzwerkbereich (CIDR), das DNS-Label

```
variable "instance_shape" {}
variable "instance_image_ocid" {}
variable "ssh_public_key" {}
variable "InstanceDisplayName" {}

resource "oci_core_instance" "terra_Instance" {
  availability_domain = "${lookup(data.oci_identity_availability_domains.ADs.availability_domains[var.AD - 1], "name")}"
  compartment_id      = "${var.compartment_ocid}"
  display_name        = "${var.InstanceDisplayName}"
  shape               = "${var.instance_shape}"

  create_vnic_details {
    subnet_id = "${oci_core_subnet.terra-public-subnet.id}"
  }

  source_details {
    source_type = "image"
    source_id   = "${var.instance_image_ocid}"
  }

  metadata {
    ssh_authorized_keys = "${trimspace(file(var.ssh_public_key))}"
  }
}
```

Listing 20

und der Anzeigename angegeben. Damit sich die Außenwelt mit der zu erstellenden Compute-Instanz verbinden kann, benötigt das VCN ein Gateway. Es gibt je nach Verbindungstyp ein spezielles Gateway. In unserem Fall wird die Verbindung über das Internet aufgebaut, daher verwenden wir ein Internet Gateway. Zum Schluss wird in dem VCN ein Subnetz angelegt. Bei der Erstellung des

Subnetzes werden – wie eben beschrieben – DHCP-Optionen und Security-Listen per Default erzeugt. Die Security-Listen sind so konfiguriert, dass man sich über SSH-Protokoll am Port 22 mit den Komponenten im VCN verbinden kann. Der Routing Table muss so angepasst werden, dass das Routing vom Internet Gateway zum entsprechenden Host funktioniert.

```
# Provider and Tenancy Details
compartment_ocid = "ocidl.compartment.oc1..geheimeCompartmentOCID"
tenancy_ocid     = "ocidl.tenancy.oc1..geheimeTenancyOCID"
user_ocid        = "ocidl.user.oc1..geheimeUserOCID"
fingerprint      = "9d:cb:a6:f8:32:geheimerFingerprint"
private_key_path = "~/oci/pfadUndNameZurPrivateKeyDatei"
region           = "eu-frankfurt-1-DieRegionFrankfurt"

# Availability Domain
AD              = "1"
vcn-cidr        = "10.0.0.0/16"

# Instance Variables
instance_shape = "VM.Standard2.1"
instance_image_ocid = "ocidl.image.oc1.eu-frankfurt-1.aaaaaaaamrvusixu33rzvgvjpfjflwkjeyfwnnoyoefoqxnmttds5vukj4vq"
ssh_public_key = "./Keys/publickey.pub"
InstanceDisplayName = "Terraform_Instance"
```

Listing 21

Anlegen der Compute-Instanz

Im letzten Schritt wird eine Compute-Instanz mit Oracle Linux angelegt. Die Konfiguration der Instanz bietet die gleichen Parameter wie die grafische Benutzeroberfläche. Es können jedoch eine Reihe von Parametern weggelassen werden, wenn diese den Default-Wert behalten sollen. Die Compute-Instanz wird mit der folgenden Konfiguration erzeugt: *Listing 20* zeigt den Inhalt der Datei „instance.tf“

Listing 20 im Detail:

In diesem Listing werden nur die minimal nötigen Parameter angegeben, der Rest der Ressourcen wird automatisch mit Standardparametern versehen. Hier die Erklärung der wichtigsten Parameter: Shape ist die Größe der VM, in unserem Fall verwenden wir die neueste Servergeneration mit 1 OPCU/Core und 15 GB RAM. Der Virtual Network Interface Controller VNIC wird mit Standard-Parametern angelegt. In den Source-Details sind Anga-

```
terraform destroy
```

Listing 22

```
terraform apply
```

Listing 23

ben zum verwendeten OS-Image, in unserem Fall verwenden wir die neueste Version des Oracle Linux 7.6. Eine Auflistung aller Images gibt es in der Oracle-Cloud-Infrastruktur-Dokumentation unter „<https://docs.cloud.oracle.com/iaas/images/>“. Die Verbindung zur Instanz erfolgt über ein SSH-Private-Public-Key-Paar, die Lokation des Public-Keys wird in den Metadaten angegeben.

Die Variablen-Datei wird in diesem Fall mit den benötigten Variablen angeichert und sieht folgendermaßen aus: *Listing 21* zeigt den Inhalt der Datei „terraform.tfvars“: Das Erzeugen der Ins-

tanz erfolgt in dem Verzeichnis, in dem die „*.tf“- und „terraform.tfvars“-Datei liegen. Wie bei dem ersten Beispiel beschrieben, wird die Instanz mit der Kombination von terraform init, terraform plan und terraform apply erzeugt. *Listing 22* zeigt den Befehl für das Löschen der Instanz. Dieser Befehl löscht alle erzeugten Ressourcen, also Compute-Instanz und Netzwerk-Ressourcen.

Wenn man Änderungen in der Beschreibung durchführen möchte, erfolgt das direkt im Text. Anschließend kann man mit dem Befehl aus *Listing 23* die Änderungen durchführen lassen. Sie können in unserem Beispiel die Variable InstanceDisplayName verändern und anschließend mit terraform apply übernehmen. Nach Durchlauf des Programms ist der Anzeigename nach Ihren Vorgaben verändert.

4. Erstellen einer Autonomous-Transaction-Processing-Instanz

Zum Schluss kommen wir zu einer leichten Übung. Um den Oracle-DB-Teil zu repräsentieren, erzeugen wir eine Autonomous-Transaction-Processing-Instanz. Diese Bereitstellung kommt ebenfalls mit wenig Pflichtparametern aus. Um die anderen Terraform-Projekte nicht zu mischen, legen wir ein neues Verzeichnis an und kopieren die „provider.tf“- und die „terraform.tfvars“-Dateien in das neue Verzeichnis. Die „provider.tf“ kann eins zu eins übernommen werden, wir müssen keine Anpassungen vornehmen.

Anschließend erzeugen wir eine neue Datei, in diesem Fall mit den Namen „autonomous_ATP.tf“, sie sieht folgendermaßen aus: *Listing 24* zeigt den Inhalt der Datei „autonomous_ATP.tf“: Das Listing im Detail: Für die ATP werden wenige Parameter benötigt; das Admin-Passwort, Anzahl der Cores/OCPUs, Storage-Größe und DB-Namen sind alle Parameter, die angegeben werden müssen. In dem Fall ATP wird die Variablen-Datei „terraform.tfvars“ folgendermaßen geändert: *Listing 25* zeigt den Inhalt der Datei „terraform.tfvars“: Anschließend kann nach dem bewährten Verfahren (terraform init, plan, apply) die Instanz angelegt werden bzw. mit terraform destroy gelöscht werden.

```
# Provider and Tenancy Details
variable "db_admin_password" {}
variable "db_cpu_count"      {}
variable "db_storage_size"   {}
variable "db_name"           {}

resource "oci_database_autonomous_database" "terraform_autonomous_
database" {
  #Required
  admin_password = "${var.db_admin_password}"
  compartment_id = "${var.compartment_ocid}"
  cpu_core_count = "${var.db_cpu_count}"
  data_storage_size_in_tbs = "${var.db_storage_size}"
  db_name = "${var.db_name}"
}
```

Listing 24

```
# Provider and Tenancy Details
compartment_ocid = "ocidl.compartment.oc1..geheimeCompartmentOCI"
tenancy_ocid     = "ocidl.tenancy.oc1..geheimeTenanyOCID"
user_ocid        = "ocidl.user.oc1..geheimeUserOCID"
fingerprint      = "9d:cb:a6:f8:32:geheimerFingerprint"
private_key_path = "~/oci/pfadUndNameZurPrivateKeyDatei"
region           = "eu-frankfurt-1-DieRegionFrankfurt"

# Database Infos
db_admin_password = "<DBpasswortNachOracleVorgaben>"
db_cpu_count      = "1"
db_storage_size   = "1"
db_name           = "TerraformDB"
```

Listing 25

Ein paar Tipps:

Der Debug-Modus von Terraform ist sehr einfach einzuschalten, sie setzen eine Umgebungsvariable und die Ausgabe ist entsprechend. Die Umgebungsvariable heißt TF_LOG und hat die möglichen Parameter TRACE, DEBUG, INFO, WARN oder ERROR. Nach dem Angeben eines dieser Parameter werden die Informationen im Standard-Out von Terraform wiedergegeben.

Wie schon in den Konfigurations-Listings gesehen, werden die Zeilen mit # auskommentiert; wenn es ein größerer Bereich sein soll, verwendet man /*, das wird auskommentiert */. Wer keinen geeigneten Text-Editor hat, kann für die Beschreibung der Terraform-Konfigurations-Dateien das Tool „Sublime“ verwenden. Für Sublime gibt es ein spezielles Terraform-Modul, das die Schlüsselwörter entsprechend markiert.

Fazit

Terraform ist eine relativ einfach zu beherrschende Konfigurationssprache, die Installation ist schnell und unkompliziert, das Erzeugen von Instanzen ist bewusst einfach gehalten. Die Dokumentation auf der Hashicorp-Seite ist sehr detailliert und ermöglicht es, die Minimalkonfiguration schnell auszutesten. Terraform ist ein sehr mächtiges Tool für die Erstellung, Veränderung und Löschung komplexer Umgebungen. Terraform hat sich mittlerweile quasi als Standard im Bereich Infrastructure as Code etabliert. Die einzelnen Beschreibungen von Ressourcen der verschiedenen Hersteller sind zwar nicht identisch, das Vorgehen bei der Bereitstellung ist allerdings gleich, daher lohnt es sich, in der Lernphase in nur ein Tool zu investieren. Ich hoffe, Sie konnten die gezeigten Beispiele ausführen und hatten Spaß an der Lektüre dieses Artikels.



Marcus Schröder
marcus.schroeder@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 · Bern · Nyon · Zürich · dbi-services.com



Universal Credits: Einfacher Kauf und Verbrauch von Oracle IaaS und PaaS Cloud Services

Gertrud Abram-Retschitzegger, Oracle Deutschland B.V. & Co. KG

Viele Kunden sind an einem Wechsel in die Cloud interessiert, stellen sich aber folgende Fragen: Wir wollen in die Cloud, aber wie sieht es mit unseren aktuellen On-Premises-Lizenzen (Investitionssicherheit) aus? Wie sieht das „neue“ Cloud-Lizenzierungsmodell von Oracle aus und für welche Produkte kann dieses verwendet werden? Wir wollen Oracle-Cloud-Lösungen einsetzen, möchten uns aber nicht festlegen oder wissen noch nicht, welche Produkte wir benötigen und wie viel wir brauchen. Antworten auf (nicht nur) diese Fragen finden Sie in diesem Artikel.

Seit Ende 2017 bietet Oracle für die Cloud Services in den Bereichen IaaS und PaaS Cloud Services ein flexibles Vertragsmodell an, das es ermöglicht, mit einem Vertrag alle aktuellen und sogar zukünftige Oracle-PaaS- und -IaaS-Angebote zu nutzen: die sogenannten „Universal Credits“. Mit einem Universal-Credits-Vertrag erwirbt der Kunde ein Guthaben, das über die vereinbarte Laufzeit hinweg für alle zur Verfügung stehenden IaaS- und PaaS-Cloud-Services-Komponenten in allen verfügbaren Konfigurationen verwendet werden kann. Es besteht also nicht die Notwendigkeit, bereits die finale Konfiguration oder das gewünschte Rechenzentrum festzulegen, bevor ein Vertrag abgeschlossen werden kann. Vielmehr kann seitens der IT flexibel auf neue Anforderungen reagiert und neue Services können ohne aufwendigen Einkaufs- und Genehmigungsprozess aktiviert und bereitgestellt werden.

Das aktuelle Serviceangebot kann immer dem Service-Description-Dokument „Oracle PaaS and IaaS Universal Credits Service Descriptions“ entnommen werden (siehe „<https://www.oracle.com/assets/PaaS-IaaS-universal-credits-3940775.pdf>“).

Jeder Universal-Credit-Vertrag beinhaltet im Übrigen die Entwicklungsplattform „Oracle Foundation Services“ (1 Developer-Cloud-Service-Instanz pro Cloud Services Account) und 20 GB kumulativen Storage (Stand: 14. Januar 2019). Das Rechenzentrum für den Cloud Service oder die Instanz kann frei gewählt werden.

Folgende Abrechnungsmodelle sind bei Universal-Credit-Verträgen möglich:

- Pay as you go (PAYG)
- Monthly Flex

PAYG bietet komplette Flexibilität: Keine Vorauszahlung, keine Vertragslaufzeit, die monatliche Abrechnung erfolgt nach Verbrauch.

Beim Monthly-Flex-Modell wird ein Vertrag über eine Laufzeit von einem bis zu sieben Jahren abgeschlossen. Dabei wird ein Betrag festgelegt, der monatlich fällig wird, unabhängig vom tatsächlichen Verbrauch. Die Nutzung, die über den festgelegten monatlichen Betrag hinausgeht, wird „Overage“ genannt und gemäß dem im Vertrag vereinbarten Preis zusätzlich für den entsprechenden Monat in Rechnung gestellt. Das Modell kann ab

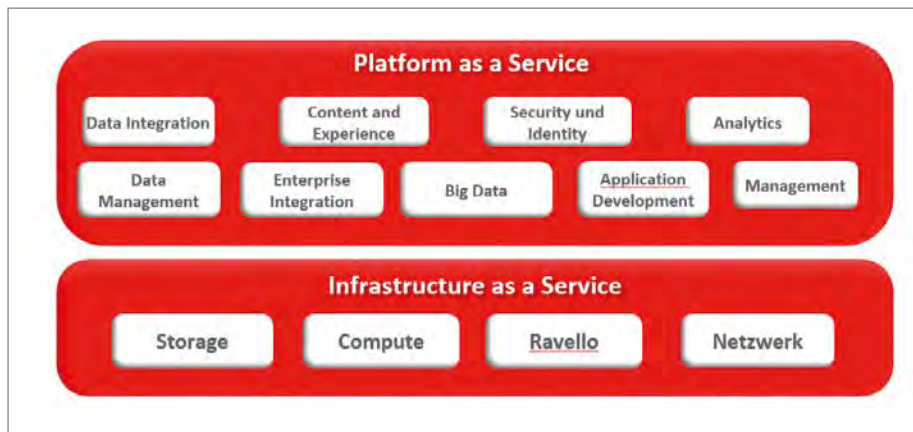


Abbildung 1: Oracle-IaaS- und -PaaS-Dienste (Stand Januar 2019)

einem Betrag von mindestens 1000 US Dollar pro Monat gewählt werden. Preislich handelt es sich aufgrund der verbindlichen Zahlungsvereinbarung um das deutlich attraktivere Modell. Die Einsparung beträgt mindestens 33 Prozent.

Auch bei über Partner erworbenen Credits erfolgt die Abrechnung für die Nutzung, die über den monatlich vereinbarten Betrag hinausgeht (Overage), direkt durch Oracle.

Wie werden die Universal Credits nun verbraucht? Jeder Universal-Credit-Vertrag enthält sogenannte „Rate Cards“, also Preislisten, in denen die Preise für jeden Service festgeschrieben sind.

Die Abrechnungskenngröße für jeden Service, zum Beispiel OCPU pro Stunde (Datenbank) oder Gigabyte-Storage-Ka-

pazität pro Monat (Storage), ist ebenfalls direkt in der Rate Card als Metrik für den jeweiligen Service hinterlegt. Eine Beschreibung für die einzelnen Metriken ist in der Service Description für Oracle PAAS und IAAS Universal Credits hinterlegt.

Die Abrechnung beginnt, sobald die Account-Login-Informationen an den Kunden verschickt wurden (Activation Email), und erfolgt ab dann immer monatlich für alle Services, die auf Basis des Universal-Credit-Vertrages konsumiert werden.

Unter der Adresse „<https://cloud.oracle.com/cost-estimator>“ gibt es einen Kostenrechner, mit dem für verschiedene Optionen auf Basis der Standardpreisliste Kostenabschätzungen ermittelt werden können.

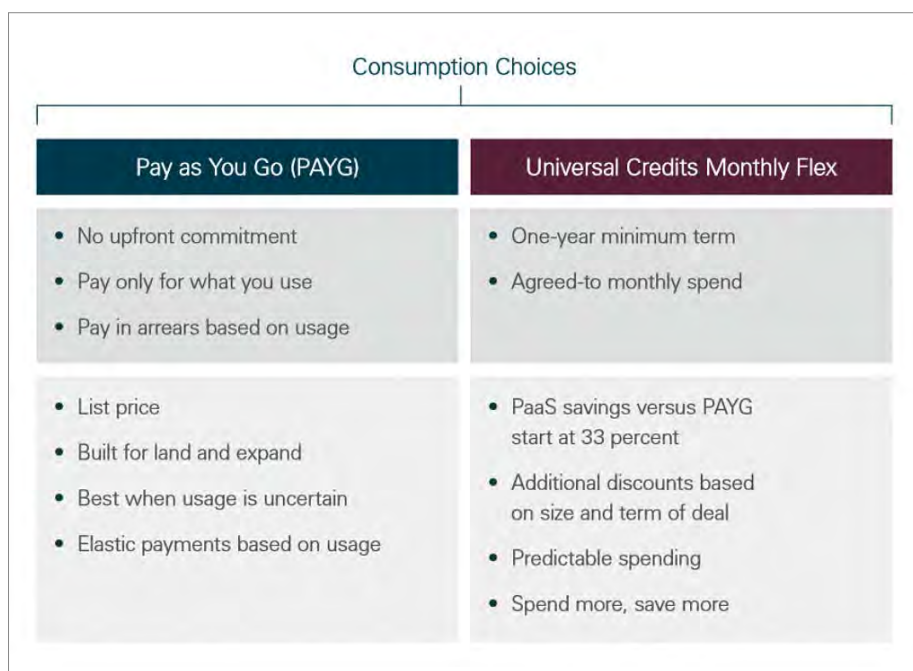


Abbildung 2: Überblick über die Universal-Credit-Abrechnungsmodelle

BYOL

Mit dem Universal-Credit-Modell können Kunden bestehende Datenbank-, Middle-ware- und Analytics-Lizenzen auch in der Cloud nutzen, wenn diese nicht mehr für andere Umgebungen benötigt werden. Voraussetzung ist, dass diese Lizenzen unter Wartung stehen. Das Programm nennt sich „BYOL“ – Bring Your Own License – und ermöglicht sowohl Investitionsschutz als auch Flexibilität für hybride Betriebsmodelle, wo Cloud und On-Premises-Welten koexistieren.

Es können nicht nur Full-Use-Lizenzen für BYOL genutzt werden, sondern auch Lizenzen mit Limited Use oder Application Specific Full Use and Proprietary Hosting (vorbehaltlich einer entsprechenden Ergänzung des ISV-Vertrages). Die spezifischen Lizenzbedingungen müssen dabei weiter eingehalten werden.

Im Universal-Credit-Modell können bestehende Lizenzen damit nicht nur auf IaaS Services, sondern auch für bereits vorkonfigurierte und automatisierte PaaS Services angewandt und genutzt werden.

Praktisch funktioniert das so, dass bei der Aktivierung die BYOL-Version des Service ausgewählt und damit der günstigere BYOL-Preis (IaaS) verrechnet wird. Eine Zuordnung, welche Lizenz verwendet wird (z.B. CSI-Nummer), ist nicht notwendig, muss jedoch im Falle eines Lizenz-Audits nachgewiesen werden. Eine Mischung von BYOL und einem vollen PaaS-Service, der auch Lizenz und Support einschließt, ist innerhalb derselben Instanz nicht möglich. Daher müssen bei einem BYOL-Service ausreichend Lizenzen für den Maximalverbrauch der Instanz (also die maximal genutzte Anzahl z.B. an OCPU) zur Verfügung stehen.

Welche Produkte beziehungsweise welche Lizenzen mit welchem Cloud Service korrespondieren und wie die Metriken umzurechnen sind, ist wiederum in der Service Description für die Universal Credit PaaS und IaaS Services einsehbar. Die Umrechnung ist auch im Oracle Core Factor Table hinterlegt (siehe „<http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>“). In der Regel korrespondiert 1 Prozessor-Lizenz mit 2 OCPU.

Eine Besonderheit: Werden Datenbank-Enterprise-Edition-Lizenzen für BYOL PaaS Services genutzt, sind Diagnostic Pack, Tuning Pack, Data Masking und Subsetting Pack sowie Real Application Testing inkludiert, ohne dass dafür separate Lizenzen benötigt werden.

Informationen zur Oracle-Cloud-Infrastruktur und Database Options (siehe Abbildung 3) finden sich unter „<https://cloud.oracle.com/database/faq>“.

Zu beachten ist in diesem Zusammenhang, dass bei einer Bestätigung im Rahmen eines Unlimited License Agreement (ULA) Cloud-Umgebungen nicht einbezogen werden können. Das bedeutet, dass keine Lizenzen gutgeschrieben werden.

Wird eine Umgebung von On-Premises in die Cloud migriert, erlaubt Oracle eine Übergangsfrist von 100 Tagen für diese Migration. Das heißt, dass für die Phase der Migration und der sich daraus ergebenden parallelen Nutzung der beiden Umgebungen, On-Premises und Cloud, für maximal 100 Tage keine zusätzlichen Lizenzen für die Nutzung im BYOL PaaS Cloud Service benötigt werden.

Wie man sieht, stellt dieses Modell maximale Flexibilität bei optimierter Kostengestaltung dar, sowohl für die Nutzung in der Oracle Cloud als auch in hybriden Umgebungen.

Die beschriebenen Dienste und Dienstleistungen spiegeln den Stand zum Datum der Verfassung dieses Artikels wider.

Weitere Informationen:

- Überblick Universal Credit Model und Bring Your Own License: <https://www.oracle.com/cloud/bring-your-own-license/>
- Universal Credit Pricing FAQ: <https://www.oracle.com/cloud/bring-your-own-license/faq/universal-credit-pricing.html>
- BYOL to PaaS FAQ: <https://www.oracle.com/cloud/bring-your-own-license/faq/>
- Cloud-Kostenrechner: <https://cloud.oracle.com/cost-estimator>
- Cloud-Service-Verträge: <https://www.oracle.com/corporate/contracts/cloud-services/>
- Oracle Core Factor Table: <http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>



Gertrud Abram-Retschitzegger
gertrud.abram@oracle.com

Database Edition	Database Options
Standard Edition	Includes the Oracle Database Standard Edition Package.
Enterprise Edition	Includes the Oracle Database Enterprise Edition Package, Data Masking and Subsetting Pack, Diagnostics and Tuning Packs, and Real Application Testing.
Enterprise Edition High Performance	Extends the Enterprise package with the following options: Multitenant, Partitioning, Advanced Compression, Advanced Security, Label Security, Database Vault, OLAP, Advanced Analytics, Spatial & Graph, Database Lifecycle Management Pack, and Cloud Management Pack for Oracle Database.
Enterprise Edition Extreme Performance	Extends the High Performance package with the following options: Real Application Clusters (RAC), In-Memory Database, and Active Data Guard.

Abbildung 3: Datenbank Cloud Services, enthaltene Datenbankoptionen und Management Packs



Oracle APEX- und PL/SQL-Entwickler (m/w/d)

TRIOLOGY GmbH

Agile Entwicklungsmethoden, exzellente Code-Qualität und aktuelle Technologien – dafür steht TRIOLOGY. Gestaltungskraft, Expertise und Leidenschaft zeichnen unsere 100 Mitarbeiter aus. Sie auch? Dann wachsen Sie mit uns und den Herausforderungen, nutzen Sie Ihren Freiraum und werden Sie Teil unseres engagierten Expertenteams! We need your talent.

Ihre Herausforderung

- ▶ Selbstständiges Betreuen und Beraten von Projekten und Kunden
- ▶ Analysieren, Konzipieren, Entwickeln und Testen von modernen Webanwendungen und Datenbanklösungen mit Oracle Application Express (APEX) und PL/SQL in agilen und klassischen Projekten
- ▶ Modellieren und integrieren von Datenbanken und Webanwendungen in Enterprise-Architekturen
- ▶ Mitwirken bei der Gestaltung und Weiterentwicklung unseres Kompetenzbereichs

Unser Angebot

Bei TRIOLOGY erwartet Sie eine herausfordernde und vielseitige Tätigkeit in einem engagiertem Team mit attraktiven Gehaltskonditionen, flachen Hierarchien und einem Miteinander mit „Wir“-Gefühl. Regelmäßige Weiterbildungsmöglichkeiten und Zusatzangebote wie z.B. Physiotherapie, Englischunterricht sowie Events schaffen das ideale Umfeld für Ihren persönlichen Erfolg.

Das bringen Sie mit

- ▶ Studium oder Ausbildung mit IT-Bezug sowie Projekterfahrung
- ▶ Erfahrung in der Entwicklung komplexer Webanwendungen mit Oracle Application Express (APEX) und PL/SQL
- ▶ Sehr gute Kenntnisse in Internettechnologien, insbesondere HTML, CSS und JavaScript
- ▶ Gute Kenntnisse im Bereich SQL und Datenmodellierung
- ▶ Lösungs- und kundenorientierte Arbeitsweise sowie sicheres Auftreten im Projekt
- ▶ Sehr gute Deutschkenntnisse sind ein Muss
- ▶ Reisebereitschaft innerhalb Deutschlands

Senden Sie uns Ihre Bewerbung gern über das Online-Bewerbungsformular oder jobs@triology.de. Für Fragen steht Ihnen unser Team Human Resources auch gern telefonisch unter **+49 531 23528-52** zur Verfügung. Besuchen Sie uns online unter triology.de oder bei



Oracle-Datenbanken in Docker-Containern

Stefan Oehrli, Trivadis AG

Oracle unterstützt seit Längerem die Nutzung von Docker für die Installation seiner Produkte, dazu gehören unter anderem die aktuellen Versionen der Oracle-Datenbank. In der Theorie wird mit einem einfachen „docker run“ aus einem Docker-Image ein entsprechender Container instanziiert. Doch wieso ist der Datenbank-Container nicht in wenigen Sekunden bereit? Wo kommt mein Oracle-Datenbank-Image überhaupt her und was geschieht, wenn der Container wieder gestoppt wird? Der Funktionsumfang sowie die Größe des Oracle-Datenbank-Containers setzen voraus, dass man sich im Vorfeld einige Gedanken zur Nutzung und zum Betrieb macht. Dazu gehören Themen wie Datenpersistenz, Lizenzierung sowie weitere betriebliche Aspekte. Dieser Artikel erklärt, wie Oracle-Datenbanken in einem Docker-Image installiert, konfiguriert und anschließend als Container betrieben werden können.

