

# In Nullkommanix zur Datenbank in der eigenen Cloud

Ralf Durben & Manuel Hößfeld  
Oracle Deutschland

## Schlüsselworte

Private Cloud, Datenbank, Provisionierung, Self-Service, Automatisierung, Deployment, DBaaS, Enterprise Manager, Cloud Control

## Einleitung

Die Idee der automatisierten Bereitstellung von Datenbanken durch ein auch von Nicht-DBAs bedienbares Portal bietet viele Vorteile. So wird ein DBA-Team nach der einmaligen Einrichtung eines solchen „Database as a Service“ (DBaaS) weniger mit Routine-Tätigkeiten beschäftigt sein, und durch die standardisierten Prozesse sinkt auch die Fehleranfälligkeit. Besonders die für Test- und Entwicklungszwecke oft benötigten neuen Datenbanken oder „Klone“ (teilweise oder vollständige inhaltliche Kopien) von Datenbanken können so schnell und dennoch kontrolliert bereitgestellt werden.

Dieser Vortrag beschreibt, wie der Aufbau und die Nutzung einer solchen Infrastruktur mit den Mitteln von Oracle Enterprise Manager 12c Cloud Control funktioniert.

## DBaaS in Enterprise Manager 12c – Varianten und Voraussetzungen

Bei näherer Betrachtung des Themas „Database as a Service“ wird schnell klar, daß es je nach Anwendungsszenario durchaus unterschiedliche Ausprägungen geben muss: Mal wird ein vollständiger Klon einer Produktionsdatenbank zum Testen benötigt, mal ist es ausreichend nur einzelne, leere Datenbank-Schemas für Entwickler zu provisionieren.

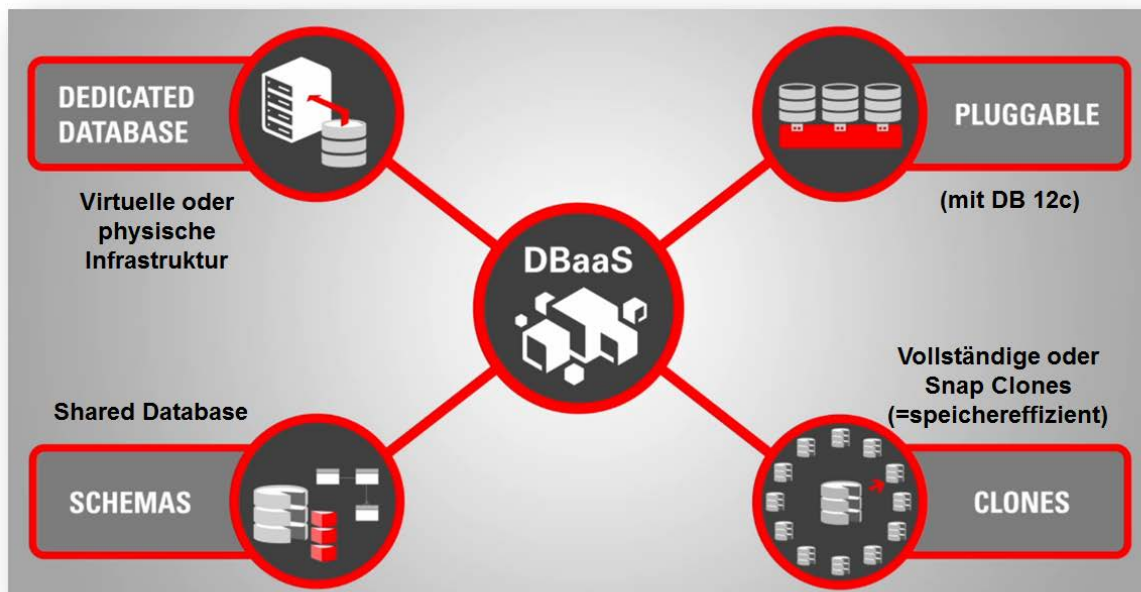


Abbildung 1: Schematische Darstellung der im Enterprise Manager möglichen DBaaS Szenarien

Erfreulicherweise berücksichtigt Enterprise Manager 12c die oben genannten Szenarien und bietet demzufolge auch verschiedene Ausprägungen von DBaaS an:

1. Ein Self-Service Nutzer benötigt eine **neue Datenbank** (ggf. mit bereits enthaltenen Stammdaten darin)
  - a. Der Nutzer erhält tatsächlich eine eigene, vollständige Datenbank mit entsprechend weitreichenden Möglichkeiten und Privilegien. Diese Variante bietet aus Sicht des Nutzers natürlich die meisten Möglichkeiten, stellt aus Betriebs-Sicht aber die höchsten Anforderungen an die verwendeten Ressourcen und ist für viele Fälle letztlich auch „