Für viele Oracle Produkte wie beispielsweise Datenbanken, Fusion Middleware, Java usw. gibt es in der Oracle Container Registry [1] fertige Docker-Container, die zumindest für Entwicklungsarbeiten direkt genutzt werden können. Daneben findet man auf Docker Hub [2] ein offizielles Oracle-Image für die Oracle Enterprise Edition 12c R2. Die Verwendung solcher vordefinierter Docker-Images ist ein großer Zeitgewinn. Trotzdem muss immer eine gewisse Sorgfaltspflicht eingehalten werden, bevor man einem vordefinierten Docker-Image vertraut. Es stellen sich Fragen rund um das Vertrauen gegenüber

dem Docker-Image-Provider, dem Inhalt des Docker-Image sowie der Aktualität. Werden Docker-Images nicht regelmäßig aktualisiert, steigt das Risiko, dass die Images Sicherheitslücken enthalten. Häufig fehlen Sicherheitsupdates sowie die Unterstützung für neue Docker-Features.

Benötigt man spezielle Oracle-Versionen oder soll der Container an die eigenen Bedürfnisse angepasst werden, lohnt sich in jedem Fall der Mehraufwand, die Docker-Images selbst zu erstellen. Man findet hierzu im offiziellen Docker GitHub [3] Repository von Oracle diesbezügliche Dockerfiles sowie Build-

und Konfigurationsskripte. Alternativ stehen im GitHub Repository des Autors [4] weitere Dockerfiles für Oracle Database Enterprise Edition 11g bis 19c sowie Oracle Unified Directory zur Verfügung. Mit einer entsprechenden Build-Umgebung hält sich der Mehraufwand für das manuelle Erstellen der Docker-Images in Grenzen. Zudem ist man so sicher, dass die Datenbank-Version, Patch Release sowie Konfiguration genau den Bedürfnissen entspricht, die man für den konkreten Anwendungsfall nutzen möchte. Das im Folgenden beschriebene Vorgehen für das Erstellen der Docker-Images basiert

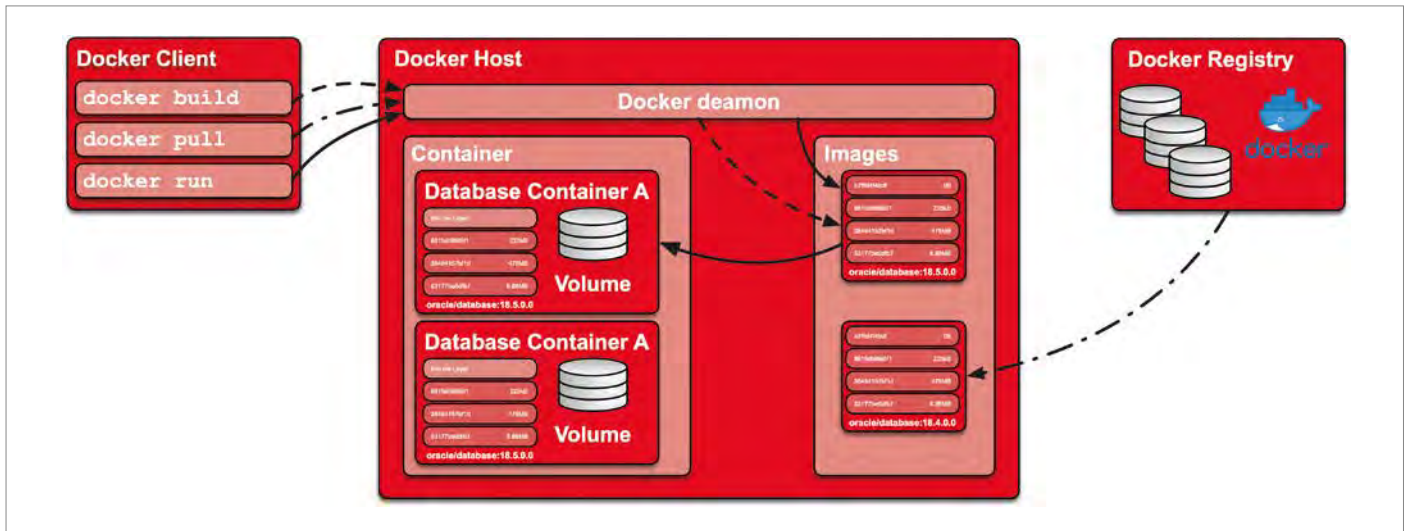


Abbildung 1: Docker-Core-Komponenten

auf den Dockerfiles aus dem GitHub Repository des Autors [4].

Support und Lizenzierung

Obwohl Docker-Images für Oracle-Datenbanken verfügbar sind, bedeutet dies nicht, dass diese zwangsläufig auch eine unterstützte Architektur ist. Etwas Klarheit schafft hier die My Oracle Support Note 2216342.1 [5]. Oracle gibt darin an, dass jeweils die neueste Version der Oracle-Datenbank für den Betrieb in Docker-Containern unterstützt und zertifiziert ist. Wobei Oracle voraussetzt, dass die Docker Engine auf einem Oracle Linux als Host läuft und die Single-Instanz-Datenbanken in Docker-Containern mit Oracle Linux 7 und UEK4 Kernel (oder neuer) oder Red Hat Enterprise Linux 7 laufen. Für Test- und Entwicklungsumgebungen werden zudem Real Application Clusters (RAC) Konfigurationen unterstützt. Somit steht einem nichts im Weg, um produktive Oracle-Datenbanken in Docker-Containern zu betreiben.

Moment, da war doch noch etwas! Ja genau: die Lizenzierung. Sämtliche Oracle-Datenbanken und Optionen, die in Docker-Containern genutzt werden, sind entsprechend zu lizenzieren. Docker erlaubt zwar die Limitierung von Ressourcen, wie Memory und CPU, diese Limitierung kann aber im Kontext von Oracle nicht genutzt werden, um die Oracle-Lizenzierung einzuschränken. Somit bleibt die Lizenzierung von Oracle-Produkten im Container-Umfeld eine

ähnliche Herausforderung wie bei der Virtualisierung. Die Anzahl der CPU Cores der Docker Hosts bestimmen die benötigten Lizenzen. Dies ist insbesondere bei gemischten Workloads problematisch. Ein ULA (Unlimited License Agreement) oder eine sorgfältig geplante Docker-Container-Infrastruktur sind mögliche Lösungsansätze. Als weitere Alternative bietet sich insbesondere der Einsatz von Oracle Database Express Edition (XE) 18c [6] an. Mit der frei verfügbaren Version können Datenbanken mit 2 CPU-Threads, 2 GB RAM und bis zu 12 GB Benutzerdaten erstellt werden. Zudem stehen in dieser Version verschiedene Enterprise Features [7] zur Verfügung.

Docker-Images, -Container und -Volumes

Die Abbildung 1 zeigt schematisch die Core-Komponenten von Docker. Vordefinierte Docker-Images werden mit `docker pull` lokal auf den Docker Host geladen. Alternativ können Images mit `docker build` sowie entsprechenden Dockerfiles erstellt werden. Die mitunter wichtigste Eigenschaft von Docker-Images ist, dass sie unveränderlich sind. Dies bedeutet, dass ein einmal erstelltes Image nicht verändert werden kann, sondern komplett neu erstellt werden muss. Alternativ lässt sich ein neues Image basierend auf einem bestehenden Image erstellen. Ein Docker-Container dagegen entspricht einem Docker-Image, das mit `docker run` gestartet wurde und auf der Docker En-

gine ausgeführt wird. Ein wesentlicher Aspekt des Docker-Containers ist die Datenpersistenz respektive das Fehlen derselbigen. Veränderte Daten in einem Docker-Container sind nur so lange verfügbar, wie dieser existiert. Wird ein Container gestoppt und anschließend basierend auf einer aktuellen Version eines Image neu gestartet, gehen die zuvor erstellten Daten verloren.

Durch die Verwendung von Docker-Volumes kann die Datenpersistenz trotzdem sichergestellt werden. Veränderbare Daten werden dabei nicht mehr innerhalb des Containers, sondern in einem oder mehreren Volumes abgespeichert. Somit bleiben die Daten über die Lebensdauer eines Containers bestehen.

Voraussetzungen

Am einfachsten kann man ein Oracle-Datenbank-Image basierend auf einem Oracle Enterprise Linux Image erstellen. Zum einen ist das Image auf Docker Hub verfügbar und zum anderen lassen sich die fehlenden Pakete einfach mit `yum` installieren, z.B. die Oracle-Preinstall-Pakete `oracle-database-preinstall-18c-1.0`. Mit dem Docker-Kommando in Listing 1 kann das Basis-Image direkt von Docker Hub heruntergeladen werden. Wird dies nicht vorgängig gemacht, erfolgt der Pull

```
docker pull oraclelinux:7-slim
```

Listing 1

```
cd OracleDatabase/18.5.0.0
docker build -t oracle/database:18.5.0.0 .
```

Listing 2

Variable	Verwendung
00_setup_oradba_init.sh	Skript für die Installation der aktuellsten OraDBA-Init-Skripte.
01_setup_os_db.sh	OS-Setup-Konfiguration. Benutzer und Gruppen erstellen. YUM-Software-Pakete installieren.
10_setup_db_18.5.sh	Installation der Oracle Binaries sowie, falls vorhanden, das aktuellste Release, Updates und Oracle JVM Update
20_setup_basenv.sh	Installation von Trivadis BasEnvTM
5n_xxxx_database.sh	Verschiedene Skripte für die Konfiguration zur Laufzeit des Containers. Diese werden im Folgenden ausführlicher beschrieben.

Tabelle 1

Request von Docker Hub implizit beim Build des Oracle-Datenbank-Image. Im Rahmen dieses Artikels werden wir uns auf folgende Versionen beschränken:

- Betriebssystem Oracle Enterprise Linux 7.5 (slim)
- Oracle Database Enterprise Edition 18c (18.4.0.0)
- Aktuelle Updates vom Januar 2019 (Release Update und OJVM Update)
- Docker-Build-Skripte [4]
- Optional die Umgebungsskripte von Trivadis BasEnvTM

Weitere Versionen werden durch Oracle unterstützt. Im GitHub Repository des Autors [4] finden sich Dockerfiles für weitere Oracle-Datenbank-Versionen sowie für Oracle Unified Directory. Als Entwicklungsumgebung wird die aktuellste Version der Docker Community Edition mit der Docker Engine 18.09.2 auf einem Laptop mit MacOS 10.14 verwendet. Es ist ohne weiteres möglich, das vorgestellte Setup in aktuellen Docker-Umgebungen auf anderen Betriebssystemen nachzubauen. Neben der lauffähigen Docker-Umgebung wird vorausgesetzt, dass sowohl die Docker-Build-Skripte als auch die Software lokal vorhanden sind. Die einfachste Methode dafür ist, ein Git Clone vom GitHub Repository des Autors [4] zu erstellen. Die Links für die Downloads der oben aufgeführten Software sind, wie man es vom offiziellen Oracle Docker GitHub [3] Repository gewohnt ist, als entsprechende *.download-Textdateien aufgeführt.

Datenbank-Image

Nachdem das Basis-Image vorhanden ist, muss die Software, d.h. die Datenbank Binaries „LINUX.X64_180000_db_home.zip“ sowie das aktuelle Release Update „p28822489_180000_Linux-x86-64.zip“ und das OJVM Update „p28790647_180000_Linux-x86-64.zip“, in den Build-Order (\$HOME/docker/OracleDatabase/18.5.0.0) kopiert werden. Anschließend wird das Oracle-Datenbank-Docker-Image mit dem Kommando in Listing 2 erstellt.

Das Dockerfile nutzt für das Erstellen des Image das Multi-Stage-Build-Feature [8], das seit Docker 17.05 verfügbar ist. Bei diesem Verfahren werden mehrere FROM-Anweisungen in einem Dockerfile verwendet, wobei jede von ihnen eine neue Phase des Builds aufruft. Diese Methode erlaubt es, den Build-Prozess und das Ziel-Image zu optimieren. Insbesondere wird dabei die Image-Größe klein gehalten. Zusammenfassend werden mit dem „docker build“-Kommando folgende Schritte aus dem Dockerfile durchgeführt:

1. Setzen der verschiedenen Umgebungsvariablen abhängig von etwaigen Build-Argumenten für ORACLE_ROOT, ORACLE_DATA oder ORACLE_BASE. Mit diesen Parametern lässt sich bei Bedarf die Verzeichnisstruktur der Oracle-Installation, insbesondere das Verzeichnis für die Datenbank, sowie Oracle Base anpassen. In der Regel ist dies jedoch nicht nötig.

2. Download und Installation der aktuellsten Version der Skripte direkt von GitHub mit 00_setup_oradba_init.sh.
3. Konfiguration des Basis-Image mit dem Skript 01_setup_os_db.sh. Erstellen des Benutzers und Gruppen für die Oracle-Installation sowie Anlegen der verschiedenen Verzeichnisse. Zusätzlich werden noch fehlende Pakete aus dem Oracle Public YUM Repository nachinstalliert (tar, gzip und oracle-database-preinstall-18c).
4. Neuer Build Stage für die Oracle-Installation. Basierend auf dem Basis-Image sowie auf den oben durchgeführten Schritten wird ein Stage Builder respektive ein Build-Image erstellt.
5. Kopieren der Oracle Binaries und Patch Files in das Build-Image.
6. Installation der Oracle Software mit dem Skript 10_setup_db_18.5.sh. Das Skript installiert sowohl die Oracle Binaries als auch eventuelle Patches.
7. Optionale Installation von Trivadis BasEnvTM, sofern das entsprechende Software-Paket vorhanden ist.
8. Neuer Build Stage für das Ziel-Docker-Image, basierend auf dem Basis-Image.
9. Kopieren der installierten Software aus dem Build-Image. Dies bedeutet, dass die in Schritt 6. und 7. installierte Software nun direkt in das Ziel-Image übernommen wird. Allfällige temporäre Dateien wie ZIP, Tar Archive, Cache Files etc. bleiben im Build-Image, das am Ende verworfen wird.
10. Abschluss der Oracle-Installation und Ausführen der root.sh-Skripte.
11. Festlegen der Ports für den späteren Export.
12. Definition des Volume-Pfades für die persistenten Daten. Die Datenbank sowie Log- und Konfigurationsdateien sind in einem separaten Verzeichnis ORACLE_DATA respektive standardmäßig in /u01 abgelegt.
13. Abschließend werden das Healthcheck-Skript sowie das Datenbank Start-Skript festgelegt.

Um die Installation und Konfiguration im Dockerfile einfacher zu halten, werden verschiedene Skripte verwendet. Die aktuellste Version dieser Skripte wird jeweils direkt aus dem GitHub Repository [9] geladen. Die folgende Tabelle 1 gibt eine Übersicht über die verwendeten Skripte.

Datenbank-Container

Mit dem so erstellten Docker-Image lässt sich nun ein Datenbank-Container erstellen und starten. Das folgende Kommando in *Listing 3* startet im Hintergrund einen Container mit dem Namen tdb185s und erstellt im Volume-Pfad /data/docker/volumes/tdb185s die Oracle DB mit der Oracle SID TDB185S.

Das Erstellen der Datenbank dauert einen Moment. Bis die Datenbank-Konfiguration abgeschlossen ist, wird der Status mit dem folgenden Kommando (*siehe Listing 4*) abgefragt.

Der Datenbank-Container ist bereit, sobald in der Docker-Container-Log-Datei, wie in der *Abbildung 2* ersichtlich, eine entsprechende Meldung ausgegeben wird.

Beim Erstellen des Containers wurde explizit das Port-Mapping angegeben. So ist der Container von „außen“, also vom Host-System erreichbar. Auf diesen Container kann nun via Kommandozeile mit `docker exec` interaktiv oder mit SQL Developer auf den exportierten Port zugegriffen werden. In der *Abbildung 3* sieht man ein Beispiel dafür, wie mit `docker exec` eine Bash-Shell im Container tdb185s gestartet wird. Auf diese Weise lässt sich die Datenbank wie gewohnt mit der Kommandozeile verwalten.

Wie oben ersichtlich ist, läuft im Container die Datenbank-Instanz TDB185S. Der Grund dafür liegt beim Skript, das beim Start des Containers eine vorhandene Datenbank startet beziehungsweise eine neue angelegt, falls keine vorhanden ist. Dabei werden verschiedene Standardwerte verwendet, die sich durch Setzen von Umgebungsvariablen steuern lassen. Dreh- und Angelpunkt ist dabei das Skript `50_run_database.sh`. Falls keine Datenbank vorhanden ist, erstellt das Skript `52_create_database.sh` mit dem Oracle DBCA eine neue Datenbank. Bei Bedarf werden mit dem Skript `55_config_database.sh` zusätzliche Konfigurationsskripte ausgeführt, um die Datenbank zu konfigurieren. Die *Abbildung 4* zeigt schematisch den Ablauf beim Start respektive beim Erstellen des Datenbank-Containers.

Neben `50_run_database.sh` und `52_create_database.sh` werden folgende Hilfe-Skripte im Docker-Container genutzt:

- **50_run_database.sh** prüft das Volumen und startet die Datenbank mit

```
docker run --detach \  
--volume /data/docker/volumes/tdb185s:/u01 \  
-p 1521:1521 -p 5500:5500 \  
-e ORACLE_SID=TDB185S \  
--hostname tdb185s --name tdb185s \  
oracle/database:18.5.0.0
```

Listing 3

`50_start_database.sh`. Definiert zusätzlich Signal Handler für SIGINT, SIGTERM und SIGKILL. Ist keine Datenbank vorhanden, wird das Skript `52_create_database.sh` aufgerufen.

- **50_start_database.sh** startet den Listener und die Datenbank-Instanz.
- **52_create_database.sh** erstellt bei Bedarf eine Datenbank mit dem Oracle DBCA. Basis-Parameter lassen sich mit Umgebungsvariablen anpassen. Das Skript generiert explizit ein Passwort für den SYS-Benutzer und legt dieses in der Datei `/${ORACLE_BASE}/admin/${ORACLE_SID}/etc/${ORACLE_SID}_password.txt` ab.
- **55_config_database.sh** prüft, ob entsprechende Konfigurationsdateien (`*.sh` oder `*.sql`) im Verzeichnis `/${INSTANCE_INIT}/setup` beziehungsweise `/${IN-`

```
docker logs -f tdb185s
```

Listing 4

`STANCE_INIT}/startup` vorhanden sind. Falls die Dateien vorhanden sind, werden sie sequenziell ausgeführt. Wobei die Skripte aus dem Verzeichnis `setup` nur beim Erstellen der Datenbank aufgerufen werden und jene aus dem Verzeichnis `startup` bei jedem Start der Datenbank.

- **55_check_database.sh** wird für den Healthcheck des Docker-Containers genutzt und prüft den Status der Datenbank-Instanz. Das Skript ist verantwortlich für die Angabe eines Status „healthy/unhealthy/starting“ beim Aufruf von `docker ps`.

```
SQL> Disconnected from Oracle Database 18c Enterprise Edition Release  
18.0.0.0.0 - Production  
Version 18.5.0.0.0  
/u00/app/oracle  
/u00/app/oracle  
The Oracle base remains unchanged with value /u00/app/oracle  
-----  
- DATABASE TDB185S IS READY TO USE!  
-----  
Tail output of alert log from TDB185S:  
-----  
The following output is now a tail of the alert.log:  
2019-02-25T20:25:54.227589+00:00  
Starting background process CJQ0  
Completed: ALTER DATABASE OPEN
```

Abbildung 2: Docker-Container-Log

```
soe@gaia:~/ [ic12201] docker exec -it tdb185s bash --login  
TYPE (Cluster|DG) : SID/PROCESS STATUS HOME [2019-02-25 21:00:14]  
-----  
Dummy rdbms_ee : rdbms18500 n/a /u00/app/oracle/product/18.5.0.0  
DB-instance (N|N) : TDB185S open /u00/app/oracle/product/18.5.0.0  
Listener : LISTENER up /u00/app/oracle/product/18.5.0.0  
oracle@tdb185s:/u00/app/oracle/ [TDB185S] █
```

Abbildung 3: Interaktiver Zugriff auf tdb185s via `docker exec`

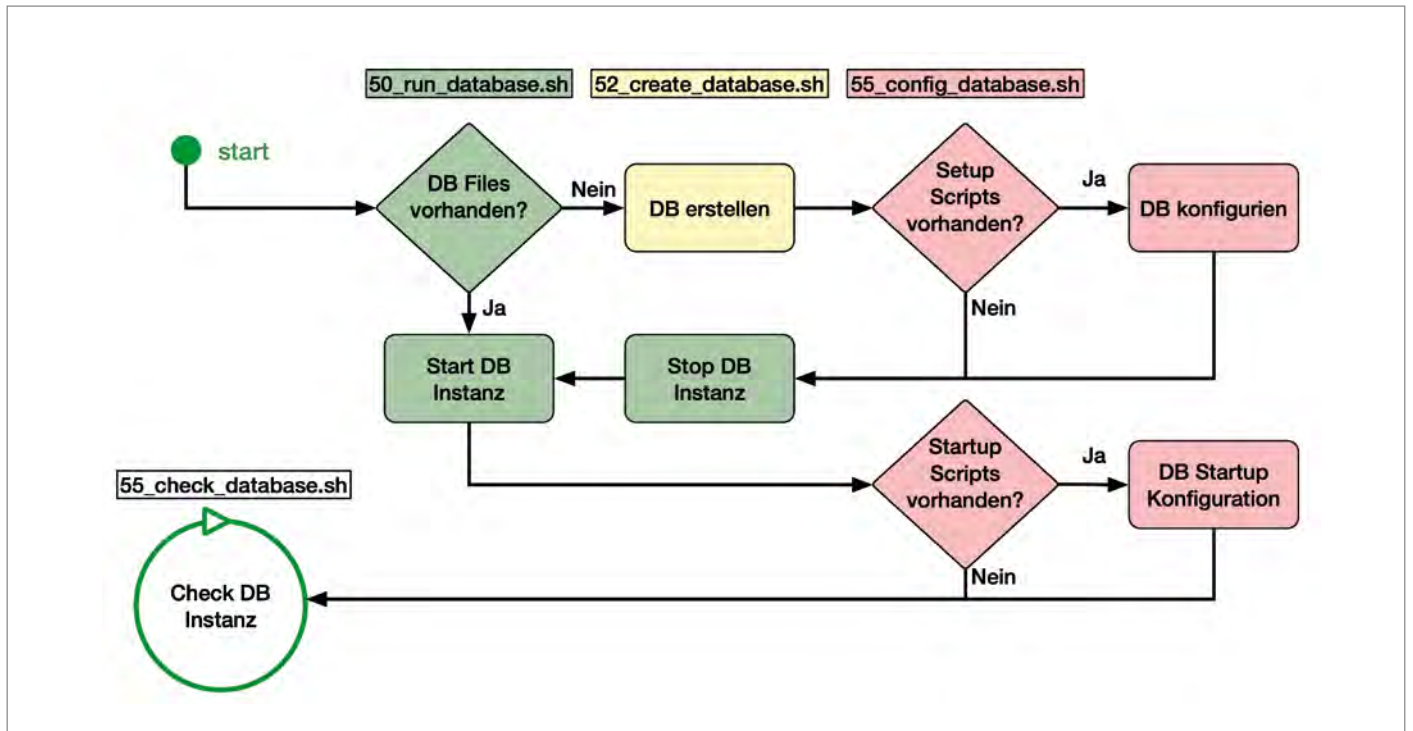


Abbildung 4: Ablauf Start DB-Instanz

Variable	Verwendung
ORACLE_SID	Oracle SID respektive Datenbank-Name. Standardwert ist TDB185S
CONTAINER	Flag für das Erstellen einer Oracle-Container-Datenbank Standardwert ist FALSE
ORACLE_PDB	Oracle-PDB-Name. Standardwert PDB1
ORACLE_CHARACTERSET	Oracle-Zeichensatz. Standardwert AL32UTF8.
ORACLE_PWD	Password für den SYS-Benutzer. Wird normalerweise automatisch generiert und im admin-Verzeichnis abgelegt. Durch Setzen von ORACLE_PWD kann explizit ein Passwort angegeben werden. Dies bleibt aber im Container als Umgebungsvariable gesetzt und ist somit auch immer sichtbar.
INSTANCE_INIT	Verzeichnis für die Instanz-Konfigurationsdateien für die Konfiguration bei Setup sowie Startup der Datenbank. Standardwert \${ORACLE_BASE}/admin/\${ORACLE_SID}/scripts
ORADBA_RSP	Standard-Verzeichnis für die DBCA-Response-Files /opt/oradba/rsp
ORADBA_TEMPLATE_PREFIX	Präfix für das DBCA Template Dateiname. Standardmäßig ist diese Variable nicht gesetzt.
ORADBA_RSP_FILE	DBCA-Response-Datei. Standardwert ist DBCA18.0.0.rsp.tmpl
ORADBA_DBC_FILE	DBCA-Template Dateiname. Standardwert \${ORADBA_TEMPLATE_PREFIX}DBCA18.0.0.dbc.tmpl

Tabelle 2

Die Parametrisierung der Skripte erfolgt grundsätzlich via Umgebungsvariablen bei der Instanziierung des Docker-Containers mit dem Parameter -e. Mit wenigen Ausnahmen ist eine nachträgliche Anpassung nicht sinnvoll. Ein Beispiel für diese Parametrisierung ist im folgenden Code-Beispiel (siehe Listing 5) ersichtlich. Dieser Aufruf von docker run erstellt in

diesem Fall eine Oracle-Multi-Tenant-Datenbank mit der Oracle SID TDB185C und dem Instanz-Konfigurationsverzeichnis /u01/config.

Neben der Oracle SID können noch weitere Parameter zur Konfiguration des Datenbank-Containers respektive der Datenbank-Instanz genutzt werden. Die Tabelle 2 gibt eine kurze Übersicht.

Mit der Kombination der Umgebungsvariablen, den Konfigurationsskripten und den DBCA-Templates lassen sich Datenbank-Instanzen für unterschiedliche Use Cases anlegen. Alternativ kann man vorher im entsprechenden Volume-Pfad manuell eine Datenbank erstellen. Die manuell erstellte Datenbank wird bei einem späteren Starten des Containers durch 50_run_database.sh automatisch gestartet. Läuft dabei etwas schief, hat man die Möglichkeit, direkt im Docker-Container die Log-Dateien zu prüfen oder mit docker logs das Datenbank-Alert-Log anzuzeigen. Mit „docker logs <CONTAINER>“ sieht man zudem immer auch den STDOUT der Hilfskripte beim Erstellen, Starten oder Stoppen einer Datenbank.

Weitere Aspekte

Die Datenbank-Container für die Single-Tenant- sowie die Multi-Tenant-Datenbank wurden zuvor jeweils direkt mit docker run erstellt. Bei dieser Methode wird das Docker-Kommando relativ schnell komplex und vor allem unübersichtlich. Es empfiehlt sich daher, den Datenbank-Container mithilfe einer Docker-Compose-Datei anzulegen. So wird sichergestellt, dass der Container immer gleich aufgebaut wird. Zudem kann der

```
docker run --detach \
--volume /data/docker/volumes/tdb185c:/u01 \
-p 2521:1521 -p 6500:5500 \
-e ORACLE_SID=TDB185C \
-e CONTAINER=TRUE \
-e INSTANCE_INIT=/u01/config \
--hostname tdb185c --name tdb185c \
oracle/database:18.5.0.0
```

Listing 5

```
version: '3'
services:
  tdb185s:
    image: oracle/database:18.5.0.0
    container_name: tdb185s
    hostname: tdb185s
    restart: unless-stopped
    network_mode: bridge
    volumes:
      - ${DOCKER_VOLUME_BASE}/tdb185s:/u01
      - ./config:/u01/config
    ports:
      - "1521"
    environment:
      INSTANCE_INIT: /u01/config
      ORACLE_SID: TDB185S
```

Listing 6

```
docker-compose up -d
```

Listing 7

```
docker-compose stop
docker-compose start
```

Listing 8

```
docker-compose down
```

Listing 9

Datenbank-Service mit weiteren Services, etwa einem Applikations-Container oder einem Oracle-Unified-Directory-Container für die Enterprise User Security, kombiniert werden. Für den Container mit der Oracle SID TDB185S sieht die Datei docker-compose.yml wie in Listing 6 aus. Der Container wird mit dem Kommando docker-compose administriert (siehe Listing 7).

Nachdem der Container mit einem up -d einmal erstellt und gestartet wurde, kann er später einfach mit stop oder

start gestoppt respektive gestartet werden (siehe Listing 8). Ein down stoppt und löscht den Container (siehe Listing 9).

Die Datenbank ist in diesem Beispiel nicht im Container, sondern in einem Volume abgelegt. Somit ist die Datenpersistenz sichergestellt. Umfangreichere Beispiele für Docker-Compose-Dateien sowie weitere Use Cases sind im GitHub Repository des Autors [4] im Verzeichnis samples aufgeführt.

Zusammenfassung

Egal ob man Oracle-Datenbanken nur einmal anschauen möchte oder einen konkreten Anwendungsfall verfolgt – mit der hier vorgestellten Methode lassen sich Oracle-Datenbanken schnell und einfach in einem Docker-Container aufsetzen und betreiben. Die Möglichkeit, Datenbanken zusätzlich mit Setup- und Startup-Skripten anzupassen, liefert die nötige Flexibilität, spezifische Bedürfnisse abzudecken. Mit einem docker run ist ein neuer Datenbank-Container schnell erstellt und man hat Zeit, sich auf das Wesentliche zu konzentrieren, ohne sich

zuerst mit der Installation und Konfiguration zu befassen. Zudem steht einer produktiven Nutzung von Oracle-Datenbanken auf Docker grundsätzlich nichts im Weg. Vorausgesetzt, man hat die nötigen Lizenzen für die Oracle Enterprise oder Standard Edition. Ist das nicht der Fall, bietet Oracle mit der Oracle Express Edition 18c eine spannende Alternative, die ebenfalls viele Enterprise Features enthält und sich zudem einfach in Docker betreiben lässt. Der Knackpunkt bei Datenbanken im Container-Umfeld ist und bleibt allerdings die Datenpersistenz. Unabhängig vom Datenbank-System wird ein entsprechendes Konzept für den Umgang mit persistenten Daten in Volumes benötigt.

Quellen

- [1] Oracle Container Registry <https://container-registry.oracle.com>
- [2] Docker Hub Oracle Enterprise Edition https://hub.docker.com/_/oracle-database-enterprise-edition
- [3] Oracle Docker GitHub <https://github.com/oracle/docker-images>
- [4] Stefan Oehrli GitHub Docker Repository <https://github.com/oehrlis/docker>
- [5] My Oracle Support Note 2216342.1 Oracle Support for Database Running on Docker <https://support.oracle.com/epmos/faces/DocumentDisplay?id=2216342.1>
- [6] Oracle Database Express Edition XE 18c <https://url.oradba.ch/oraXE18c>
- [7] Oracle Database Express Edition XE 18c Lizenz-Informationen <https://url.oradba.ch/oraXE18cLicense>
- [8] Docker Multi-stage build <https://url.oradba.ch/2pKnOM9>
- [9] Stefan Oehrli GitHub Repository OraDBA Init Scripte https://github.com/oehrlis/oradba_init



Stefan Oehrli
stefan.oehrli@trivadis.com



Security im Oracle-Cloud-Rechenzentrum der Generation 2 in Frankfurt

Michael Fischer, Oracle Deutschland B.V. & Co. KG

Oracle bietet seit mehreren Jahren Cloud Services im Bereich IaaS, PaaS und SaaS an. In der Keynote der Oracle Open World 2018 hat Larry Ellison, CTO von Oracle, die Generation 2 der Oracle-Cloud-Rechenzentren mit herausragenden Sicherheitsmerkmalen vorgestellt. Eines dieser Rechenzentrumsverbunde befindet sich in Frankfurt. Der folgende Artikel beschreibt dessen Besonderheiten.

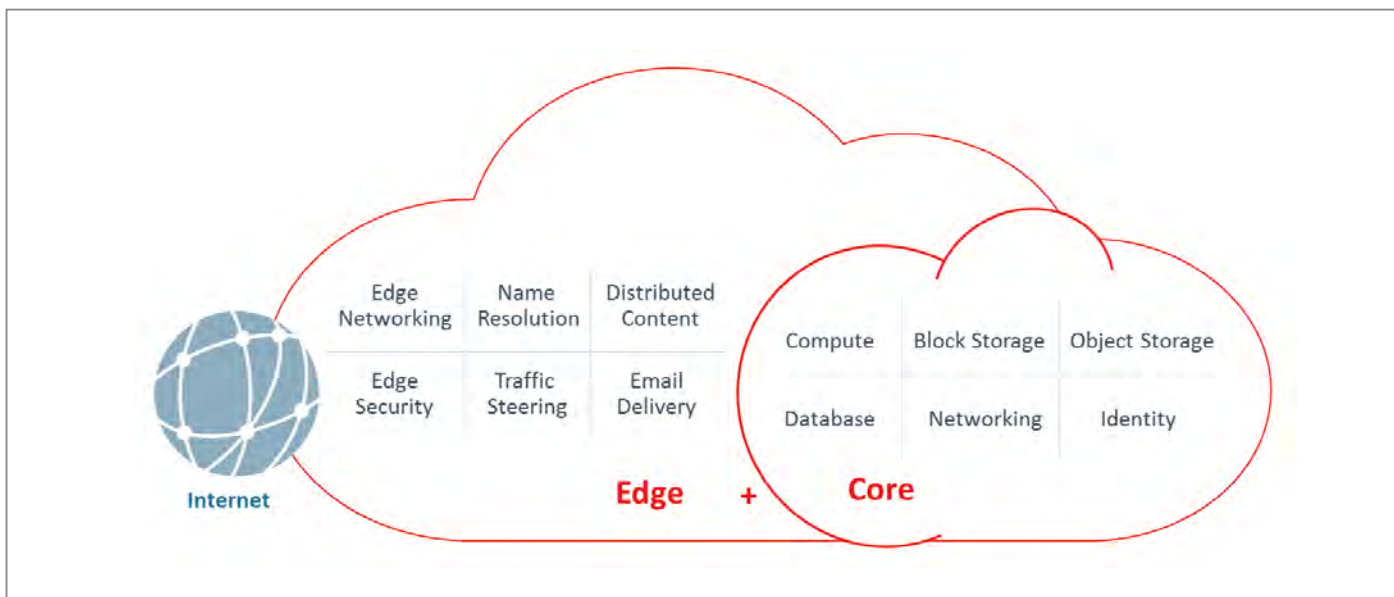


Abbildung 1: Bestandteile der Oracle Infrastructure Cloud

Oracle Cloud Infrastructure Services werden auch in den Rechenzentren in Frankfurt bereitgestellt. Dort werden drei örtlich getrennte Rechenzentren genutzt, um Ausfallsicherheit und entsprechende Disaster-Recovery-Funktionen bereitzustellen. Es handelt sich dabei um Oracle's Generation 2 Cloud (kurz Gen2 Cloud), die von Grund auf neu entwickelt wurde, um den aktuellen Anforderungen von Anwendungen gerecht zu werden. Unternehmen jeder Größe können damit eigenständig ihre geschäftskritischen, großvolumigen, leistungshungrigen Anwendungen und Datenbanken in der Cloud betreiben. Dies gilt für traditionelle und Cloud-native Workloads. Als Cloud-Infrastrukturen werden mehrere Optionen angeboten: Dedizierte, auch Bare Metal Server, virtuelle Maschinen, Container, serverlose Systeme und PaaS-Services. Die Plattformen können sowohl klassische On-Premises-Systeme einbeziehen als auch On-Premises über „Cloud at Customer“ zur Verfügung gestellt werden. Um Hybridlösungen zu ermöglichen, bietet Oracle dedizierte Verbindungen zwischen der Cloud und den Kunden-Rechenzentren. Die Nutzung von VPN ist möglich. Wird das öffentliche Internet genutzt, kommen DNS-Services von Oracle zum Tragen, um optimierten und geschützten Zugriff zu ermöglichen.

Oracle arbeitet bei der Bereitstellung der Cloud mit einem dualen Ansatz, bestehend aus Eigenentwicklungen und der Bereitstellung von Open-Source-Lösun-

gen. Die Entwicklung eigener Technologien erfolgt beispielsweise mit der Cloud-Infrastruktur selbst oder mit einigen PaaS-Services wie denjenigen für Datenbanken. Um als Anwender Cloud-native Anwendungen zu erstellen und betreiben zu können, werden in der Open-Source-Community entwickelte Bausteine wie z.B. Kubernetes, Container, Terraform oder Kafka angeboten. Oracle ist ein Platin-Mitglied der Cloud Native Computing Foundation (CNCF), um das native Cloud-Ökosystem entsprechend zu unterstützen.

Die Oracle Infrastructure Cloud besteht aus zwei logischen Hauptbestandteilen, den sogenannten Edge- und Core-Services (siehe Abbildung 1). Core-Services sind die klassischen Cloud Services wie Compute, Storage, Datenbanken etc. Edge-Services sichern unter anderem Anwendungen, bevor Requests und Daten an die Oracle Cloud, an einen anderen Cloud-Provider oder an lokale Infrastrukturen von Kunden weitergeleitet werden. Die Edge-Services sind für alle Netzwerke nutzbar und umfassen DNS inklusive Sammeln und Auswertungen von Routinginformationen, (D)DoS-Abwehr, Web-Application Firewall (WAF), Email und Content-Delivery-Netzwerke (CDN).

Die Oracle Infrastructure Cloud bietet neben den webbasierten Benutzerschnittstellen auch Schnittstellen zur automatisierten Verwaltung, der Auswertung der durchgeführten Aktionen und des Monitorings. Für das Security und Compliance Monitoring stehen sowohl Audit Services

innerhalb der Cloud zur Verfügung als auch Oracle Services wie CASB, Security Monitoring und Compliance Service, um tiefgreifende Funktionen nutzen und weitere Umgebungen einbeziehen zu können. Das in der Keynote dargestellte „Dedicated Network of Cloud Control Computers“ beschreibt die Absicherung der Cloud durch die sogenannten Edge-Services und auch die Off-Box-Netzwerk-Virtualisierung. Beide Funktionalitäten sind im Folgenden weiter ausgeführt.

Sicherheitsprinzipien der Gen2 Cloud

Sicherheit ist ein grundlegendes Konstruktionsprinzip der Oracle Cloud. Die Sicherheitsprinzipien sind charakterisiert durch folgende Merkmale:

- Abwehr gegen Cyberangriffe
- Kundenisolierung
- Verschlüsselung
- Security Controls
- Überprüfbarkeit/Audit
- Verfügbarkeit

Die folgende kurze Beschreibung beginnt mit den Edge-Services, gefolgt von den Core-Services, die auch die Darstellung von externen Security- und Compliance-Services enthält.

Organisatorische Themen wie Cloud Operations, Prozesse und Policies oder Vertragsgrundlagen werden hier nicht dargestellt. Das generelle Cloud-Modell mit der gemeinsamen Verantwortung für Sicherheit, die sogenannte „shared res-

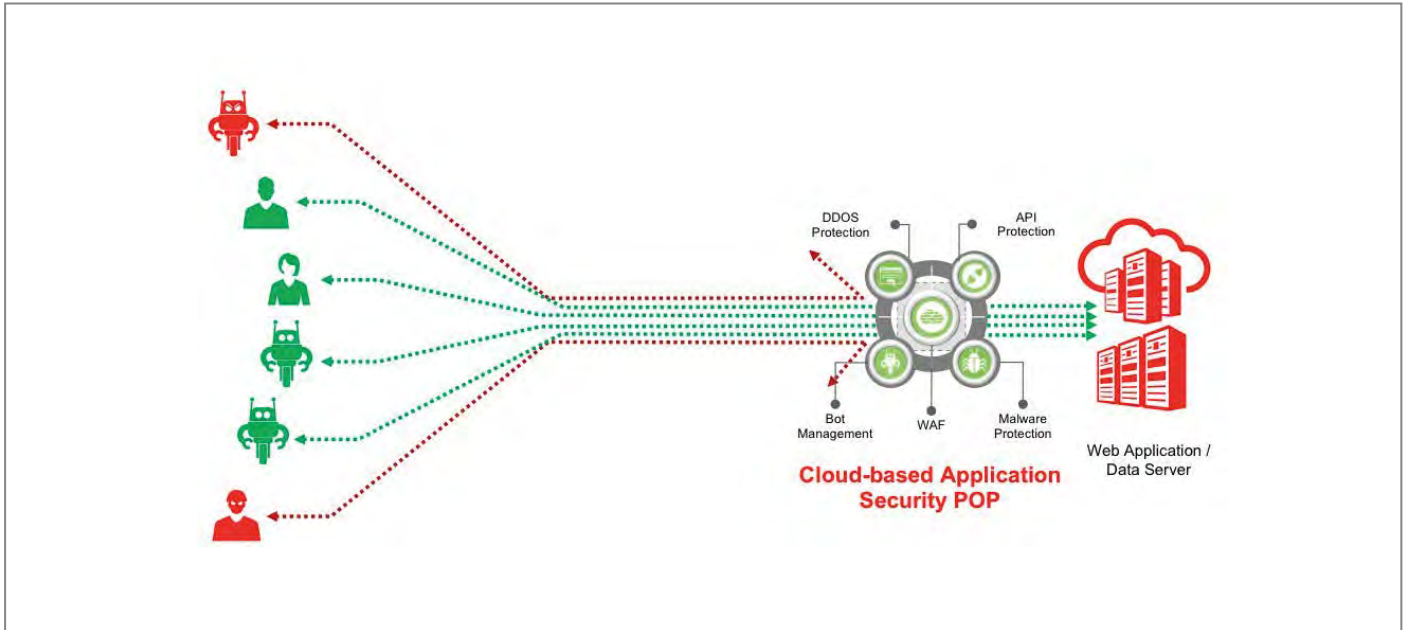


Abbildung 2: Security-relevante Funktionen der Oracle Edge-Services

possibility“, wird hier ebenfalls nicht weiter ausgeführt.

Oracle Infrastructure Cloud Edge-Services

Die Edge-Services sichern die Oracle Cloud vor einer Vielzahl von externen und internen Bedrohungen. Sicherheitsbedrohungen wie DDoS, bösartige Bots sowie Cyberangriffe werden beim Eintritt in das Netzwerk analysiert und – sobald erkannt – auch mitigiert. Edge-Services stellen darüber hinaus auch die Verfügbarkeit und optimale Erreichbarkeit der Anwendungen sicher. Edge-Services werden aus 16 global verteilten Edge-Computing-Standorten mit hoher Kapazität bereitgestellt.

Die Edge-Services von Oracle stellen Lösungen für Webanwendungssicherheit, DDoS-Schutz und DNS mit folgenden Merkmalen bereit (siehe Abbildung 2):

1. Zusätzliche Verteidigungsebenen, die Benutzer, Anwendungen, Daten und Infrastruktur schützen und die Verfügbarkeit der Netzwerkressourcen der Oracle Cloud Infrastructure bei einem Angriff sicherstellen:
 - DDoS-Angriffserkennung und -Minderung auf Layer 3 und 4, z. B. SYN-Floods, UDP-Floods, ICMP-Flood und NTP Amplifier.

- Globales Anycast-Netzwerk mit mehreren über Kontinente verteilten Rechenzentren und redundanten Tier-1-Transit Anbietern für Ausfallsicherheit und Schutz vor DDoS-Angriffen.
- Echtzeit-Internet-Health-Analyse mit einem eigenen Überwachungsnetzwerk, das Leistungsabfälle, Änderungen beim Internet-Routing, Netzwerksicherheitswarnungen und DDoS-Schutzaktivierungen von DDoS-Anbietern einbezieht und berücksichtigt.
- WAF-Funktionen (native Web Application Firewall) zum Schutz auf Layer 4-7 sowie Regelwerke gegen Botnets, Cross Site Scripting, SQL Injection und DDoS-Angriffe.
- Cybersicherheitsexperten, die die Webanwendungsfirewall (WAF), Bot-Management, Malware-Schutz und API-Sicherheitslösungen verwalten.

2. Optimierung für die Erreichbarkeit und Traffic-Performance von Workloads in Clouds und On-Premises:
 - DNS basiertes Load Balancing mit Lastverteilung, DNS-Auflösung und lokalem Caching.
 - DNS-Abfragen in weniger als 30 Millisekunden und Propagierung von DNS-Einträgen in weniger als einer Minute.

- Intelligente Weiterleitung des Benutzerverkehrs basierend auf den Informationen der Echtzeit-Internet-Health-Analyse und damit u.a. Reduktion der Latenzen.

Die aufgeführten Edge-Services werden aus zusätzlichen Edge-Rechenzentren erbracht und sind zum Teil implizit in der Oracle Infrastructure Cloud vorhanden (DDoS Layer 3 und 4 Abwehr) oder können explizit bezogen werden (z.B. DYN DNS, WAF).

Oracle Infrastructure Cloud Core-Services

Die Oracle Cloud Infrastructure besteht aus verschiedenen Schichten und Cloud-Services, wie aus Abbildung 3 zu erkennen ist.

Service Layer

Unter „Compute“ finden sich dedizierte Server (auch Bare Metal) sowie virtuelle Maschinen (kurz VMs). Der Speicher steht mit „Storage“ (in Form von lokalem Speicher, Block, File, Object und Archive-Speicher) zur Verfügung. Oracle-Datenbanken gibt es in den verschiedenen Ausprägungen: Shared, dediziert, Exadata und Autonomous. Ergänzt um „Edge“-Services wie DNS, EMAIL und WAF, die im vorhergehenden Kapitel beschrieben wurden. Weitere nicht im Bild aufgeführ-

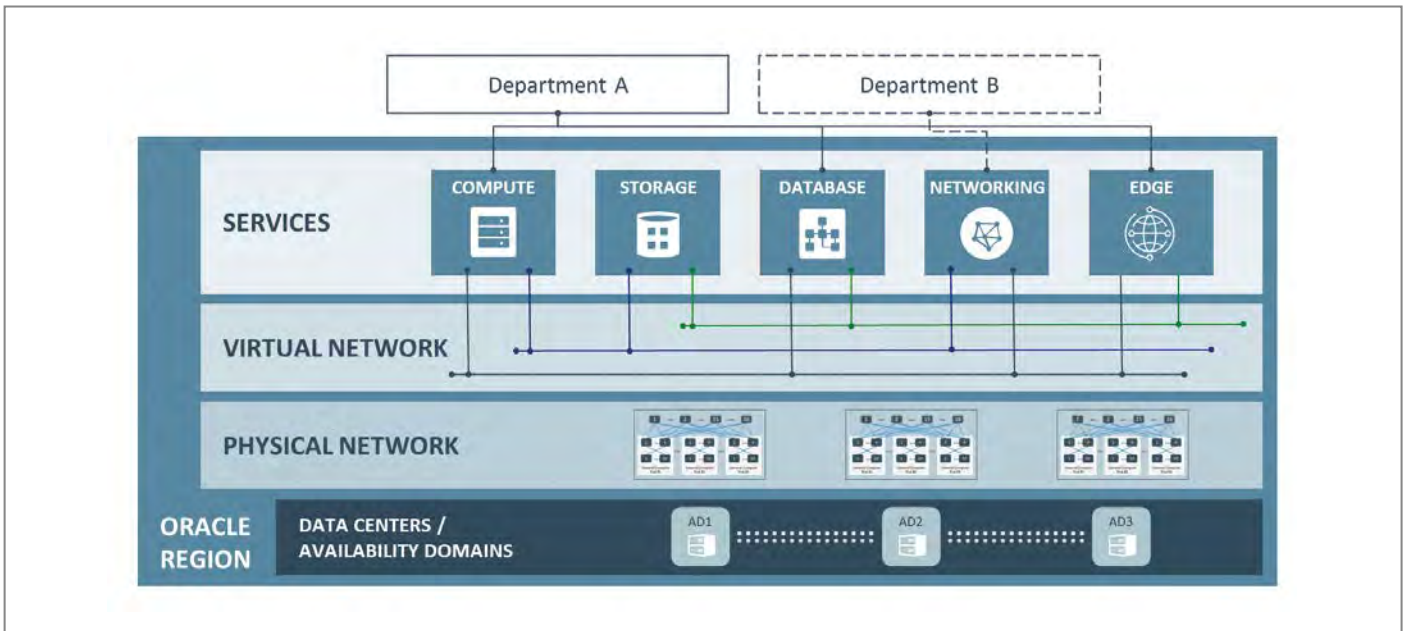


Abbildung 3: Funktionalitäten in der Infrastructure Cloud von Oracle

te Services sind Container, Container Engines wie Kubernetes, Functions und Container Pipeline (CI/CD) sowie die schon länger bekannten Oracle PaaS- Services wie BI-Services etc.

Für die Verwaltung werden in den Services Identitäts- und Zugriffsmanagement, Verschlüsselung inklusive Key Management, Monitoring und Auditierung bereitgestellt. Das enthaltene Cloud Tooling unterstützt bei den typischen Tätigkeiten wie Erzeugen, Starten, Stoppen, Löschen von Services, Backups, Restores oder Patching.

Virtual und Physical Network Layer

Im Oracle-Cloud-Infrastructure-Netzwerk gibt es keine Noisy-Neighbor-Probleme im Netz und in den Instanzen. Da keine Over-Subscription erfolgt, werden SLAs sowohl für Netzwerk-Bandbreiten als auch für Latenz-Zeiten zugesagt. Es erfolgt eine Isolierung der Kundennetzwerke auf Layer 3. Dies wird durch Software Defined Networks (SDNs) ermöglicht, die physisch auf einer speziellen Architektur, sogenannten flachen Clos-Netzwerken, basieren und Off-Box-Netzwerkvirtualisierung ohne zentralen Netzwerk-Hypervisor nutzen.

Der Kunde definiert sein Virtuelles Cloud-Netzwerk (VCN) unter Verwendung aller bekannten Sicherheitsfunktionalitäten wie IPs (public, private), Subnetze (public, private), Routingtabellen und Security Lists (Firewalls). Durch vom

Kunden konfigurierte Gateways werden die Netzübergänge mit dem Internet, einem VPN (IPSec), zwischen den Subnetzen und zwischen Regionen kontrolliert. Es kann eine private Verbindung zwischen dem Rechenzentrum des Kunden und der Oracle Cloud implementiert werden. Durch diese Maßnahme sind das Layer-3-Netzwerk sowie offene Ports und erlaubte Protokolle vollständig unter der Kontrolle des Kunden.

Regionen, Availability Domains und Cloud-Rechenzentren

Oracle Cloud Infrastructure bietet Kunden die Möglichkeit, sich ihre Cloud-Region aus den verschiedenen Cloud-Service-Lokationen auszusuchen wie zum Beispiel die Region Frankfurt. Jede Region besteht aus drei Standorten mit örtlich voneinander getrennten Rechenzentren (5-15 km), den sogenannten Availability Domains. Dies sind somit nicht einfach nur Zonen, die lediglich innerhalb eines Rechenzentrums durch Brandschutztüren voneinander getrennt sind. Availability Domains innerhalb der gleichen Region sind durch ein sicheres, schnelles und latenzarmes Netzwerk verbunden. Die Verbindungen sind verschlüsselt und je nach Servicetyp erfolgt transparent eine redundante Speicherung an mehreren Standorten innerhalb der Region, z.B. Frankfurt. Dadurch sind die Daten selbst bei einem Ausfall eines Rechenzentrums (einer AD) sicher gespeichert. Backup-

und Snapshot-Funktionen stehen dem Kunden zur Verfügung. Die Availability Domains können auch explizit vom Kunden genutzt werden, um hochverfügbare Anwendungen aufzubauen.

Geo-Redundanz kann durch den Kunden erzielt werden, indem die Services über Regionen hinweg gespannt werden. Der Oracle Edge DNS / Oracle DYN kann dabei dieses Load Balancing bereitstellen.

Die Rechenzentren selbst sind entsprechend redundant ausgelegt (ANSI / TIA-942-A Tier III oder Tier IV mit N2 Redundanz der kritischen Komponenten) und verfügen über zwei unabhängige Points of Presence (PoP) je Region. Kunden haben die Möglichkeit, ein Datacenter direkt, auch mehrfach, via Fast-Connect anzubinden.

Isolierung der Umgebungen

In Oracle Cloud Infrastructure werden Rechen- und Netzwerk-Ressourcen isoliert, um sicherzustellen, dass kein Kunde die Daten oder den Datenverkehr anderer Kunden lesen kann. Dazu ist die sogenannte Off-Box-Netzwerk-Virtualisierung implementiert, die die Netzwerk- und I/O-Virtualisierung aus dem Software-Stack herausnimmt und in Hardware und das Netzwerk einfügt. Compute-, Netzwerk- und Speicherlösungen sind dabei genauso wie bei On-Premises-Systemen vollständig steuer- und konfigurierbar. So werden die Vorteile der Public Cloud (On-

Demand, Self-Service, Skalierbarkeit etc.) mit den Vorteilen, die normalerweise mit lokalen Umgebungen verbunden sind – nämlich Vorhersagbarkeit, Leistung und Kontrolle –, kombiniert.

In der Oracle Cloud Infrastructure wird je nach Typ keinerlei Oracle-Software zum Management der Cloud-Funktionalität auf den Ressourcen ausgeführt. Damit hat Oracle Cloud Operations keinen Zugriff auf Server oder Instanzen der Kunden. So entsteht eine zusätzliche Isolationsstufe zum Cloud-Service-Provider. Oracle ermöglicht den Kunden die Konfiguration von Bare Metal Servern zur alleinigen Nutzung und damit von allen anderen Mandanten isoliert. Der Kunde teilt sich dann keinen Prozessor, Speicherplatz oder Netzwerk mit anderen Cloud-Nutzern.

Oracle Cloud Bare Metal, VM- und DB-Instanzen verwenden als Standard schlüsselbasiertes SSH für die Verbindung. Die Schlüssel werden vom Kunden festgelegt. Im Falle der Datenbank vergibt der Kunde zusätzlich die DB-Credentials. Alle Zugänge müssen generell explizit vom Kunden konfiguriert werden, nur der Port 22 ist geöffnet, um nach der Provisionierung auf die Instanzen zugreifen zu können. Bei Services, die keinen Zugriff auf Port 22 ermöglichen, ist ein äquivalenter Port mit verschlüsselter Verbindung geöffnet.

Es können eigene oder von Oracle bereitgestellte Images verwendet werden. Alle von Oracle bereitgestellten Images verfügen über sichere Standardeinstellungen, wie zum Beispiel Firewalls auf Betriebssystem-Ebene, die standardmäßig aktiviert sind.

Verschlüsselung

Verschlüsselung erfolgt sowohl am Speicherort als auch auf dem Transportweg, zum Beispiel beim Zugriff von außen oder zwischen den Instanzen.

Load Balancer unterstützen End-to-End-SSL, SSL-Tunneling oder SSL-Terminierung. Im Standard wird TLS 1.2 und Forward-Secrecy Chiffre verwendet. Kundeneigene Zertifikate können verwendet werden.

Alle gespeicherten Daten im externen Block Storage einschließlich Boot Volumes, Objekt Storage und File Storage werden durch Verschlüsselungen geschützt. Die Verschlüsselungs-Keys werden im Standard von Oracle gemanagt.

Um Kunden zu ermöglichen, die Block- und Object-Storage-Schlüssel selbst zu verwalten, stellt Oracle eine Hardwarebasierte Schlüsselverwaltung bereit. Die Schlüsselverwaltung bietet zentralisierte Verwaltungsfunktionen auf Basis einer isolierten Partition in FIPS 140-2 Level-3-zertifizierten Hardware-sicherheitsmodulen (HSMs).

Neben den oben angeführten Verschlüsselungen des Speichers unterliegen die Schlüssel der Oracle-Cloud-Datenbanken der Verwaltung des Kunden. Die Datenbanken sind ebenfalls „zwangsverschlüsselt“: Ein Key muss beim Erstellen der DB erzeugt werden und wird dann durch den Kunden verwaltet. Neben der Key Rotation kann der DB Key auch auf ein On-Premises-Key-Management-System ausgelagert werden.

Security Controls

Zur Unterstützung von Compliance und Governance bietet die Oracle Cloud integrierte Governance-Funktionen. Durch eine Kombination von Identitäts- und Zugriffsmanagement, rollenbasierten Zugriffskontrollen, Monitoring, Auditfunktionen und differenzierten Regelwerken können auch bestehende Governance-Mechanismen beibehalten werden.

Wird ein Cloud Account angelegt, bekommt der Kunde den notwendigen administrativen Zugriff. Mit diesem Zugriff können Berechtigungsstrukturen definiert sowie weitere Accounts angelegt und verwaltet werden. Benutzer können entweder im Cloud Service über die Oberfläche angelegt, von außen provisioniert oder im Rahmen des Single Signon (SSO) übernommen werden. Diese SSO „on-the-fly“-Übernahme erfolgt optional basierend auf dem Federation-Standard via SAML, etwa von einem Active Directory Federation Service aus. Passwort-Policies können bei Benutzern mit direkter Anmeldemöglichkeit definiert werden. Starke Authentifizierung und kontextbasierte Authentifizierung können im jeweiligen führenden Anmeldesystem konfiguriert werden.

Das Zugriffsmanagement ermöglicht die Umsetzung des „least privilege“-Prinzips und optional die direkte Steuerung der Rechte vom lokalen oder fremden Identity-Management-System aus, wie z.B. die Deaktivierung des Benutzers beim Verlassen des Unternehmens. Über

die Steuerungsmöglichkeit können auch Abteilungen und/oder Test- und Produktionssysteme abgebildet werden.

Alle Ressourcen sind im Standard geschützt und können nur mit entsprechenden Berechtigungen angelegt, geändert oder gelöscht werden. Dazu werden Gruppen definiert, Policies erstellt und die Gruppen den Benutzern oder Services zugewiesen. Die Architektur lässt den angemeldeten Account oder Service nur diejenigen Operationen ausführen, für die er berechtigt wurde. Das Berechtigungsmanagement wirkt übergreifend über Regionen und Availability Domains.

Überprüfbarkeit/Audit

Durch Nutzung der Oracle Cloud profitieren Kunden direkt von der umfassenden Expertise von Oracle und den kontinuierlichen Investitionen in die Sicherheit. Die Entwicklung von Cloud Services erfolgt unter ISO-27001:2013-Standards und den Oracle Security Policies. Die Oracle Cloud Services in Frankfurt besitzen SSAE16/ ISAE3402 (SOC-1 Typ2), AT101 (SOC-2 Typ2), SOC-3, PCI-DSS, HIPAA, ISO-270017- und -270018-Attestierungen sowie ISO/IEC 27001:2013-Zertifizierungen. Weitere, auch C5, folgen in Kürze. Anpassungen bezüglich der Datenschutzgrundverordnung sind erfolgt.

Die Rechenzentren, die die Cloud Services betreiben, sind zertifiziert nach ISO 9001, ISO 14001, OHSAS 18001, ISO 27001, ISO 50001 und PCI-DSS. Vom Typ sind sie ANSI / TIA-942-A Tier-III- oder Tier-IV-Standards des „Uptime Institute“ und der Telecommunications Industry Association (TIA).

Alle Zugriffe der Web-Konsole oder API/SDKs laufen über ein zentrales API und werden daher identisch und vollständig durch den enthaltenen Oracle Cloud Infrastructure Audit Service protokolliert. Der Audit-Service stellt die Auditdaten über eine grafische Konsole und als JSON-Daten über ein authentifiziertes, filterbares Abfrage-API zur Integration in einem übergeordneten System bereit. Weitere Services von Oracle, die AI/Künstliche Intelligenz bzw. Machine Learning verwenden, stehen für das Monitoring zur Verfügung. Dies sind der Oracle Cloud Access Security Broker (CASB) und das Oracle Management Cloud Security Monitoring. Sie überwa-

chen die Sicherheit der Oracle Cloud Infrastructure von außen. Dazu werden Policies eingesetzt, das Benutzerverhalten wird gemonitort und es erfolgt eine Analyse gegen potenzielle Threats (Gefährdungen). Policies sind eine Kombination aus vordefinierten, für diesen Service spezifischen Sicherheits-Policies sowie von Kunden konfigurierbaren Sicherheits-Policies. Ein Beispiel für eine Policy ist die Warnung bei öffentlich zugänglichen OCI Object Storage Buckets. Eine Erkennung von Anomalien erfolgt mithilfe von Machine Learning (sogenanntes UBA – User Behavior Analytics). Parallel dazu erfolgt eine Bedrohungsanalyse zum Erkennen von risikobehafteten Interaktionen. Dazu werden auch Threat Feeds von Drittanbietern integriert wie z.B. verdächtige IP-Adressen.

Hochverfügbarkeit

Der Hochverfügbarkeit zugrunde liegen die Redundanz der Komponenten und die mehrfache Speicherung der Daten

(siehe Abschnitt „Regionen, Availability Domains und Cloud-Rechenzentren“ im Anschluss an *Abbildung 3*).

Fazit

Oracle stellt mit der Oracle Cloud Infrastructure in Frankfurt eine Cloud-Umgebung bereit, mit der unternehmenskritische Workloads unter Wahrung der Sicherheitsanforderungen aus der Cloud heraus genutzt werden können. Sichere hybride Umgebungen werden durch die bereitgestellten Mechanismen unterstützt. Kunden erhalten die Kontrolle und Absicherung unter anderem durch Security und Isolation per Default, Compliance und Cyber-Security. Für das Daten-Hosting kann die entsprechende Region, z.B. Frankfurt, gewählt werden.

Aufgeführte Services wie Functions, CDN, WAF und DNS-basiertes Load Balancing standen zum Zeitpunkt der Erstellung des Artikels erst unter Controlled

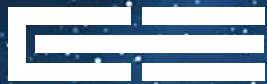
Availability innerhalb der OCI zur Verfügung. WAF und DNS sind und bleiben wie bisher als eigenständige Oracle-Dyn-Services nutzbar.

Weitere Informationen

Oracle Cloud Security: https://cloud.oracle.com/iaas_compliance



Michael Fischer
Michael.Fischer@oracle.com



Das E-3 Magazin

Information und Bildungsarbeit von und für die SAP-Community

**Wir waren zwar nicht die Ersten
auf dem Mond,
dafür sind wir die Ersten,
die unabhängig
über SAP® berichten.**





Datenbank in der Wolke – Teil 1: Abrechnungsmodelle in der Oracle Cloud

Borys Neselovskyi, Opitz Consulting Deutschland GmbH

Die Cloud-Technologie ist heute nicht mehr wegzudenken. Viele Unternehmen nutzen bereits Public Cloud Services oder betreiben einige Teile ihrer Infrastruktur in der Cloud. Auch der Datenbankbetrieb wird immer häufiger von eigenen Rechenzentren in die Cloud verlagert. Die Vorteile der Cloud sind dabei vielfältig. Sie reichen von konkreten Kosteneinsparungen über die Entlastung von Fachkräften bis hin zu einem Mehr an Übersichtlichkeit und Flexibilität. Dieser Artikel ist der erste Teil einer Artikelserie zu diesem Thema. In diesem Teil sollen zunächst die Vorteile der Cloud diskutiert und danach Abrechnungsmodelle sowie verschiedene Dienste von Oracle im Detail vorgestellt werden. In zwei weiteren Teilen, die in den nächsten Ausgaben des Red Stack Magazin erscheinen, wird es ausführlich um Infrastrukturen in der Cloud sowie um konkrete Kundenszenarien gehen.

Die Vorteile des Cloud Computing

Die Anbieter von Cloud Services werben seit Jahren sehr intensiv für ihre Dienste. Dennoch zögern viele Unternehmen und sind sich unsicher, ob sie den Schritt in die Cloud wirklich verantworten können. Im Folgenden erhalten Sie eine kleine Einführung in die Cloud-Thematik: Wie gestalten sich die wichtigsten Vorteile im Einzelnen? Und was sind typische Treiber des Cloud Computing in konkreten Anwendungsfällen?

Kosteneinsparungen

Der erste Grund, warum sich Unternehmen oftmals mit Cloud-Lösungen beschäftigen, sind die geringeren Kosten, mit denen deren Anbieter werben. Wenn eine On-Premises-Infrastruktur in die Cloud überführt wird, können tatsächlich nicht unerhebliche Kosten, zum Beispiel für den Betrieb von Rechenzentren, eingespart werden. Auch das Finanzierungsmodell „Pay as you go“ (PAYG) bietet Kostenvorteile: Dabei wird die tatsächliche Nutzung von Services in Rechnung gestellt. So kann in der Cloud zum Beispiel eine Datenbankumgebung, die nur in Geschäftszeiten verfügbar sein muss, abends und am Wochenende heruntergefahren werden. Ressourcen und der durch die Infrastruktur belegte Storage werden dann nur für die genutzten Zeiten abgerechnet.

Entlastung von Mitarbeitern

Ein weiterer Vorteil, den die Cloud mit sich bringt, ist die Entlastung oder Freistellung hochqualifizierter Mitarbeiter, die für die Aufrechterhaltung der Infrastruktur zuständig sind. Diese können sich mit der Cloud auf das Kerngeschäft ihres Unternehmens konzentrieren und neue Herausforderungen angehen.

Komplexität reduzieren

Darüber hinaus bietet die Cloud vielen Unternehmen auch fachliche Vorteile. Zum Beispiel hilft sie dabei, die Komplexität bestehender Infrastrukturen zu reduzieren: Die Infrastrukturen in der Cloud basieren auf Standards. Installationen und Lifecycle-Operationen erfolgen in der Cloud in der Regel automatisch oder nach bestimmten Vorlagen und Regeln. So können Infrastruktur-

Komponenten einheitlich installiert und gewartet werden.

Einfache Modernisierung von Alt-Systemen

Auch zur Modernisierung der IT-Landschaft kann der Umzug in die Wolke beitragen. So bieten Cloud Services die Möglichkeit, bestehende IT-Landschaften zu inventarisieren und alte, nicht genutzte Anwendungen zu deaktivieren. Alle Programme werden auf ihre Cloud-Affinität hin überprüft und alte Applikationen, die weiterverwendet werden, werden modernisiert oder durch moderne Cloud-Dienste ersetzt.

Höhere Flexibilität

Heute ist am Markt eine schnelle Anpassungsfähigkeit entscheidend für den Geschäftserfolg. Cloud-Dienste können hier gegebenenfalls zu einem Vorsprung verhelfen: Sie ermöglichen eine schnelle Veränderung, weil sie sich in der Regel sehr gut skalieren lassen. Bei Performance-Engpässen können Cloud-Umgebungen so sehr schnell erweitert werden. Dabei werden die Ressourcen eines Servers in der Cloud erweitert (vertikale Skalierung) oder zusätzliche Maschinen (horizontale Skalierung) provisioniert.

Typische Cloud-Treiber

Was treibt Unternehmen dazu, ihre Infrastrukturen in die Cloud zu verlagern? Welche Use Cases sind für den Übergang in die Cloud sozusagen prädestiniert?

Oft wird eine Datenbank nur für eine kurze Zeit benötigt, um etwas zu testen, wie zum Beispiel die Migration auf eine neue RDBMS-Version. Oder eine Anwendung soll aktualisiert werden. Für solche Szenarien wird eine Datenbank benötigt, die sehr schnell bereitgestellt und nach dem Testen wieder gelöscht werden kann. Kapazitäten in eigenen Rechenzentren sind dafür oft zu begrenzt. Häufig reichen auch bereits erworbene On-Premises-Lizenzen nicht aus. So könnte ein Oracle-Kunde beispielsweise eine Enterprise Edition ohne die Partitioning-Option erworben haben. Getestet werden soll aber eine Anwendung, die auf der Partitionierung aufsetzt.

In der Cloud gibt es diese Grenzen nicht. Dort können Datenbanken sehr

einfach und schnell provisioniert werden. Wenn die Tests abgeschlossen sind, wird die Datenbank mit wenigen Klicks gelöscht. Kosten fallen dabei nur für die tatsächliche Nutzung der Datenbank an.

In einem anderen Fall reicht die Kapazität des Rechenzentrums nicht mehr für den Betrieb: Die Datenbanken sind sehr schnell gewachsen und die daruntergelegene Infrastruktur muss ständig erweitert werden. Wenn die Kapazitätsgrenzen erreicht sind, müssten entweder bestehende Rechenzentren ausgebaut werden oder man verlagert Datenbanken in die Cloud, wo die Ressourcen nach Bedarf skaliert werden können.

Ein weiterer Fall, den wir ebenfalls häufig beobachten, ist der Austausch von Servern aufgrund von Lifecycle-Kriterien. Auch hier kann es sinnvoll und legitim sein, eine Verlagerung in die Cloud zu erwägen.

Einfacher ist die Entscheidung, wenn es beispielsweise um ein neues Data Warehouse geht, das auf der „grünen Wiese“ implementiert werden soll. Solche Projekte werden mittlerweile oft direkt in einer Cloud umgesetzt.

Auch für viele Startup-Unternehmen ist der IT-Betrieb in der Cloud die schnellste und kostengünstigste Option. Diese Firmen verfügen in der Regel weder über Kapazitäten in Rechenzentren noch über eine ausreichende Anzahl von Spezialisten für den On-Premises-Betrieb. Mithilfe passender Cloud Services können sich Startups auf ihr Kerngeschäft konzentrieren und brauchen kein Know-how im Infrastruktur-Bereich aufzubauen.

Vorbereitung ist alles!

Wie wir sehen, bietet die Migration in die Cloud viele Vorteile. Da die Reise in die Wolke allerdings natürlich auch viele Veränderungen technischer, rechtlicher und ethischer Art mit sich bringt, sollte sie gut vorbereitet werden.

Wichtig ist es, sich im Vorfeld mit der Cloud vertraut zu machen und die Umstellung detailliert zu beschreiben. Hier einige Fragen, die dabei helfen können:

- Ist meine Anwendung Cloud-fähig?
- Wo sind meine Daten in der Cloud? Wer darf auf diese zugreifen?
- Wie sicher sind meine Daten in der Cloud?

Konfiguration	Anzahl CPU	PGA/SGA/ In-Memory (GB)	DB Storage (GB)	PAYG – Ein Monat	Monatlicher Flexpreis
X20	1	3,5/4	20	228 €	152 €
X250	1	3,5/4	250	521 €	347 €
X500	2	7,5/7,5	500	1.042 €	694 €
X1000	4	15/15	1024	2.083 €	1.389 €
X1000IM	4	15/15/10	1024	5.208 €	3.472 €

Tabelle 1: Die Varianten des Exadata Express Cloud Service im Überblick (s. Quelle: [2])

- Kann der Cloud-Betreiber meine Daten lesen?
- Welcher Datenbank-Service passt am besten zu meinen Anforderungen?
- Was kostet der Datenbank-Betrieb in der Cloud? Welche Abrechnungsmöglichkeiten gibt es?
- Wie migriere ich meine Datenbank in die Cloud?
- Ist der Betrieb in der Cloud für die modernen Sicherheitsanforderungen geeignet?
- Kann die Cloud-Infrastruktur meine Performance-Anforderungen erfüllen?
- Wie binde ich meine Datenbank in der Cloud an Legacy-Systeme an?
- Ist es möglich, Teile meiner Infrastruktur in der Cloud und andere On-Premises zu betreiben? Welche Risiken bestehen bei einem Hybrid-Cloud-Einsatz?
- Kann ich für meine Cloud-Datenbank On-Premises eine Data Guard Standby bauen? Welche Netzwerk-Bandbreite brauche ich?
- Wie verbinde ich meine Cloud-Umgebung netzwerktechnisch mit vorhandenen On-Premises-Systemen?
- Wie werden in der Cloud die SLAs für Dienste definiert und eingehalten?
- Die Wartungsarbeiten in der Cloud verursachen Downtime. Wie kann ich Datenbanken, die rund um die Uhr online verfügbar sein müssen, in der Cloud betreiben?
- Wie sind die Zuständigkeiten für die Cloud-Infrastruktur verteilt? Wie sind die Verantwortlichkeiten zwischen Cloud-Anbieter und Cloud-Nutzer geregelt?
- Muss ich mich in der Cloud um Backups kümmern?
- Welche Datenbankoptionen stehen zur Verfügung?

Fragen wie diese sollten unbedingt beantwortet werden, bevor ein Unternehmen

die Reise in die Cloud wagt. Ebenfalls ist sehr wichtig zu verstehen, wie einzelne Cloud Services abgerechnet und finanziert werden. Im Folgenden betrachten wir die wichtigsten Abrechnungsmodelle daher einmal etwas genauer.

Abrechnungsmodelle in der Oracle Cloud

Bevor es an die eigentliche Nutzung geht, sind noch einige Voraussetzungen zu erfüllen und es gilt, ein passendes Modell für die Abrechnung mit dem Cloud Provider auszuwählen.

Will ein Unternehmen beispielsweise die Cloud-Dienste von Oracle nutzen, so muss es zunächst sogenannte „Universal Credits“ erwerben. Diese ermöglichen das Erstellen und Betreiben von Ressourcen in der Cloud. Für die Abrechnung seiner Cloud Services bietet Oracle die folgenden Modelle an:

Abrechnung nach Verbrauch

Beim Pay-as-you-go-Modell (PAYG) rechnet Oracle die tatsächlich genutzten Ressourcen nachträglich ab. Der Anbieter generiert dafür monatlich eine Rechnung. Benötigt der Cloud-Nutzer beispielsweise eine Testdatenbank, die nur in den Geschäftszeiten verfügbar sein muss, werden auch nur die Kosten für die tatsächliche Nutzung abgerechnet.

Abrechnung per Dauerauftrag

Möchte ein Unternehmen bestimmte Datenbanken über eine längere Zeit in der Cloud betreiben, lohnt es sich vielleicht, einen Dauerauftrag mit Oracle abzuschließen. Dieses Modell nennt sich „Monatlicher Flexpreis“. Die Abrechnung erfolgt im Voraus mit einer Laufzeit von mindestens einem Jahr. Je länger die Vertragsdauer angelegt ist,

desto größer fällt die Ermäßigung auf den Listenpreis aus.

„Bring Your Own License“

Unter bestimmten Voraussetzungen können Kunden bereits erworbene On-Premises-Lizenzen in die Cloud übertragen. Dieses Modell nennt sich „Bring Your Own License“ (BYOL) und deckt, sehr vereinfacht gesagt, die Kosten für die Cloud-Infrastruktur, schließt jedoch Lizenzen aus, die bereits direkt von Oracle erworben wurden. Die Nutzung von BYOL bietet sich beispielsweise an, wenn eine On-Premises-Datenbankumgebung in die Cloud migriert wird.

Weitere Informationen zu den grundsätzlichen Abrechnungsmodellen von Oracle siehe [1].

Die Oracle Cloud unter der Lupe

Oracle bietet eine Reihe von Cloud Services für seine Datenbankkunden an. Die Angebotsskala reicht von einer kleinen und preiswerten Entwicklungsumgebung bis zu missionskritischen Datenbanken, die höchsten Sicherheits-, Verfügbarkeits- und Performance-Anforderungen entsprechen. Nachfolgend haben wir einzelne Cloud-Angebote für Datenbanken unter die Lupe genommen.

Oracle Exadata Express Cloud Service

Mit Stand vom August 2018 bietet der Oracle Exadata Express Cloud Service eine dedizierte Container-Datenbank, eine sogenannte Pluggable Database (PDB), in der Version 12.2 auf einer Exadata-Maschine in der Oracle Cloud. Mit diesem Service können die Endanwender alle Eigenschaften der Enterprise Edition nutzen, inklusive Optionen wie Partitioning oder Advanced Compression.

Der Exadata Express entspricht dem Database-as-a-Service-Prinzip und wird vollständig durch den Cloud-Anbieter verwaltet. Oracle übernimmt demnach die Wartung aller Systemkomponenten einschließlich der Datenbank. Die Cloud-Nutzer kümmern sich lediglich um die Benutzerverwaltung in der Datenbank sowie um eigene Applikationen.

Weitere Annehmlichkeiten: Neue Cloud-Datenbanken können mit wenigen Klicks in der webbasierten Konsole provisioniert werden. Der Zugriff auf die Datenbank in der Cloud erfolgt über die SQL*Net-Schnittstelle. Es können herkömmliche Tools wie SQL*Plus, SQL Developer und viele andere für die Arbeit mit der Datenbank genutzt werden. Entwickler können mit der installierten APEX-Komponente neue Anwendungen erstellen. Die Datenbank kann über die herkömmlichen JDBC-Treiber auch mit Middleware-Komponenten verbunden werden. Die administrativen Aufgaben wie das Starten bzw. Stoppen des Dienstes können entweder über die webbasierte grafische Cloud-Konsole oder über das Kommandozeilen-Tool Cloud CLI erledigt werden. Es ist zudem möglich, die Datenbank mithilfe des REST API zu administrieren. Doch es gibt auch Stolpersteine, die man bei der Wahl dieses Service unbedingt mitbedenken sollte:

- Die vom Benutzer verwalteten User-Definitionen und Datenbank-Objekte werden nicht automatisch gesichert. Man kann eigene Inhalte per Export/Import über die administrativen Tools in den Cloud Storage sichern bzw. aus dem Storage restaurieren. Dafür muss zusätzlicher Cloud Storage für Backups erworben werden.

- Exadata Express Cloud Service bietet keinen Zugriff auf das Betriebssystem des Datenbankservers.
- Der Exadata Express Cloud Service ist für Datenbanken mit hohen Verfügbarkeitsanforderungen nicht geeignet, da es nicht möglich ist, RAC- und Data-Guard-Optionen in der Verbindung mit dem Exadata Express zu nutzen.

Für die Abrechnung bietet Oracle Exadata Express Service fünf Varianten an, die sich in der Ausstattung unterscheiden. Bei der kleinsten Konfiguration mit der Bezeichnung X20 handelt es sich um eine Container-Datenbank, die mit einer CPU, drei Gigabyte Memory für Shared Global Area (SGA), vier Gigabyte Memory für Program Global Area (PGA) und 20 Gigabyte Datenbank-Storage ausgestattet ist.

Die Abrechnung erfolgt nach dem Modell „Pay as you go“, es werden also die tatsächlich verbrauchten Dienste abgerechnet. Wer die Datenbanken langfristig nutzen möchte, kann Geld sparen, indem er das Modell „Monatlicher Flexpreis“ auswählt. Die minimale Laufzeit beträgt ein Jahr; Oracle gewährt in diesem Modell einen Rabatt auf den Stundenpreis.

Tabelle 1 zeigt alle Exadata-Express-Cloud-Service-Varianten im Überblick. Die Werte in der Spalte „PAYG – Ein Monat“ basieren auf 744 Stunden beziehungsweise 31 Tagen mit jeweils 24 Stunden.

Oracle Database Cloud Service

Der Oracle Database Cloud Service bietet die Möglichkeit, verschiedene Datenbank-Typen und -Editionen in der Cloud zu erstellen und zu betreiben. Anders als beim Exadata Express bekommt man nach der Provisionierung einen oder mehrere Server, Netzwerkressourcen, eine Datenbank und einen Storage-Bereich für die Speiche-

rung von Datenbankdateien und Sicherungen. Server können virtualisiert oder als Bare Metal bereitgestellt werden. Der Provisionierungs-Prozess kann bequem über die grafische Cloud-Konsole oder mit dem Kommandozeilentool Cloud CLI angestoßen werden. Das Erstellen von Zweiknoten-RAC-Umgebungen oder die Data-Guard-Konfiguration erfolgen sehr einfach mit wenigen Klicks von der Oberfläche aus. Jeder, der bereits eine RAC-Umgebung in einer herkömmlichen Umgebung aufgebaut hat, wird diese Vereinfachung der Cloud-Provisionierung schätzen. Skalierung von Ressourcen können mithilfe einiger Cloud Tools sehr schnell durchgeführt werden. Der Database Cloud Service unterstützt die folgenden Datenbanktypen:

- Single Instance: Datenbank, die auf einem virtuellen oder physikalischen Server betrieben wird.
- Cluster (RAC) Datenbank: Datenbank, die auf zwei Servern im Cluster betrieben wird. Diese Umgebung verfügt über einen gemeinsamen Storage.
- Single Instance mit Data Guard: In der Cloud wird eine Umgebung aus zwei Servern aufgebaut. Primäre und sekundäre Datenbanken werden komplett in der Cloud betrieben. Oft werden dabei Server in unterschiedlichen Verfügbarkeitszonen installiert.
- RAC-Datenbank mit Replizierung mittels Data Guard: In dieser Konstellation werden sowohl primäre als auch Standby-Datenbanken „geclustert“.
- Data Guard Standby for Hybrid Disaster Recovery: Die primäre Datenbank wird in einer On-Premises-Umgebung ausgeführt. Die Standby-Seite läuft in der Cloud. Damit kann die Georedundanz der Datenbank implementiert werden.

Standard Package	Enterprise Package	High Performance	Extreme Performance
<ul style="list-style-type: none"> • Standard Edition 2 • Transparent Data Encryption 	<ul style="list-style-type: none"> • Enterprise Edition • Transparent Data Encryption • Data Masking • RAT • Diagnostics Pack • Tuning Pack • Hybrid Columnar Compression 	<ul style="list-style-type: none"> • Enterprise Package • Multitenant • Partitioning • Advanced Compression • Advanced Security • OLAP/Spatial • Lifecycle/Cloud Management Packs 	<ul style="list-style-type: none"> • Enterprise Package High Performance • RAC/RAC One Node • In-Memory Database • Active Data Guard

Tabelle 2: Die Software-Pakete des Oracle Database Cloud Service im Überblick

Produkt	PAYG (OCPU pro Stunde)	Monatlicher Flexpreis (OCPU pro Stunde)
Standard Package	0,3501 €	0,2334 €5
Enterprise Package	0,7001 €	0,4667 €
High Performance Package	1,4442 €	0,9628 €
Extreme Performance Package	2,188 €	1,4587 €

Tabelle 3: Kosten für Datenbankpakete (s. Quelle [3], Stand: August 2018)

Bei der Auswahl der RDBMS-Versionen kann man in diesem Cloud Service aus dem Vollen schöpfen (Stand August 2018):

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

Der Oracle Database Cloud Service bietet zudem eine Vielzahl von Datenbank-Eigenschaften, die eingesetzt werden können: Im Gegenteil zu herkömmlichen On-Premises-Umgebungen, bei denen man zwischen Standard und Enterprise Edition wählen muss, bietet Oracle in der Cloud mit vier Software-Paketen mehr Flexibilität. So entspricht das Cloud „Standard Package“ der Standard Edition 2, die man in der On-Premises-Welt kennt. Weitere Optionen sind: „Enterprise Package“, „High Performance“ und „Extreme Performance“. In *Tabelle 2* sind die möglichen Software-Pakete mit allen Eigenschaften zusammengefasst.

Noch zwei Hinweise: Die Option „Transparent Data Encryption“ (Abkürzung: TDE) ist in allen Software-Paketen vorhanden. Sie verschlüsselt die Daten in der Datenbank. Außerdem sollte man wissen, dass die TDE-Option in klassischen On-Premises-Umgebungen nur mit der Enterprise Edition in Kombination mit der kostenpflichtigen Option „Advanced Security“ erworben werden kann. *Tabelle 3* zeigt die Kosten für die einzelnen Datenbankpakete. In der Cloud sind alle Datenbanken mit der TDE-Option verschlüsselt.

Shapes

Um Datenbanklandschaften schnell in der Cloud zu erstellen, hat Oracle viele Server-Vorlagen entwickelt. Diese werden als Shapes gekennzeichnet. Shapes stellen unterschiedliche Hardware-Kombinationen dar, die aus diesen Ressourcen bestehen:

- Ein oder mehrere Prozessoren: Diese werden als Oracle CPU (OCPU) gekennzeichnet. Eine OCPU bietet eine Prozessor-Kapazität, die einem physischen Kern eines Intel-Xeon-Prozessors mit aktiviertem Hyper Threading entspricht. Jede OCPU entspricht zwei Threads, die als vCPUs bezeichnet werden.
- Serverspeicher (RAM): Es gibt unterschiedliche RAM-Module, die in Shapes verwendet werden.
- Storage: In einigen Shapes steht ein lokaler Storage zur Verfügung.

Fazit

Oracle bietet mit seinen Cloud Services die Voraussetzungen für den erfolgreichen Betrieb nahezu aller Datenbank-Typen und -Optionen. Sehr komplexe Umgebungen können in der Cloud per Knopfdruck erstellt und skaliert werden. Der Umstieg in die Cloud erfordert aber eine umfangreiche Analyse der verschiedenen Cloud-Angebote und eine gute Vorbereitung. Vor der Wahl eines Cloud Service sollten die eigenen Anforderungen klar definiert und die Lösung von mehreren Seiten betrachtet werden. Dazu zählen finanzielle Aspekte und Risiken, Definition der Umgebung, Klärung von Zuständigkeiten für den Betrieb, Sicherheits- und Verfügbarkeitsfragen und noch viel mehr.

Oracle bietet seine Cloud Services seit mehreren Jahren an und entwickelt seine Cloud-Landschaft ständig weiter. Wie sieht die Infrastruktur derzeit aus? Was steckt dahinter? Im nächsten Teil dieser Artikelserie widmen wir uns dieser Frage.

Quellen:

[1] <https://www.oracle.com/de/cloud/bring-your-own-license/faq/universal-credit-pricing.html>

[2] <http://www.oracle.com/us/products/database/database-exadata-express-ds-3224599.pdf>

[3] https://cloud.oracle.com/de_DE/database/pricing



Borys Neselovskyi
borys.neselovskyi@opitz-consulting.com

Finden Sie die passende Schulung im Oracle-Umfeld auf

university.doag.org

- ▶ Oracle-Technologien
- ▶ IT-Methoden
- ▶ IT-Management



Erhalten Sie als
DOAG-Mitglied einen
exklusiven Rabatt auf
den regulären
Kurspreis.



Cloud Connectivity – Herausforderungen und Lösungen

Daniel Steiger, Trivadis AG

Bereits 1984 hat die damalige Sun Microsystems (heute Oracle) als visionäre Vorreiterin mit dem Slogan „The Network is the Computer“ das Netzwerk ins Zentrum einer modernen, verteilten Infrastruktur-Architektur gerückt. Mit dem Durchbruch des Internets Ende der 90er-Jahre und der darauffolgenden rasanten Entwicklung und Verbreitung der Internettechnologie wurden die Grenzen der traditionellen, standortbezogenen IT-Landschaften mehr und mehr aufgelöst – das Terrain für die Cloud-Computing-Ära war geebnet. 2015 hat Google mit der kernigen Botschaft „The Power is in the Network“ nachgelegt und die strategische Bedeutung der Konnektivität für die Verbreitung und Entwicklung von Cloud-Computing unterstrichen.

In diesem Artikel beschreiben wir die Herausforderungen bei der Anbindung von Cloud-Services (IaaS, PaaS und SaaS) und stellen mögliche Lösungen vor. Am Beispiel der Oracle Cloud zeigen wir die unterschiedlichen Konnektivitätslösungen mit ihren Vorzügen sowie damit verbundenen Bedenken auf und weisen auf die Punkte hin, die in einer Cloud-Connectivity-Strategie zu beachten sind.

Herausforderung Cloud Connectivity

Im Grunde genommen ist die Verbindung zwischen einem Unternehmensnetzwerk und einem Cloud-Provider ein klassischer Data Center Interconnect (DCI). Cloud Computing hat jedoch den Zugriff auf Daten und Services fundamental verändert. Mit der Auslagerung von geschäftskritischen Services und Daten zu einem oder mehreren Cloud Service Providern (CSP) steigen auch die Anforderungen an die Konnektivität zu den Services. Durch die Bereitstellung unterschiedlicher Verbindungsmöglichkeiten für die End-User nimmt die Komplexität der Netzwerkarchitektur zu und verlagert sich in Bereiche, die außerhalb des bisherigen Sicherheitsperimeters liegen. Die Anbindung soll zudem unterschiedlichste Services (IaaS, PaaS und SaaS) unterstützen und ein breites Spektrum an Workloads abdecken.

Um diesen Herausforderungen zu begegnen, ist eine Cloud-Connectivity-Strategie erforderlich. Das Ziel einer solchen Strategie kann wie folgt beschrieben werden: „Bereitstellung einer flexiblen, performanten, sicheren und kostengünstigen Netzwerkverbindung zur Nutzung von unterschiedlichsten Cloud-Services über einen Mix von öffentlichen und privaten Netzwerkverbindungen und Nutzung von physischen und virtuellen Netzwerkkomponenten.“ Wesentliche Punkte der Strategie sind:

- Klärung der Zielsetzung und der Anforderungen bezüglich Zielbild, Verfügbarkeit, Sicherheit, Compliance, Kosten und Betrieb
- Schaffung eines gemeinsamen Verständnisses mit allen Stakeholdern, die Nutzung von Cloud-Diensten im Unternehmen und deren Auswirkung auf die Netzwerkarchitektur betreffend

- Auswahl sinnvoller Szenarien und deren Validierung, beispielsweise im Rahmen eines Proof of Concept
- Feedback: laufende Überprüfung der Resultate und Erfahrungen, um Netzwerkgpässe rasch zu erkennen und zu beheben

Als einen ersten wichtigen Schritt bei der Erstellung eines Konnektivitätskonzeptes empfehlen wir, sich mittels einer Datenfluss- und Netzwerkverkehrs-Analyse Transparenz über die Datenflüsse zu verschaffen:

- Wie groß ist das In-Bound- und Out-Bound-Datenvolumen?
- Wie hoch sind die Lastspitzen? Wann und wie häufig treten sie auf?
- Wie hoch ist der durchschnittliche/maximale Datendurchsatz?
- Welche Protokolle und Ports werden verwendet?
- Wie groß sind die tolerierten Netzwerk-Antwortzeiten (Latenz) der betroffenen Applikationen?
- Wie viele Netzwerkverbindungen/Sessions sind gleichzeitig aktiv?
- Muss der Netzwerkverkehr verschlüsselt werden (encryption in motion)?
- Welche Anforderungen bestehen bezüglich Quality of Service (QoS)?
- Wie ausfallsicher muss die Netzwerkverbindung sein?
- Welche SLA-Anforderungen bezüglich Performance und Bandbreite müssen eingehalten werden?
- Wie sieht das zukünftige Datenvolumen aus (Skalierung, Wachstum, Fluktuation)?
- Welche Cloud-Dienste sollen genutzt werden – SaaS, PaaS oder IaaS – und woher erfolgen die Zugriffe?
- Welche Authentisierungs- und Autorisierungsmethoden werden für die Nutzung der Cloud-Dienste verwendet? Wie sieht die IAM-Architektur aus?
- Wie sind die Anwendungen in der Cloud hinsichtlich ihrer Geschäftskritikalität einzustufen?

Insbesondere bei der Auslagerung zentraler Applikationen, wie beispielsweise eines Customer-Relationship-Management, in die Cloud sind die Netzwerkanforderungen aufgrund der veränderten Datenflüsse und Kommunikationswege sorgfältig zu klären.

**Connectivity-Empfehlung #1:
Beschaffen und analysieren Sie die Daten für die „Verkehrsplanung“.**

Oracle Virtual Cloud Network

Das Virtual Cloud Network (VCN) bildet mit seinen Komponenten ein virtuelles Datacenter in der Oracle-Cloud-Infrastruktur (OCI). Ein VCN hat einen festen Bezug zu einer OCI-Region. Regionen sind über den Erdball verteilte Datacenter-Verbünde, bestehend aus jeweils mindestens drei Availability-Domänen. VCNs können untereinander durch VCN-Peering verbunden werden. Man unterscheidet dabei zwischen Local VCN Peering innerhalb einer Region und Remote VCN Peering bei regionsübergreifenden Verbindungen.

Die Komponenten eines Oracle Virtual Cloud Network (VCN) – Region, Subnet, Availability Domain, Security Lists, Route Table, Dynamic Routing Gateway (DRG), Internet Gateway und VCN Peering – sind in *Abbildung 1* dargestellt. Innerhalb eines VCN werden Public- und Private-Subnetze angelegt. Infrastruktur-Services wie Applikations-Server, Datenbanken etc. werden jeweils einem Subnetz zugewiesen. Jedes Subnetz verfügt über eine Security List (virtuelle Firewall), über die der Zugriff auf die Ressourcen im Subnetz verwaltet wird.

Der Dynamic Routing Gateway (DRG) ist der Eingangspunkt in das VCN und dient als virtueller Router zwischen dem lokalen (privaten) Netzwerk und dem VCN. Über das DRG kann eine Verbindung via IPsec VPN oder Oracle FastConnect in das VCN hergestellt werden. Das DRG dient zudem als Gateway in ein VCN in der gleichen oder in einer anderen Region.

Das Netzwerk macht den Unterschied

Die klassische Internetverbindung, also die physische Nutzung eines öffentlichen Netzwerkes, ist zwar vergleichsweise kostengünstig. Das Public-Internet ist jedoch eine geteilte Ressource, die nach dem „Best Effort“-Prinzip funktioniert. Die Limitierungen bezüglich Performance, Si-

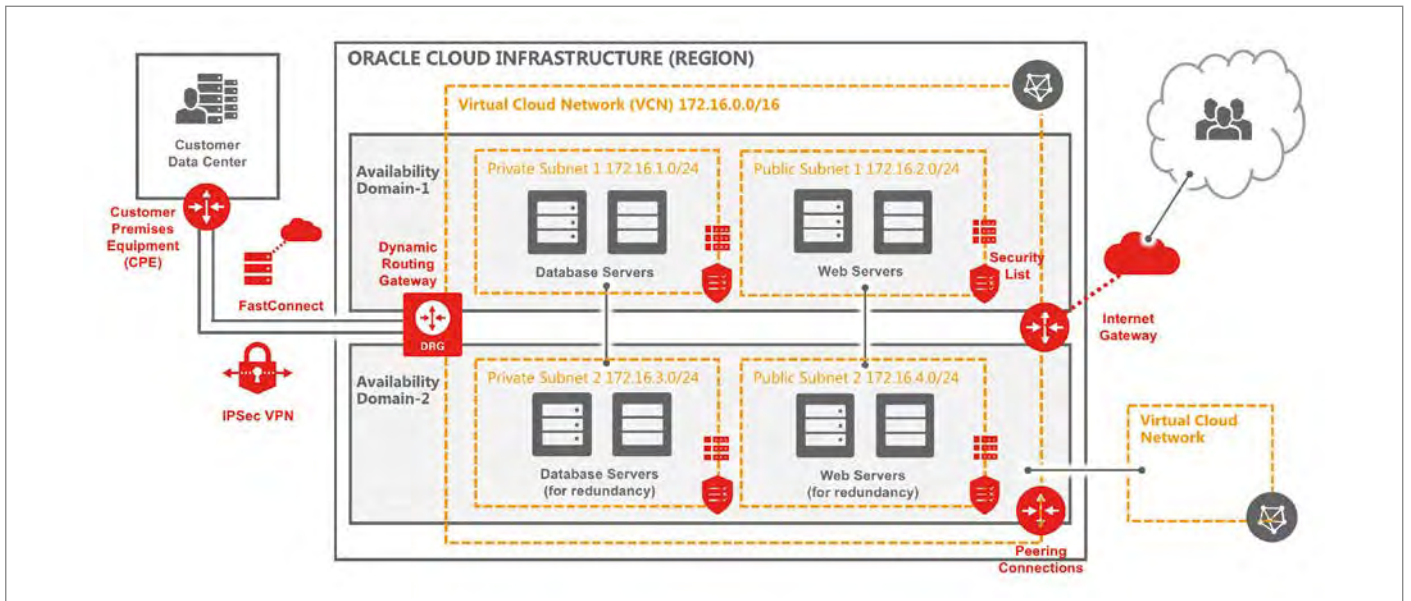


Abbildung 1: Oracle Virtual Cloud Network (VCN)

icherheit und Zuverlässigkeit sollten nicht ausgeblendet werden. Die effektiv verfügbare Bandbreite, der zu erwartende Datendurchsatz, die Latenz und die Verfügbarkeit können über eine öffentliche Verbindung nicht garantiert werden. Die Bandbreite wird von den jeweils gleichzeitig aktiven Nutzern geteilt. Sind viele Nutzer aktiv, kann es zu Verzögerungen bis hin zur Blockierung des Netzwerkverkehrs kommen. Auf Public-Internetverbindungen ist zudem kein QoS (Quality of Service) möglich, weil das Routing der Netzwerkpakete und auch die beteiligten Netzwerkgeräte nicht in der Kontrolle des Netzwerknutzers liegen (siehe Abbildung 2).

Die Qualität des Public-Internets reicht sicherlich für die meisten private Zwecke aus; für Unternehmensanforderungen ist sie jedoch in vielen Fällen ungenügend. Insbesondere wenn unternehmenskritische Anwendungen in der Cloud betrieben werden, muss die entsprechende Netzverbindung eine zuverlässige Performance liefern. Dafür ist eine öffentliche Internetverbindung nicht zu empfehlen; eine private Netzwerkverbindung ist besser geeignet.

In einem Unternehmensnetzwerk kann zudem der Enterprise Connectivity Stack, bestehend aus Firewalls, Web Application Firewalls, Routers, LAN/WAN-Komponenten, Intrusion-Detection-Systemen und Netzwerk-Appliances, sehr komplex sein. Datenpakete, welche die Unternehmensgrenzen über eine nicht-private Verbindung verlassen (respektive

in das Netzwerk eingeschleust werden), müssen aus Sicherheitsgründen durch den gesamten Stack transferiert werden, was letztendlich in erhöhter Latenz resultiert. Bei einer privaten, „trusted“ Anbindung kann auf die umfangreiche Prüfung weitgehend verzichtet werden.

Oracle Cloud Connectivity

Für eine sichere und zuverlässige Verbindung in die Oracle Cloud stellt Oracle zwei Methoden zur Verfügung:

- **IPsec VPN:** Site-to-Site-VPN-Verbindung (Authentisierung und Verschlüsselung)

- **FastConnect:** Private, vom Internet physisch separierte Verbindung, direkt oder via vorintegrierten Netzwerkpartner oder Cloud Exchanger

IPsec VPN ist eine Komponente, die eine Site-to-Site-VPN-Verbindung zwischen einem VCN und dem lokalen Netzwerk bildet. IPsec steht für Internet Protocol Security oder IP Security – ein Protokoll, das den gesamten IP-Verkehr verschlüsselt, bevor die Pakete von der Quelle zum Ziel übertragen werden. Die physische Verbindung erfolgt dabei über die öffentliche Internet-Infrastruktur oder über eine dedizierte Leitung. IPsec kann im Transport Mode oder im Tunnel Mode betrieben werden. Oracle Cloud

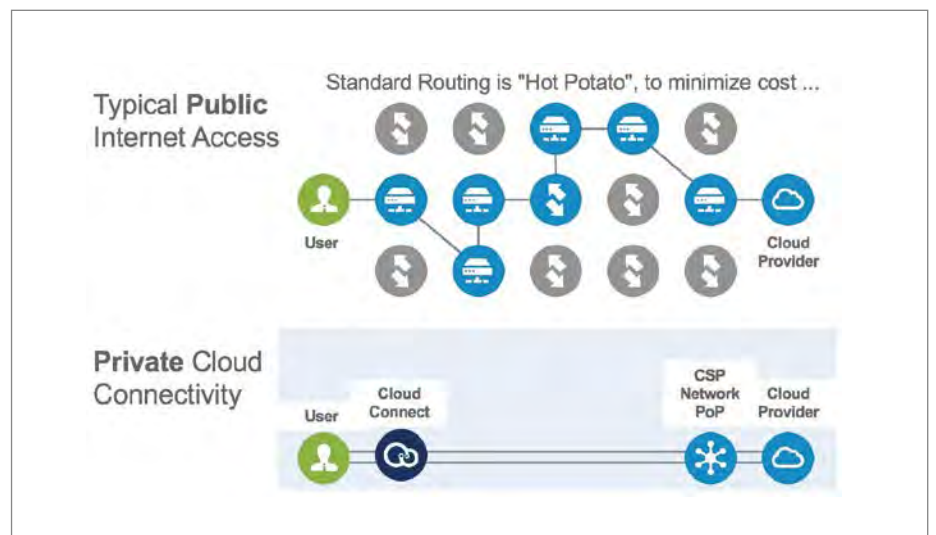


Abbildung 2: Public vs. Private Cloud Connectivity

Infrastructure unterstützt nur den Tunnel Mode. Eine IPSec-VPN-Verbindung besteht aus mehreren IPSec-Tunnels, die statische Routen für den Netzwerk-Traffic zwischen einem Dynamic Routing Gateway (DRG) und dem Customer Premises Equipment (CPE) verwenden (siehe Abbildung 3). Oracle stellt redundante VPN-Tunnels bereit, die sich auf physisch und logisch isolierten Tunnel-Endpunkten befinden. IPSec VPN Site-to-Site-Tunnel bietet folgende Vorteile:

- Für die Datenübertragung werden öffentliche Telekommunikationsleitungen verwendet. Es sind somit keine teuren, dedizierten Mietleitungen von einem Standort zum anderen erforderlich
- Der VPN-Service von Oracle ist kostenlos
- Die internen IP-Adressen der beteiligten Netzwerke und Knoten sind von außen nicht sichtbar
- Die gesamte Kommunikation zwischen Quell- und Zielsystem ist verschlüsselt

Voraussetzung für eine IPSec-Verbindung ist ein IPSec-kompatibler Router auf Kundenseite. Eine Liste der von Oracle verifizierten Hardware ist unter folgendem Link publiziert: <https://docs.cloud.oracle.com/iaas/Content/Network/Reference/CPE-list.htm>.

Oracle FastConnect ist eine dedizierte, private Verbindung über 1 Gbps oder 10 Gbps Ports zur Oracle Cloud Infrastructure. FastConnect ermöglicht eine zuverlässigere und konsistentere Netzwerkverbindung als eine internetbasierte Verbindung. Im Unterschied zu IPSec VPN erfolgt die Verbindung über eine FastConnect-Lokation – ein Rechenzentrum mit einem Oracle FastConnect Edge (siehe Abbildung 4). Die Liste der FastConnect-Edge-Anbieter ist unter dem Link https://cloud.oracle.com/en_US/fastconnect abrufbar.

Aus Latenzgründen befinden sich die Kundenserver idealerweise ebenfalls in der FastConnect-Lokation (Co-Location oder Hosting). Es ist aber auch möglich, einen Remote-Port in einer FastConnect-Lokation zu nutzen und über einen geeigneten Netzwerkprovider das Remote-Datencenter anzubinden. Die Überbrückung der letzten Meile ist natürlich mit zusätzlichen Port- und Netzwerkkosten verbunden.

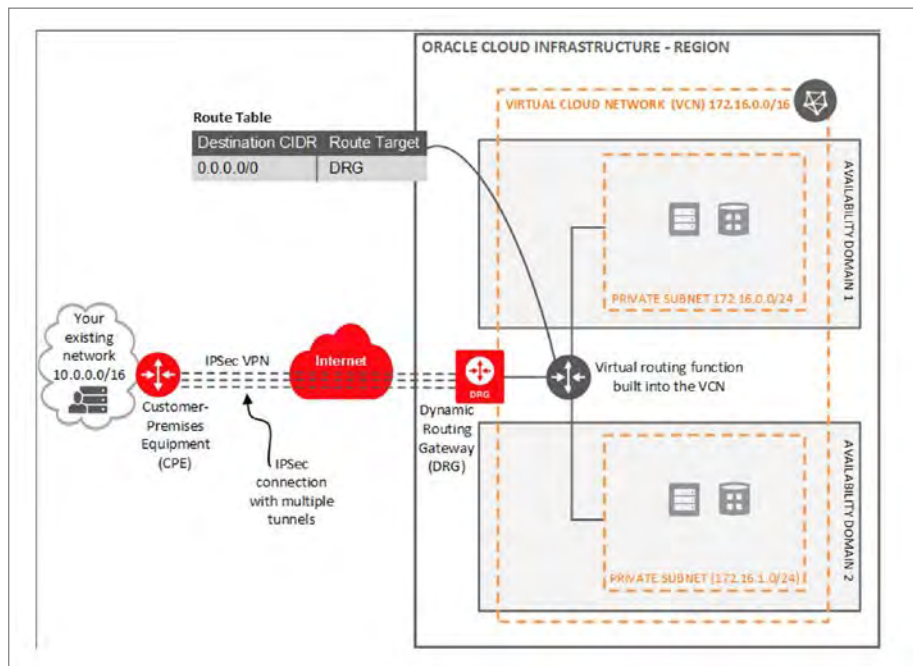


Abbildung 3: IPSec-VPN-Verbindung (Quelle: Oracle)

**Connectivity-Empfehlung #2:
Vermeiden Sie Single-Point-of-Failure.**

Für die Implementierung einer hochverfügbaren, ausfallsicheren Anbindung stellt Oracle eine mehrstufige Redundanz zur Verfügung:

- Data Center Redundancy: mehrere Oracle-FastConnect-Lokationen in jeder Region
- Provider Redundancy: mehrere Provider in jeder Oracle-FastConnect-Lokation
- Circuit Redundancy: mehrere Physical Circuits in jeder Oracle-FastConnect-Lokation

Oracle unterstützt zudem das Link Aggregation Control Protocol (LACP). Damit können mehrere Ethernet-Interfaces zu einer logischen Point-to-Point-Verbindung gebündelt werden. Folgende Tipps helfen bei der Konzeption einer ausfallsicheren Cloud-Anbindung:

- Erstellen Sie eine redundante Verbindung in die Oracle Cloud
- Verwendung von redundanter physischer Hardware auf Kundenseite
- Verwendung eines zweiten Netzwerkproviders für die Abdeckung der letzten Meile

- Planen Sie genügend Netzwerkkapazität für die FastConnect Virtual Circuits ein (auch bei Ausfall eines redundanten Circuit)
- Sorgen Sie für Service-Level-Redundanz durch einen IPSec-VPN-Service zusätzlich zu FastConnect. Oracle priorisiert FastConnect- gegenüber VPN-Verbindungen.

Für welchen Einsatzzweck eignet sich nun IPSec VPN respektive FastConnect? Grundsätzlich sind beide Methoden sicher. Ist der erforderliche Datendurchsatz jedoch konstant hoch (> 30...50 MB/sec) und die Anbindung geschäftskritisch, ist eine dedizierte Netzwerkverbindung mit garantierter Bandbreite zu bevorzugen. Für höhere Anforderungen ist eine FastConnect-Verbindung zu prüfen. Viele Kunden beginnen mit einem VPN-Setup und wechseln mit steigenden Anforderungen auf FastConnect. In Tabelle 1 sind die Eigenschaften der beiden Methoden nochmals einander gegenübergestellt.

Hybride Cloud-Anbindung

Aufgrund der unterschiedlichen Merkmale und Benutzer der verschiedenen Cloud-Service-Modelle (IaaS, PaaS, SaaS) entstehen auch unterschiedliche Anforderungen an die Konnektivität – wobei

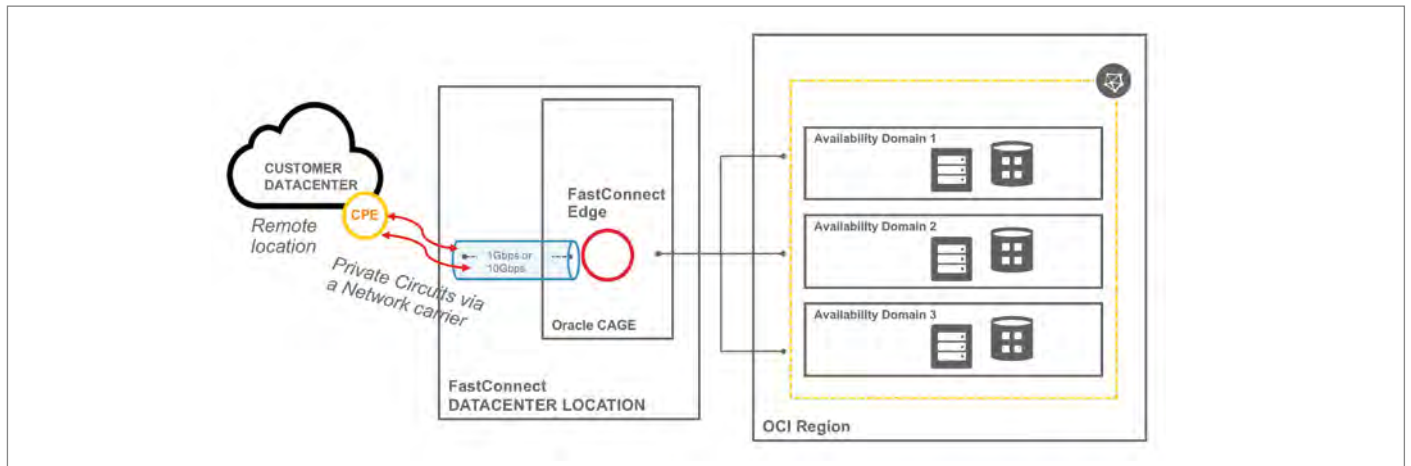


Abbildung 4: Oracle FastConnect

sich IaaS und PaaS von SaaS unterscheiden. Für IaaS und PaaS sind eine garantierte Bandbreite, stabiler Durchsatz und Verfügbarkeit wichtig, weil mit der Nutzung dieser Dienste typischerweise größere Datenmengen transferiert werden. Dies trifft besonders dann zu, wenn ein Cloud-Dienst, wie DBaaS, als Erweiterung des Rechenzentrums genutzt wird.

Für SaaS-Anwendungen stehen geringe Latenz, geringer Packet-Verlust und geringe Jitter-Rate (Unregelmäßigkeiten in der Übertragung von Netzwerkpaketen) im Vordergrund. Jitter betrifft vor allem Voice-Anwendungen, während Schwankungen in der Latenz die Qualität und das End-User-Erlebnis von SaaS generell beeinträchtigen können. Die Verbindung zu einem SaaS-Dienst sollte – unabhängig davon, von wo aus der Service genutzt wird – einfach, sicher und zuverlässig sein.

In *Abbildung 5* ist die Anbindung von IaaS- und PaaS-Diensten via FastConnect über einen Cloud-Exchanger sowie via

Internet zu den SaaS-Diensten beispielhaft dargestellt. Der Zugang zu den (kritischen) IaaS-/PaaS-Cloud-Diensten wird dabei zentral verwaltet und kann nur vom Unternehmensnetzwerk aus erfolgen. Der Zugang zu den SaaS-Diensten hingegen kann von überall her erfolgen – vorausgesetzt, die notwendige Authentisierung und Autorisierung wurde geprüft.

**Connectivity-Empfehlung #3:
Erwägen Sie unterschiedliche Anbindungen für SaaS und IaaS/PaaS**

Cloud Exchanger

Eine zentrale Rolle bei der Anbindung von Cloud-Services an das private Data-Center via FastConnect spielen die sogenannten Cloud Exchanger. Cloud Exchanger sind typischerweise Co-Location-Anbieter, die neben der klassischen RZ-Infrastruktur

(Breitband-Netzanschluss, Rack-Space, Strom, Kühlung und physische Sicherheit) auch einen Crossconnect zu den FastConnect-Edge-Geräten von Oracle zur Verfügung stellen. Die Liste der pro Region verfügbaren Oracle-FastConnect-Partner ist unter https://cloud.oracle.com/de_DE/fast-connect publiziert.

Cloud Exchanger bieten neben Oracle FastConnect noch weitere Cloud Exchange Services wie AWS Direkt Connect oder MS Express Route an. Darüber lässt sich die Oracle Cloud mit anderen Clouds verbinden.

**Connectivity-Empfehlung #4:
Ziehen Sie Netzwerk Service Provider und Cloud Exchanger in Betracht, die flexibel auf Bandbreitenanforderungen reagieren können.**

Freie Fahrt hat ihren Preis: Natürlich spielen auch die Kosten bei der Wahl der

	IPSec VPN	FastConnect
Use Case	Entwicklung/Test und kleinere Produktions-Workloads	Mission Critical Workloads, Oracle Apps, Backup, DR
Bandbreite ¹	Typischerweise < 250 Mbps	Höherer Bedarf; Inkremente von 1 Gbps und 10 Gbps Ports
Protokolle	IPsec	BGP (Border Gateway Protocol)
Routing	Static Routing	Dynamic Routing
Connection Resiliency	active-active	active-active
Verschlüsselung	Ja (default-artig)	Nein * (mit virtueller FW realisierbar)
SLA	Kein SLA	99.9% Availability SLA

Tabelle 1: Vergleich von IPSec VPN und Oracle FastConnect (¹ Bandbreite (Mbps) – Die maximale Bandbreite in Megabit pro Sekunde. Durch 8 teilen, um den erwarteten Durchsatz in Megabyte pro Sekunde zu erhalten)

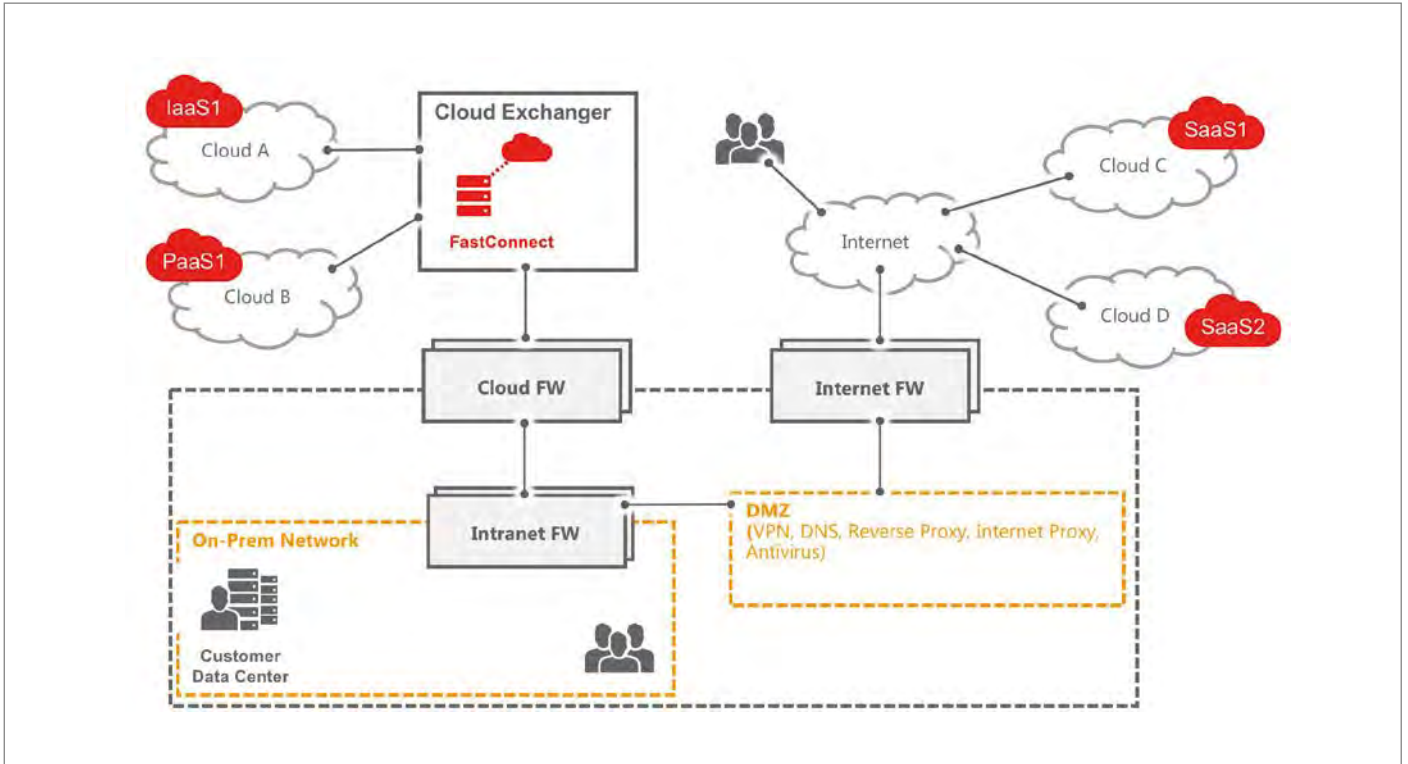


Abbildung 5: Hybride Cloud-Anbindung

Konnektivitätslösung eine Rolle. Neben den Initialkosten sind vor allem die wiederkehrenden Port-Kosten zu beachten. Die OCI-FastConnect-Port-Preise können über den Oracle Cost-Estimator (siehe <https://cloud.oracle.com/cost-estimator>) berechnet werden. Zusätzlich kommen die Kosten für einen Cloud Access Port bei einem Cloud Exchange Provider dazu sowie gegebenenfalls die Kosten für die letzte Meile.

Fazit

- Cloud Connectivity ist ein strategischer Baustein in einem Hybrid- und Multi-Cloud-Modell.
- Die Connectivity-Lösung muss auf das Service-Modell – IaaS, PaaS oder SaaS – abgestimmt sein.
- Das Internet erfüllt Enterprise-Anforderungen für eine stabile, zuverlässige und sichere Cloud-Anbindung nur bedingt. Mit IPsec VPN und FastConnect stellt Oracle zwei sichere und performante Konnektivitätsservices zur Verfügung
- Cloud Exchanger dienen als praktischer Multi-Cloud Connectivity-Hub für eine leistungsfähige und zuverlässige Cloud-Anbindung.

Referenzen

- OCI User Guide; Chapter 16 (Networking); https://docs.cloud.oracle.com/iaas/pdf/ug/OCI_User_Guide.pdf
- Oracle FastConnect: <https://cloud.oracle.com/fastconnect>
- Virtual Cloud Network Overview and Deployment Guide (Status: Nov. 2018); <https://cloud.oracle.com/iaas/whitepapers/vcn-deployment-guide.pdf>
- Graphics for Topologies and Diagrams: <https://docs.cloud.oracle.com/iaas/Content/General/Reference/graphicsfordiagrams.htm>

Weiterführende Literatur

- Als gut verständliche Einführung in die Netzwerktechnologie ist das Buch „Computer Networking Problems and Solutions“ von Ethan Banks und Russ White, erschienen im Addison Wesley Verlag, Dezember 2017, zu empfehlen
- Cloud Computing – speziell auch die Konnektivitätsaspekte – werden im Buch „Cloud Computing: Concepts, Technologie & Architecture“ von Zaigham Mahmood, Thomas Erl und Riccardo Puttini, erschienen im Prentice Hall Verlag, Mai 2013, beschrieben.



Daniel Steiger
daniel.steiger@trivadis.com



Autonomous Data Warehouse im Vergleich mit Konkurrenten

Dr.-Ing. Holger Friedrich, sumIT AG

Oracles Autonomous Data Warehouse Cloud (ADW) ermöglicht es vielen Industriezweigen, leistungsfähige DWH-Installationen mit geringem organisatorischen Aufwand zu betreiben. Die Skalierbarkeit, Geschwindigkeit der Provisionierung und Leistungsfähigkeit dieses Cloud Service auf Basis von Exadata-Hardware ist beeindruckend. Allerdings steht Oracle als gemanagter Data-Warehouse-Cloud-Service nicht konkurrenzlos da, war sogar später damit am Markt als Mitbewerber. Es ist daher lohnenswert, ADW nicht nur einzeln anzuschauen, sondern auch im Vergleich mit der Nicht-Oracle-Konkurrenz, beispielsweise Google Big Query, Amazon Red Shift und Snowflake. Manche Produkteigenschaften von ADW sind denen der anderen DWH-Cloud-Anbieter überlegen, es gibt aber auch Features, die die Konkurrenz bietet, ADW bislang jedoch nicht aufweist.

Rank			DBMS	Database Model	Score		
Feb 2019	Jan 2019	Feb 2018			Feb 2019	Jan 2019	Feb 2018
1.	1.	1.	Oracle +	Relational DBMS	1264.02	-4.82	-39.26
17.	↑ 18.	↑ 20.	Amazon Redshift +	Relational DBMS	20.99	+0.94	+7.87
18.	↑ 19.	18.	Netezza	Relational DBMS	19.77	+0.12	+3.19
20.	20.	↑ 21.	Google BigQuery +	Relational DBMS	18.75	+0.38	+6.40
26.	26.	↑ 30.	Microsoft Azure SQL Data Warehouse	Relational DBMS	8.47	+0.31	+3.68
30.	↑ 31.	↑ 35.	Amazon Aurora	Relational DBMS	5.89	+0.23	+1.95
41.	41.	↑ 65.	Presto	Relational DBMS	3.47	+0.41	+2.53
53.	↓ 52.	↑ 76.	Snowflake +	Relational DBMS	2.25	+0.09	+1.53
59.	59.	↓ 50.	IBM Db2 warehouse	Relational DBMS	1.65	-0.01	+0.00

Abbildung 1: DWH Ranking nach DB-Engines Website (Stand Februar 2019)

Gemanagte Cloud-DWH-Dienste

Es tummeln sich seit vielen Jahren eine ganze Reihe von Produkten im Data-Warehouse-Bereich, sowohl Spezial-Implementierungen, wie etwa Netezza oder Teradata, als auch All-Purpose-Datenbanken wie Microsoft SQL Server und Oracle RDBMS. Mit der Einführung von Cloud-Lösungen ist der DWH-Bereich noch vielfältiger und damit auch unübersichtlicher geworden. Neben den etablierten Anbietern, die ihre Lösungen zunächst als Software-as-a-Service-Angebote, z. B. Oracles DbaaS, anboten, kamen eine ganze Reihe neuer Produkte und Dienste, sowohl von etablierten als auch von neuen Anbietern hinzu.

Herausragend und hierbei von besonderem Interesse sind dabei gemanagte DWH-Dienste, solche also, die bei einem oder mehreren der internationalen Cloud-Anbieter zur Verfügung stehen und sich dadurch auszeichnen, dass sie keine grundlegenden administrativen Tätigkeiten von Kundenseite erfordern. Alle üblichen Basis-DBA-Themen wie Backup/Restore, Ausfallsicherheit, Skalierung von Speicher und Rechenleistung werden entweder automatisch vom Dienst nach Bedarf durchgeführt oder können vom Kunden per Knopfdruck auf einer Web-Oberfläche ausgelöst werden. Diese Dienste sind es, die Kunden echten Mehrwert im Vergleich zu On-Premises-Installationen oder klassischem Outsourcing bieten.

Abbildung 1 zeigt das aktuelle Ranking der Hauptanbieter auf der DB-Engines Website [1]. Diese schätzt die Popularität von Engines aufgrund von Stellenangeboten, Social Media Postings und anderen journalistischen Veröffentlichungen ein.

Neue Wettbewerber wie Snowflake sind dabei, an Popularität zu gewinnen, sind bislang allerdings noch deutlich davon entfernt, die Reichweite der Oracle-Plattform zu erreichen. Es stellt sich die Frage, ob ein derartiger Vergleich fair ist, da unter Oracle viel mehr als nur der Autonomous Data Warehouse Service zusammengefasst ist. Grundsätzlich ist dies korrekt. Da aber, wie wir im weiteren Verlauf diskutieren werden, Oracle ADW de facto eine Standard-Oracle-Datenbank als Basis nutzt, ist der Vergleich dennoch sinnvoll und insbesondere bezüglich der Kundenanforderungen relevant.

Kundenanforderungen

Jeder Kunde hat leicht unterschiedliche Anforderungen an seine DWH-Umgebung. Eine Auswahl der folgenden Punkte ist jedoch für fast alle modernen DWH-Implementierungen von Belang.

- Agilität – schnelle Provisionierung, De-Provisionierung und Deployment
- Verfügbarkeit – 24/7-Hochverfügbarkeit und Disaster-Recovery-Funktion
- Skalierbarkeit – für Geschäftswachstum sowie tägliche und saisonale Spitzen
- Zuverlässigkeit – gleichbleibende, garantierte Ressourcen und Performance
- Einfachheit – simple Bedienung, Connectivity etc.
- Kontrolle – Überwachungs- und Eingriffsmöglichkeiten für Entwickler
- Geringer Wartungsaufwand
- Expertise am Markt – Expertenpool auf dem Arbeitsmarkt verfügbar
- Performanz – berechenbare und gleichbleibende Performanz des Service

Wettbewerbssituation

Die mittlerweile am Cloud-Markt konkurrierenden DWH-Services können grob in drei Kategorien unterschieden werden.

1. Bekannte Mitbewerber mit „cloudifizierten“ Angeboten. Hier sind zum Beispiel das entsprechende Microsoft-Angebot mit SQL-Server in der Azure Cloud oder IBM Db2 Warehouse on Cloud zu nennen. Diese Angebote sind für Kunden direkt mit Oracle ADW vergleichbar, da vor allem Automatisierung und das Cloud Interface den Unterschied zu den bekannten On-Premises-Versionen der klassischen RDBMS-Produkte ausmachen.
2. Versionen von Open-Source-RDBMS, die zu Cloud-ready- und dabei proprietären Lösungen weiterentwickelt wurden. Ein Beispiel hierfür ist Amazon Redshift, bei dem Amazon eine PostgreSQL-Version genommen, erweitert und derart adaptiert hat, dass sie nun als gemanagtes Cloud-DWH angeboten werden kann. Die Grundlage ist ein Open-Source-Produkt, das Ergebnis ist ein proprietärer Cloud-DWH-Service.
3. Neue, innovative Implementierungen. Prominente Vertreter hierfür sind Google BigQuery und Snowflake, zwei Angebote, die neue Ideen und Ansätze zum Cloud-DWH-Markt bringen. Dies macht diese Dienste einerseits schwieriger vergleichbar mit traditionellen RDBMS-basierten Lösungen, aber auch gleichzeitig interessant.

Egal zu welcher Gruppe sie gehören, Cloud-DWH-Dienste sind de facto proprietäre Lösungen. Selbst wenn sie von

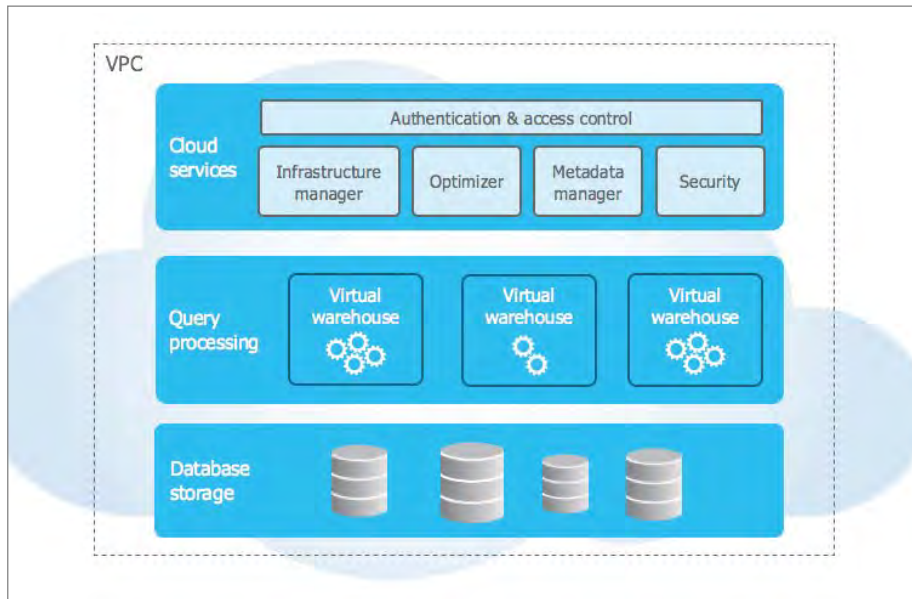


Abbildung 2: Snowflake-Architektur

Open-Source-Projekten abgeleitet sind, sind zumindest die Automatisierung, das Cloud-Interface und andere Bestandteile der Lösungen von den Cloud-Anbietern erstellt und ClosedSource. Für Kunden bedeutet dies also immer eine Locked-in-Situation. Diese ist nicht a-priori von Nachteil, sollte Neukunden jedoch bewusst sein.

Das Credo der Anbieter lautet, dass ein Cloud-DWH ein Dienst ist, weswegen sich Kunden keine Gedanken über Details und Ausführungseffizienz machen müssen

bzw. sollen. Sie bieten daher in der Regel zwar gute Überwachungsinstrumente, aber wenige Eingriffsmöglichkeiten in die Ausführung von Operationen. Der Kunde muss weitgehend darauf vertrauen, dass der Service die benötigten Ressourcen zur Laufzeit hat und diese effektiv und effizient einsetzt. Sollte die Performanz des Dienstes nicht oder nicht mehr ausreichen, bleiben dem Kunden kaum andere Eingriffsmöglichkeiten, als mehr Leistung mit zusätzlichen Kosten zu erkaufen.

Interessant sind aus Kundensicht ins-

besondere neue, Cloud-native Angebote, die versprechen, vieles besser, billiger und einfacher zu machen als die etablierten Anbieter. Als typische Vertreter dieser Art werfen wir im Folgenden einen Blick auf Snowflake und Google BigQuery

Snowflake

Snowflake ist eine vollständig neu entwickelte Cloud-only Data-Warehouse-Plattform. Die Gründer, zwei Veteranen der Oracle-Parallel-Server- und RAC-Entwicklung, haben ihre lange Erfahrung mit der Oracle-Codebasis genutzt und mit Wagniskapital eine neue Lösung gebaut, die ganz auf das Cloud-Geschäftsmodell ausgerichtet ist. Dabei verfügt Snowflake nicht über eigene Rechenzentren, sondern läuft als Gast bei großen Cloud-Anbietern. Zurzeit sind dies Amazons AWS und Microsoft Azure.

Die Architektur von Snowflake ist in *Abbildung 2* wiedergegeben. Datenbankinstanzen, die in VMs/Containern nach Bedarf gestartet werden, werden als Virtual Warehouses bezeichnet. Virtual Warehouses können mit unterschiedlichen Rechenressourcen ausgestattet sein und verhalten sich ähnlich wie die Instanzen in einem Oracle RAC Cluster, die gemeinsam auf einer Menge von Dateien arbeiten.

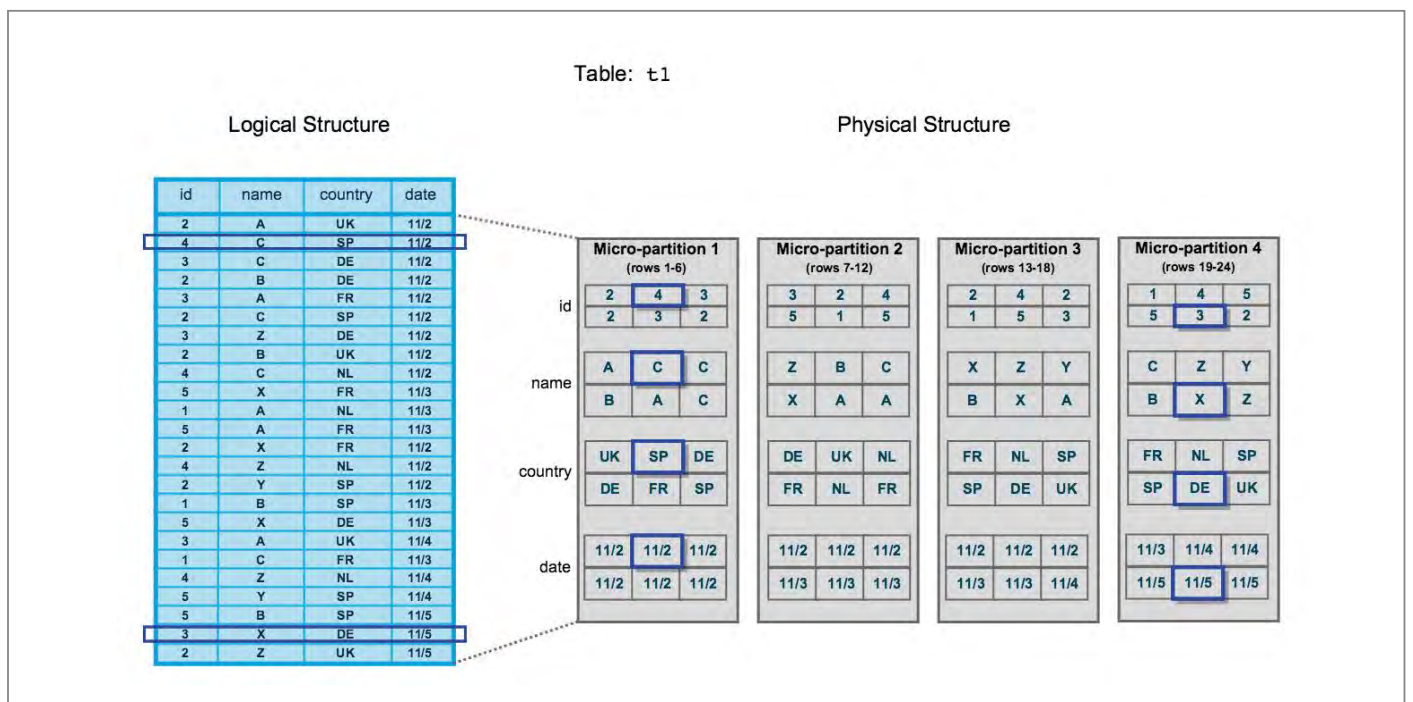


Abbildung 3: Micro-Partitionen in Snowflake

Im Gegensatz zu Oracle ADW speichert Snowflake die Datenbankdateien auf Write-Once-Objektspeicher in AWS' S3 und Azures Objekt-Storage. Updates in Datenbankdateien sind daher nicht möglich. Jedes Update oder Delete in einem Snowflake-DWH erfordert, dass eine Kopie von Daten angelegt werden muss. Daten werden in sogenannten Micro-Partitionen (MP) gespeichert, die den Compression Units in Oracles Hybrid Columnar Compression ähneln. Eine MP enthält zwischen 50MB und 500MB unkomprimierter Daten in Spaltenformat[2]. Wird ein Datenfeld in diesen Units manipuliert, wird die gesamte MP kopiert (siehe Abbildung 3).

Diese Form der Speicherung hat den Vorteil, dass kein UNDO Tablespace gebraucht wird. Flashback-Operationen sind einfach, da auf ‚alte‘ Kopien zurückgegriffen werden kann, solange diese vorgehalten werden. Dafür sind Update-, Delete- und Merge-Operationen aber auch relativ teuer und aufwendig.

Gesammelte Statistiken auf MPs erlauben, Pruning analog zu Oracles Exadata Storage Indexes durchzuführen und so I/O zu vermeiden. Reorganisation, also die Sortierung der Daten, gestattet es, die Effizienz von Abfragen bezüglich bestimmter Attribute zu verbessern. Jede Reorganisation erfordert jedoch das erneute Schreiben aller MPs einer Tabelle. Weitere Partitionierungsmechanismen sind derzeit (noch) nicht implementiert.

Komfortable Monitoring-Funktionalität, vergleichbar der des ADW, ist verfügbar, es gibt aber praktisch keine Eingriffsmöglichkeiten zur Steuerung der Abfrageausführung, analog zu Oracle Hints oder ähnlichem.

Google BigQuery

Einer der ersten Anbieter, der einen automatisierten DWH-Dienst auf Write-once-Speicher angeboten hat, ist Google. Mit dem BigQuery-Service verfolgt Google offensichtlich die bekannte Strategie, Technologien, die zunächst für interne Zwecke entwickelt wurden, zu erweitern, der Öffentlichkeit zugänglich zu machen und auf diese Weise zu monetarisieren. Der Dienst ist ausschließlich in der Google Cloud verfügbar.

Abbildung 4 zeigt die Infrastrukturkomponenten des BigQuery-Service. Man

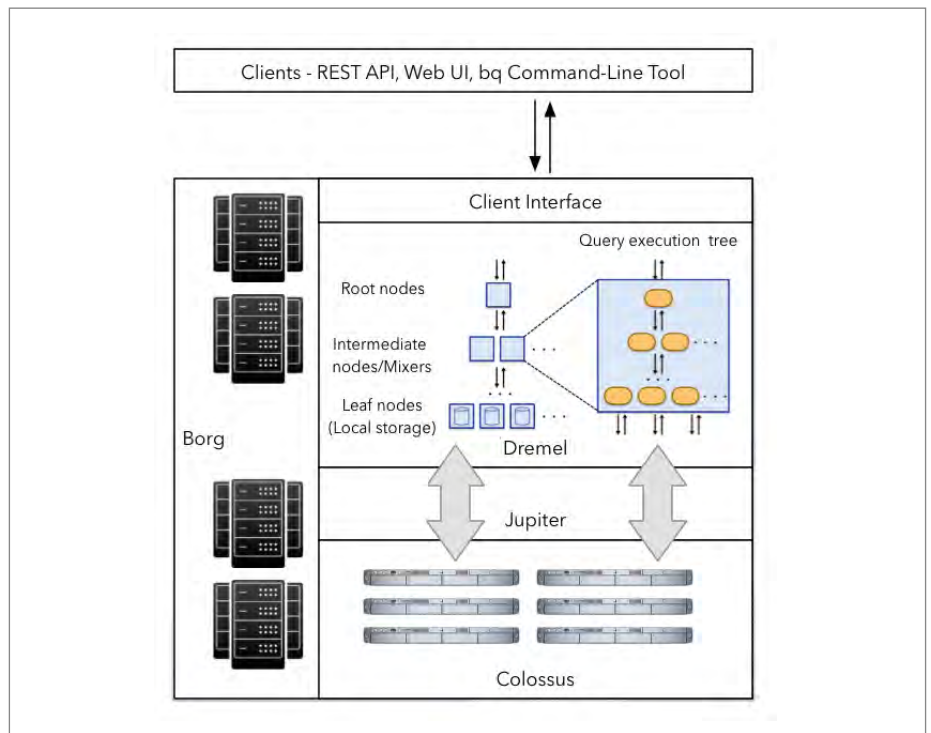


Abbildung 4: Architektur von Google BigQuery

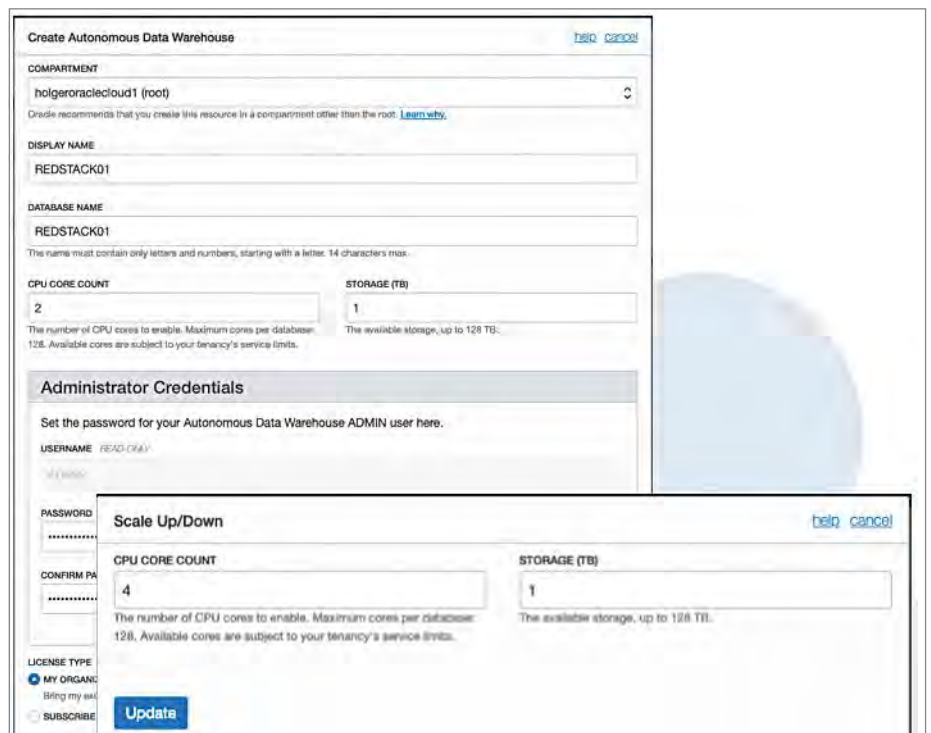


Abbildung 5: Provisionierung und Skalierung des ADW per OCI

sieht, dass die Standard-Google-Cloud-Komponenten Borg, Jupiter und Colossus für Compute, Netzwerk und Speicherung verwendet werden. Darauf arbeitet die Dremel-Engine, die den Dienst implementiert und koordiniert.

Da BigQuery seine Daten, analog zu Snowflake, auf einmal beschreibbarem

Speicher verwaltet, sind auch in diesem Dienst Daten-verändernde und -löschende Operationen sehr aufwendig (s.o.). Konsequenterweise erlaubt Google je Tabelle nur eine bestimmte Anzahl dieser Operationen pro Tag. Weitere Grenzen bezüglich gleichzeitiger Abfragen etc. gibt es ebenfalls[3]. Für Real-Time-DWHs,

List of Restricted and Removed Oracle Features	
The following Oracle Database features are not available in Autonomous Data Warehouse:	
• Oracle Real Application Testing	• Oracle Data Masking and Subsetting Pack
• Oracle Database Vault	• Oracle Cloud Management Pack for Oracle Database
• Oracle OLAP	• Oracle Application Express
• Oracle R capabilities of Oracle Advanced Analytics	• Oracle Multimedia
• Oracle Spatial and Graph	• Java in DB
• Oracle Industry Data Models	• Oracle XML DB
• Oracle Text	• Context
• Oracle Tuning Pack	• Oracle Workspace Manager
• Oracle Database Lifecycle Management Pack	

Abbildung 6: Oracle-RDBMS-Features, die nicht in ADW verfügbar sind (Stand 02/19)

bei denen kontinuierlich Daten nicht nur hinzugefügt, sondern unter Umständen auch verändert, zum Beispiel Versionen zeitlich abgeschlossen werden, kann dies ein Show-Stopper sein.

Weitere Beschränkungen durch Big-Queries haben das Potenzial, die Tauglichkeit des Dienstes für kritische DWH-Applikationen einzuschränken. Partitionierung der Datenstrukturen ist beispielsweise lediglich bezüglich Datumsfeldern, mit einer Ebene Tiefe und ausschließlich mit Tagesgranularität möglich. Auch gibt es keine weiteren Eingriffsmöglichkeiten in die Ablaufsteuerung oder die Zuweisung von Ressourcen. Der Dienst läuft de facto in einem Best-effort-Modus seitens Google.

Schließlich belebt Google das Konzept der Nested Tables wieder, um so durch Denormalisierung Performanceprobleme bei großen Joins zu vermeiden[4]. As-of-Now-Auswertungen mit SCD2-Dimensionen sind dann allerdings nicht mehr auf den gleichen Strukturen möglich.

Oracle Autonomous Data Warehouse

Im Gegensatz zu den zuvor beschriebenen Diensten setzt Oracle nicht auf eine Neuentwicklung. Der Dienst basiert auf der Codebasis von Oracles Standard-RDBMS und Exadata-Infrastruktur. ADW ist daher, und das ist einer seiner großen Vorteile, eine gewöhnliche, aktuelle Oracle-Datenbank, die lediglich um wenige Eingriffsmöglichkeiten und Optionen reduziert wurde. Ergänzt wurde sie dafür mit einem vollständigen Cloud-Automa-

tisierungsframework und webbasierten Monitoring-Werkzeugen sowie PL/SQL-Paketen zum direkten Zugriff auf eine Reihe von Cloud-Storage-Diensten. Provisionierung neuer und Skalierung bestehender ADW-Instanzen erfolgen problemlos per Oracle Cloud Interface (OCI). Skalierung geschieht dabei ohne jegliche Unterbrechung des Dienstes (siehe Abbildung 5). Detaillierte Real-Time-Überwachung des Dienstes sowie einzelner Queries ist möglich. Hierzu bietet ADW ein Web-GUI in Art eines Enterprise-Manager-Express light.

Im Gegensatz zu anderen Cloud-DWH-Diensten gibt ADW den Usern eine wesentlich größere Kontrolle über die Ausführung. Alle relevanten Hints lassen sich einschalten und nutzen. Erfahrene Oracle-Entwickler können so per Monitoring die Ausführung überwachen und aufgrund der Ergebnisse ihre Strukturen und Abfragen wie bei On-Premises gewohnt

tunen, sollte dies notwendig sein. Zudem ist das vollständige Set von Indizierung und Partitionierung der Standard-Oracle-Datenbank verfügbar. Application Tuning ist also nicht per se notwendig, aber möglich, um die beste Leistung aus der gemieteten Rechenleistung zu bekommen.

Nicht alle Datenbankoptionen sind bislang verfügbar (siehe Abbildung 6). Im Laufe der Zeit wurden jedoch mehrere freigeschaltet. In naher Zukunft soll auch APEX in ADW verfügbar werden.

Ein Alleinstellungsmerkmal von ADW ist, so seltsam es für altgediente Datenbankentwickler sein mag, das Vorhandensein von Integritäts-Constraints. Hiermit sind insbesondere Primary-, Unique- und Foreign-Key-Constraints gemeint. Snowflake und Google BigQuery, aber auch Amazon Redshift und andere Cloud-DWH-Dienste lassen die Definition solcher Constraints entweder gar nicht oder höchstens als Dokumentation zu. ADW dagegen erlaubt es, den vollen Umfang von Constraint-Definitionen zu nutzen, und überprüft eingeschaltete Constraints auch wie gewohnt und erwartet.

Aus dieser Sicht ist ADW also Enterprise-DWH-ready. Eine Beschränkung gibt es allerdings zurzeit, die dem Einsatz für kritische Systeme noch entgegensteht. ADW gibt es bislang nicht mit Disaster-Recovery-Option, also Data-Guard. So ist ADW zwar hochverfügbar dank der RAC-Installation auf Exadata, aber nicht mit Stand-by-Datenbanken ausgestattet. Vollständige Ausfallsicherheit ist also bislang nicht gegeben. Dieses Feature hat Oracle angekündigt, allerdings noch nicht ausgeliefert.




Kriterium			
Pricing	Speicher & CPU-Credits	Speicher & gescannte Daten	Speicher & CPU
Ressourcen	unklar	nicht garantiert	garantiert
Skalierung	angeblich ad infinitum	angeblich ad infinitum	128CPU/128TB online mehr über Sales
Partitionierung	ausschliesslich Micro-Partitions	nach Zeitattribut	voll flexibel Oracle 18c
Query-Execution-Optimierung	nein	manual join order	Hints
Monitoring	Workload & Query-Explain-Plan	Workload & Query-Execution-Steps	Workload & Real-Time SQL Monitor
Integrität	nein, informational Constraints	nein	ja
Migration	Neu-implementierung	Neu-implementierung	lift & shift

Abbildung 7: Charakteristika der betrachteten Cloud-DWH-Dienste

Vergleich

Sucht man Performance-Vergleiche zwischen Cloud-DWH-Diensten im Internet, findet man nur wenige Informationen. Nachvollziehbare Daten, analog zu den alten TPC-Benchmarks, gibt es nicht. Vergleiche von Performance im Verhältnis zu Kosten sind noch schwieriger zu ziehen, da die Preismetriken der Dienste sehr unterschiedlich sind. Der in *Abbildung 7* dargestellte Vergleich beschränkt sich daher auf die technischen Eigenschaften der Dienste.

Die Frage, welcher Dienst zu bevorzugen wäre, hängt maßgeblich vom genauen Anwendungsfall ab. Experimentelle Implementierungen und vollständige Cloud-Implementierungen, die nie mehr in eigenen Rechenzentren laufen müssen, können mit Snowflake oder Google BigQuery durchgeführt werden. Der Kunde muss sich hierbei aber bewusst sein, dass er keine garantierten, dedizierten Ressourcen erwirbt und es auch – insbesondere in Europa – (noch) keine große Menge qualifizierter externer Berater für diese Technologien gibt.

Für IOT-Workloads und die Auswertung sehr großer (denormalisierter) Datenmengen erscheint Google BigQuery bestens geeignet, da dessen Scan-Performance bei großen Datenmengen sehr überzeugend ist.

Bei einer Reihe von Szenarien ist Oracles ADW den anderen Lösungen jedoch überlegen oder sogar die einzige mögliche Lösung, die die Anforderungen abdecken kann. Solche Szenarien sind die folgenden:

- ein bestehendes Oracle-DWH in die Cloud verlegen
- viele Datenmanipulationen innerhalb des DWH durchführen
- ein Cloud-DWH wieder ins eigene RZ zurückholen können
- qualifizierte externe Unterstützung in größerem Umfang nutzen
- garantierte, dedizierte Compute-Ressourcen haben
- Ausführungsstrategien beeinflussen können
- technisch erzwungene Integritäts-Constraints nutzen

Fazit

Die Neueinsteiger in den Cloud-DWH-Markt zeigen interessante, innovative Konzepte und sind bezüglich verschiedener Kriterien sehr überzeugend. Oracle Autonomous Data Warehouse hat also leistungsfähige Konkurrenz am Markt. Die Wettbewerber weisen aber jeweils an bestimmten Stellen Schwächen oder zumindest Einschränkungen auf und sind

aus Kundensicht immer ein Lock-in, nicht nur auf den Anbieter, sondern auch auf die Cloud.

Oracle bietet dagegen ein vollständiges Paket, das dem Kunden vollen Cloud-Komfort auf einer vertrauten, leistungsfähigen Plattform gibt, ihm jedoch gleichzeitig weitgehende Kontrolle und Eingriffsmöglichkeiten lässt, inklusive der einfachen Migration in eine andere Umgebung, sollte dies je gewünscht werden.

Referenzen

- [1] <https://db-engines.com/en/ranking/relational+dbms>
- [2] <https://docs.snowflake.net/manuals/user-guide/tables-clustering-micropartitions.html>
- [3] <https://cloud.google.com/bigquery/quotas>
- [4] <https://cloud.google.com/bigquery/docs/nested-repeated>



Dr.-Ing. Dipl.-Inform. Holger Friedrich
Holger.Friedrich@sumit.ch

DOAG Forms Day 2019 feiert 40 Jahre Oracle Forms

München - am 21. Februar fand in der Oracle-Niederlassung München zum dritten Mal mit 54 Teilnehmern der Forms Day der DOAG statt. Diesmal galt es, ein besonderes Jubiläum zu feiern. Die Ursprünge der Technologie Oracle Forms reichen bis in das Jahr 1979 zurück. Doch Oracle Forms behauptet sich weiterhin als leistungsfähige Technologie für datenbank-getriebene Anwendungen in zahlreichen Unternehmen. Da jedoch viele Forms-Anwendungen in die Jahre gekommen sind, besteht ein dringender Bedarf an Modernisierung, sowohl was das Look

& Feel als auch die Funktionalität betrifft. Die Vorträge des Forms Days (*siehe unter dem Punkt Unterlagen unter „[- neue Betriebskonzepte für Forms-Applikationen in der Cloud und als Container-Lösung
- Verbesserungen im Look & Feel bestehender Forms-Applikationen
- neue Möglichkeiten der Integration von Forms-Applikationen mit Web-Anwendungen auf dem Desktop](https://www.doag.org/de/eventdetails?tx_doagevents_single[id]=572387““</i>) beleuchteten deshalb:</p></div><div data-bbox=)*

Dr. Jürgen Menge, Der IT Macher GmbH

- die Integration von Services in Forms-Applikationen

Der Höhepunkt der Veranstaltung war zweifellos die Videokonferenz mit Bill Friend und Michael Ferrante in den USA (*siehe „[Zum Abschluss der Veranstaltung forderte Frank Hoffmann die Teilnehmer auf, eine Petition zur tiefgreifenden Modernisierung von Oracle Forms zu unterstützen.](http://stream.doag.org/img/filme/2019_02_21_FormsDay_2019_Videokonferenz_Bill_Mike_Gerd_CUT.mp4““</i>).</p></div><div data-bbox=)*



Spurensuche in Microservices

Matthias Fuchs, DevTops GmbH, Andreas Chatziantoniou, DevTops GmbH

Microservices ist ein Architekturmuster, bei dem die Anwendungssoftware in verschiedene Services, meist als Microservice bezeichnet, zerlegt wird. Diese Art der Umsetzung ist beliebt bei der Softwareentwicklung in der Public Cloud oder in Ansätzen mit Containern. Zusätzlich eignen sich die erzeugten Softwareartefakte sehr gut als Basis für eine agile Entwicklung nach z.B. Scrum, da unabhängig einzelne Komponenten erstellt werden können. Ein zentraler Aspekt aus Sicht der Infrastruktur ist, dass die Microservices beliebig oft auf beliebigen Servern existieren. Dabei haben diese dedizierte Aufgaben wie Zugriffe auf Datenbanken, Daten weiterzuverarbeiten, zu formatieren oder zu analysieren.

Es lässt sich nicht sagen, welche Prozesse auf welchem Server für eine spezielle User-Anfrage verwendet werden. Zusätzlich gibt es keinen Status eines Service, d.h. der Zustand eines Prozesses wird nicht gespeichert. Wenn ein Prozess in einem Service abbricht, kann nicht wieder an dieser Stelle aufgesetzt werden, da der nächste oder wiederholte Aufruf vielleicht auf einem anderen Server stattfindet. Eine Suche nach gezielter Information für genau einen User Context in einem Service erfordert zusätzlich Maßnahmen, die von Anfang an bedacht werden müssen.

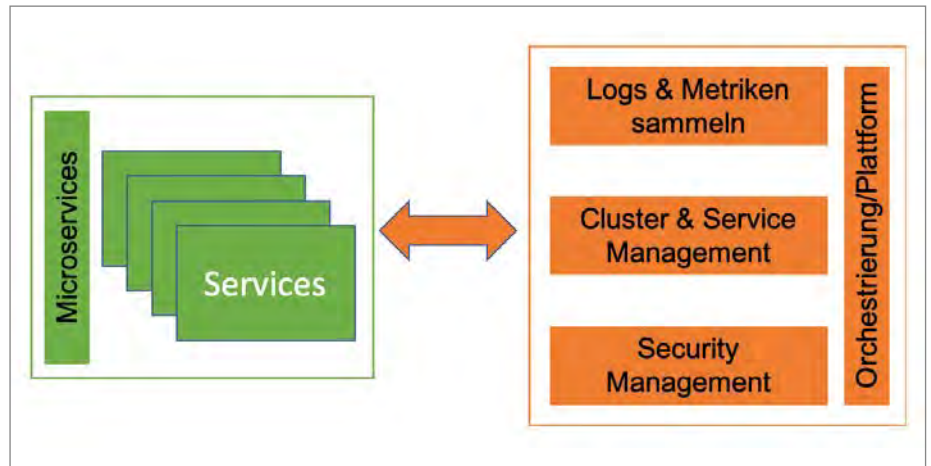


Abbildung 1: Aufgaben bei der Orchestrierung

Verteilte Systeme

Eine Applikation, die auf mehreren Microservices aufbaut, kann man als verteiltes System betrachten. Eine Microservice-Applikation, die nur aus wenigen oder einem Service beziehungsweise Programm besteht, entspricht nicht dem Gedanken einer verteilten Architektur. Durch die Verwendung von mehreren kleinen Services, die eine bestimmte Aufgabe bearbeiten, ergeben sich viele Möglichkeiten im Bereich Entwicklung und Betrieb. Ein Nachteil ist, dass eine Analyse dessen, was in welchem Prozess passiert, aufwendiger wird, da nicht bekannt ist, auf welchem Server oder in welchem Service der gerade ausgeführte Prozess des Users stattfindet.

Orchestration

Um aus Sicht des Betriebes Microservices zu verwalten, werden Orchestrierungs-

systeme eingesetzt. Dies kann sowohl ein Framework wie Kubernetes oder Cloud Foundry sein als auch Erweiterungen in Cloud-Umgebungen wie Container Services (Oracle) oder Elastic Cloud Services (AWS). Ziel ist es, die Microservices zu überwachen und gegebenenfalls gestoppte Services wieder zu starten oder bei hoher Last zusätzliche Services zu initialisieren. Als Nebeneffekt einer Orchestrierung ergeben sich ein zentrales Monitoring über Performance-Daten und meist ein Einsammeln aller Applikationslogs an einer zentralen Stelle (siehe Abbildung 1).

Correlation ID

Im Rahmen des Betriebs und des Supports wird es immer wieder Situationen geben, in denen gezielt einzelne Abläufe nachverfolgt werden müssen. Zusätzlich muss es auch über eine Vielzahl von Services möglich sein, Fehler zu analysie-

ren. Es ist nicht möglich, sich jeden Service einzeln anzuschauen, da eventuell Hunderte Services in Betracht kommen. Somit muss man gezielt Daten sammeln, die eine Analyse ermöglichen. Eine zentrale Idee ist es, jedem Programmablauf, User-Aufruf oder Prozess eine eindeutige Kennzeichnung mitzugeben. Eine Möglichkeit ist es, eine Correlation ID zu erzeugen, die über alle beteiligten Services im Rahmen eines Prozesses ausgetauscht wird. Diese ID muss immer eindeutig über die gesamte Plattform für jeden Aufruf sein. Die ID muss beim ersten Kontakt mit dem System generiert werden. Ist bereits eine ID vorhanden, darf keine neue ID generiert werden. Alle Informationen, die im Rahmen des Ablaufs entstehen und die für eine Analyse notwendig sind, sollten immer mit dieser ID gekennzeichnet werden. Die Speicherung muss zentral erfolgen, um Auswertungen über alle Services zu ermöglichen. Das Aggregieren der Logfiles

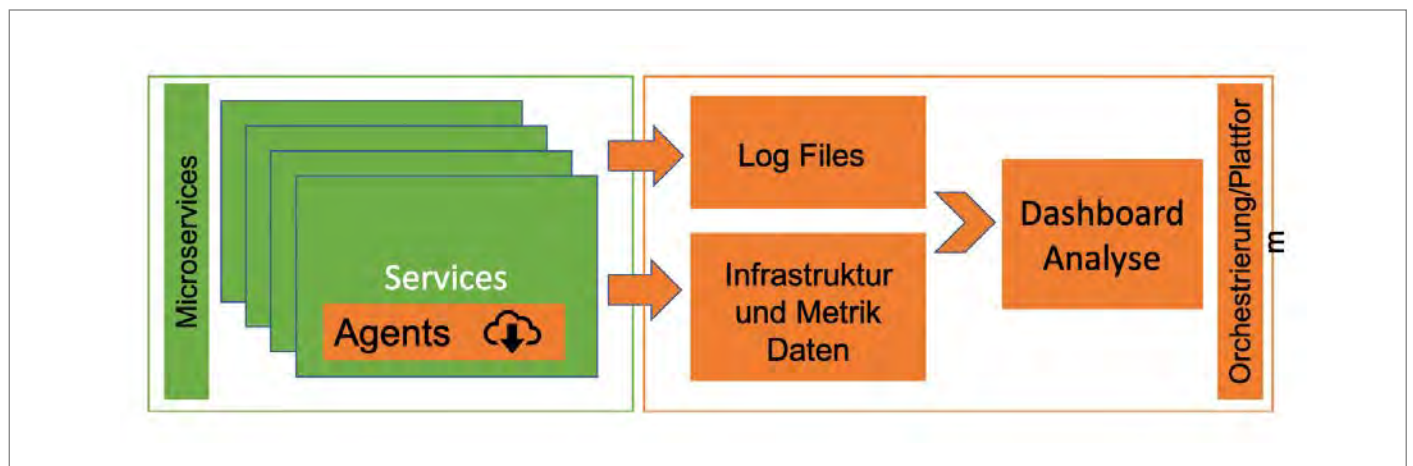


Abbildung 2: Logging und Metriken mit Orchestrierung

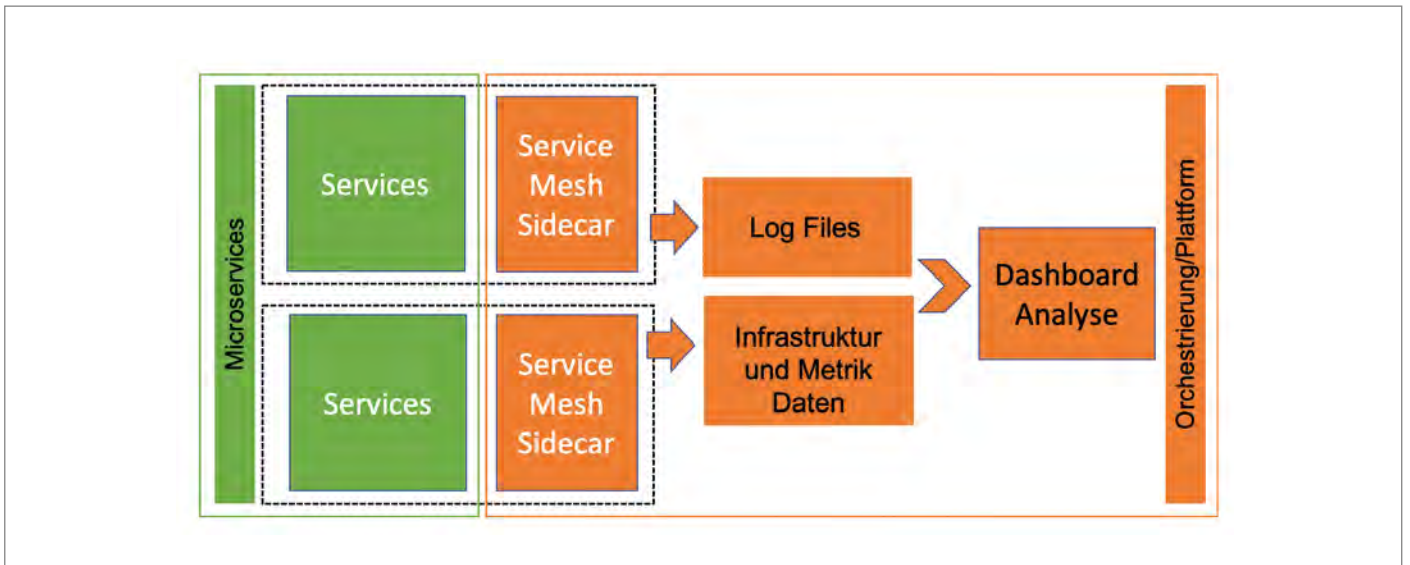


Abbildung 3: Sidecar mit Orchestrierung

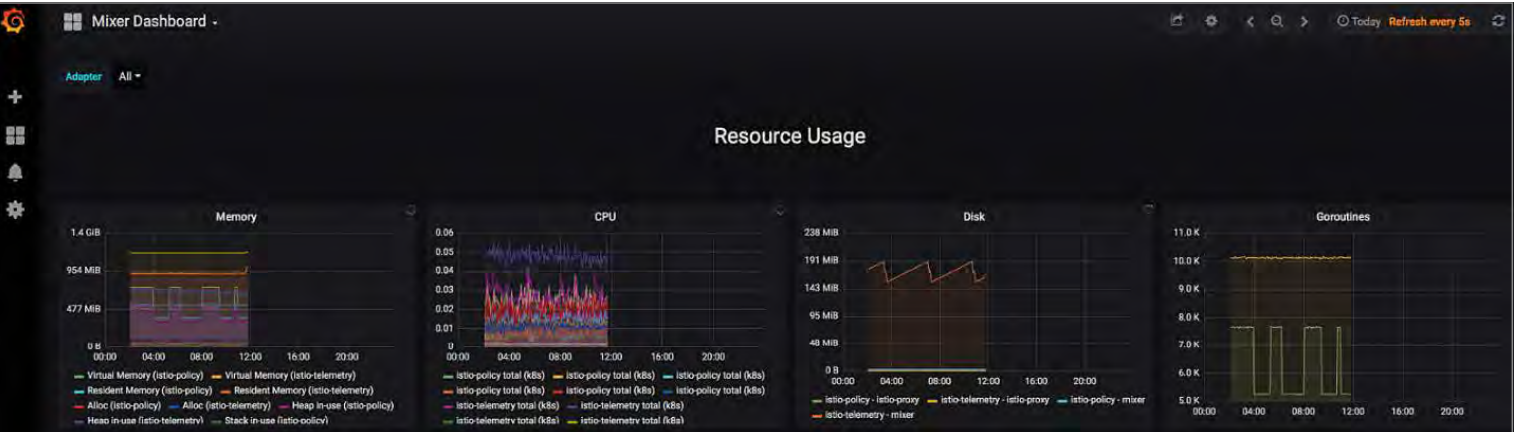


Abbildung 4: Grafana Analyse-Dashboard

der einzelnen Services wird von Cloud- oder Orchestrierungsservices übernommen. Es ist aber auch möglich, dass ein Log Service aus der Applikation heraus angesprochen wird, die die Daten verarbeitet.

Zentrales Logging

Die Log-Daten einer Applikation werden mithilfe von Infrastruktur-Services (siehe Abbildung 2) im Rahmen der Orchestrierung gesammelt. Meist leitet das Orches-

trierungsframework die Log-Daten an einen Logging Service beziehungsweise eine Datenbank weiter. Zusätzliche Agenten auf den Server oder in den Containern können installiert werden, um spezielle Daten zu sammeln. Für das Auswerten der Log-Daten werden Tools wie Elasticsearch/Kibana oder Splunk eingesetzt. In Cloud-Umgebungen gibt es oft spezialisierte eigene Services (z.B. Management Cloud Oracle), die diese Aufgabe ebenso übernehmen können. Die Aufbereitung der Logs muss so erfolgen, dass ein einfaches und schnelles Suchen anhand der Correlation ID, der Zeit und weiterer frei konfigurierbarer Kennzahlen möglich ist.

Beginn der Suche

Nachdem alle Daten in einer zentralen DB abgelegt wurden und eine Analyse-

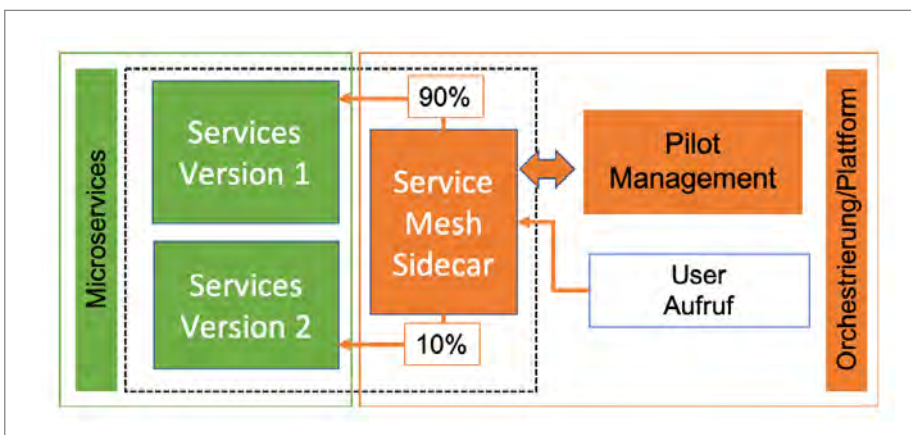


Abbildung 5: Canary Testing mit Sidecar

Software wie z.B. Kibana zur Auswertung zur Verfügung steht, kann die Suche beginnen. Über die Daten wird nach Fehlern oder Unregelmäßigkeiten gesucht. Wird ein passender Log-Eintrag gefunden, können anhand der Correlation ID die Aufrufe vorher und nachher analysiert werden. Der Vorteil ist, dass die Correlation ID über Service-Grenzen hinweg vererbt wird. Somit ist eine Zurückverfolgung bis auf den ersten Service möglich. Andererseits kann gezielt nach Problemen bei einzelnen Usern oder Prozessen gesucht werden, wenn Log-Einträge mit dieser Information vorhanden sind. Die Suche ist nur so gut wie die Logs, die von der Applikation erzeugt werden. Wird einmal ein User Context zum Log erfasst, können im weiteren Verlauf anhand der Correlation ID alle Aktionen des Users nachverfolgt werden.

Istio – mehr Daten; mehr Möglichkeiten

Die oben beschriebene Correlation ID ist eine Möglichkeit, um dem Handel und Wandel der Microservices auf den Grund zu gehen. Hierzu muss jedoch in die Applikation eingegriffen werden, um die notwendigen Log-Einträge zu generieren. Ein anderes Konzept wird durch Istio (*siehe www.istio.io*) angeboten. Istio ist eine Implementation des Open-Source-Projekts Envoy, das einen leichtgewichtigen Service Proxy zur Verfügung stellt. An einen Microservice wird ein sogenanntes Sidecar (ein Beiwagen am Motorrad) „angeschlossen“. Das hierdurch entstandene Konstrukt wird als Service Mesh beschrieben. In diesem Service Mesh werden verschiedene Funktionen angeboten, die es ermöglichen, die Interaktionen zwischen den Microservices zu betrachten (*siehe Abbildung 3*). Weiterhin bietet der Sidecar Funktionen an, um Teile des Service Mesh gesondert behandeln zu können.

Der wichtigste Punkt von Istio ist das Monitoring. Man kann auf Dashboards (*siehe Abbildung 4*) relativ einfach nachverfolgen, welche Microservices genutzt wurden. Mit Grafana lässt sich dies einfach darstellen und so entstehen schnell aussagekräftige Dashboards:

Istio bietet jedoch auch eine Modifikation des Netzwerks an, indem durch

Traffic Management Teile der vorhandenen Umgebung anders eingesetzt werden. Ein wichtiger Aspekt ist etwa das Deployment eines neuen Builds. Wenn eine neue Version eines Microservice ausgerollt wird, kann mit Istio dafür gesorgt werden, dass nur ein Teil der Microservices upgedatet wird. So kann ein Teil der Microservices den Benutzern bereitgestellt werden, um Erfahrungen „in der echten Welt“ zu sammeln, während die bewährte Version weiterhin zur Verfügung steht. Im Traffic Management kann auch beschrieben werden, wie diese neue Version erreichbar ist. Somit stehen weiter die aktuelle Version sowie auch die neuere Version nebeneinander zur Verfügung (*siehe Abbildung 5*). Sollten Fehler auftreten, wird die neuere Version wieder „vom Netz“ genommen und damit zum Beispiel die Menge von fehlgeschlagenen Transaktionen beschränkt.

Dies erlaubt auch A/B Testing, Canary Testing oder zum Beispiel Latenzverzögerungen (der Sidecar „hält“ eine Antwort lange fest). Der große Vorteil ist hierbei die Simulation von technischen Fehlern, ohne auf den Zufall zu hoffen beziehungsweise ohne gekünstelte Fehlerquellen (z.B. Netzwerkfehler, Datenbank-Performance-Probleme usw.) erstellen zu müssen. Im Zusammenspiel mit den oben erwähnten Log-Mechanismen kann so gezielt das Verhalten der Umgebung untersucht werden.

Fazit

Eine Spurensuche bei Microservices gestaltet sich anders als in herkömmlichen verteilten Systemen. Einerseits muss beim Design der Microservices dafür gesorgt werden, dass Informationen bereitgestellt und diese in der geeigneten Form verarbeitet werden, andererseits bieten Tools wie Istio die Möglichkeit, im laufenden Betrieb Tests auszuführen und die Robustheit der Microservices und der Umgebung zu erhöhen, ohne dass ein aufwendiger Neuaufbau der gesamten Infrastruktur stattfinden muss. Die Kombination versetzt ein DevOps Team in die Lage, schnell messbare Resultate eines neuen Builds zu erhalten, der unter echten Praxisbedingungen ausgeführt wird.



Matthias Fuchs
matthias.fuchs@devtops.gmbh



Andreas Chatziantoniou
andreas.chatziantoniou@devtops.gmbh



DOAG noon2noon 2019 – Mit Begeisterung der Oracle Cloud auf den Zahn geföhlt

Martin Klier, Performing Databases, Robert Marz, its-people

Frankfurt – am 16. und 17. Januar 2019 nahmen 35 hoch motivierte Techniker und Entscheider an einem Event der besonderen Art teil: Im Rahmen des DOAG noon2noon 2019 konnten sie – meist zum ersten Mal – eigenhändig der Oracle Cloud auf den Zahn föhlen. Von zwölf Uhr mittags bis zum nächsten Tag um „High Noon“ war „Ärmel hochkrepeln und selbst machen“ angesagt. Ganz ohne Marketing-Störgeräusch im Ohr, ohne Druck oder harte Agenda. Dafür mit Spaß, Spieltrieb und auf Wunsch mit sachkundiger Anleitung durch ehrenamtliche Kräfte der DOAG unterstützt!

Fachlicher Schwerpunkt war, die Struktur der Oracle Cloud zu verstehen, mit der Oracle Cloud Infrastructure (OCI) auf grafischem Wege oder per Skript umzugehen, die Servicetypen kennenzulernen, zu nutzen und eine eigene kleine Lösung für sich zu finden und zu bauen. Der praktische Umgang mit Technologien wie Software Defined Network, Virtualisierung, Autonomous Database, Object Storage und Terraform waren das Ziel. Es standen dazu Aufgaben in drei

Schwierigkeitsgraden für die eigene Inspiration zur Verfügung. Diese durften alle Teilnehmer so kombinieren, wie sie es für interessant hielten, und bekamen auf Wunsch „Support“ durch einen anwesenden Experten – und das war durchaus nicht selten der Teilnehmer auf dem nächsten Platz!

Die gelebte Freude an der Technik, am Lösen von kleineren und größeren Kniffligkeiten, einfacher Erfahrungsaustausch mit den Nachbarn sowie das Mitnehmen von Erfahrung und der eigenen Errungenschaften zählten zu den Highlights der Veranstaltung.

Das engagierte Team um die Organisatoren Robert Marz und Martin Klier bestand aus Katharina Büchse, Axel vom Stein und Sabine Heimsath. Hervorragende technische Unterstützung leistete die Oracle Business Unit Technology + Cloud, insbesondere Tobias Deml, Marcus Schröder und Manuel Hossfeld.

An alle Mitwirkenden und Teilnehmer einen herzlichen Dank für eine wirklich gelungene Veranstaltung, die es in Deutschland so noch nicht gegeben hat!



Martin Klier
martin.klier@performing-db.com



Robert Marz
robert.marz@its-people.de



Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- › Ralf Stebner
- › Martin Hümpfner
- › Silke Henderson-Stuenkel
- › Jan Döschner
- › Michael Tietz
- › Matthias Fuchs
- › Simon Haalck
- › Vincent Farreira
- › Simon Murr
- › Hildo Haenen
- › Vincent Farreira
- › Simon Murr

Firmenmitglieder DOAG

- › Deutsche Rentenversicherung Knappschaft-Bahn-See, Bernd Pohlmeier
- › CROWDCODE GmbH & Co. KG, Ingo Düppe
- › EDEKA IT-service GmbH, Alexander Rott

Termine



Mai

02.05.2019

Regionaltreffen Stuttgart

Jens-Uwe Petersen & Anja Stollberg

06.05.2019

Regionaltreffen NRW (APEX Community)

Martin Schmitter & Torsten Rosenwald
Sankt Augustin

07.05.2019

NextGen-Programm zur APEX Connect 2019

Bonn

07.05.2019

APEX Connect 2019

Bonn

08.05.2019

DOAG Berliner Expertenseminar

Richard Foote
Berlin

09.05.2019

Regionaltreffen Karlsruhe

Reiner Bünger & Jochen Kutscheruk

10.05.2019

DOAG Datenbank Webinar: Oracle GoldenGate in der Cloud

Dmitry Garshin
online

14.05.2019

Regionaltreffen Dresden/Sachsen

Helmut Marten

14.05.2019

Regionaltreffen Bremen

Ralf Kölling

14.05.2019

NextGen-Programm zum Netsuite User Day 2019

Berlin

14.05.2019

NetSuite User Day

Berlin

14.05.2019

Regionaltreffen München/Südbayern

Andreas Ströbel

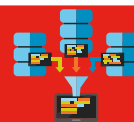
16.05.2019

Regionaltreffen Nürnberg

Martin Klier & Thomas Köppel

16.05.2019

Vorstandssitzung 2/19



Juni

03.06.2019

NextGen-Programm zur DOAG 2019 Datenbank

Düsseldorf

03.06.2019

DOAG 2019 Datenbank

Düsseldorf

05.06.2019

Regionaltreffen Berlin/Brandenburg

Michel Keemers

06.06.2019

Regionaltreffen Rhein-Neckar

Frank Stöcker
Mannheim

06.06.2019

Regionaltreffen München/Südbayern

Andreas Ströbel

13.06.2019

Regionaltreffen Nürnberg

Martin Klier & Thomas Köppel

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.)
ViSdP: Mylène Diacquenod
Redaktionsleitung: Martin Meyer
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Lisa Damerow, Mylène Diacquenod, Marina Fischer, Sanela Lukavica, Robert Marz, Yann Neuhaus, Fried Saacke

Fotonachweis:

Titel: © AndSus/Fotolia
S. 11: © Graf Vshenka/Fotolia
S. 17: © svidten/123RF
S. 26: © macrovector/123RF
S. 30: © chuyu/123RF
S. 36: © Apidech Ninkhali/123RF
S. 42: © mariok/123RF
S. 48: © Jozsef Bagota/123RF
S. 54: © Artisticco LLC/123RF
S. 60: © Bakhtiar Zein/123RF

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

Titel, Gestaltung und Satz:

Caroline Sengpiel,
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

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. 25	MuniQsoft Consulting GmbH www.muniqsoft-consulting.de	S.15	TRILOGY GmbH www.triology.de	S. 29
B4BMEDIA.NET AG www.e3zine.com	S. 41	MuniQsoft Training GmbH www.muniqsoft-training.de	S. 3	Trivadis AG www.trivadis.com	U 4
DOAG e.V. www.doag.org	S. 7, S. 47, U 3	ORACLE Deutschland B.V. & Co. KG www.oracle.com/de	U 2		



2019
DOAG
Konferenz + Ausstellung
19. - 22. November in Nürnberg

EARLY BIRD
BIS ZUM
30. SEPT.

2019.doag.org

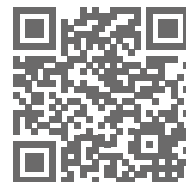


Wir leben Cloud.



■ 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. ■ ■ ■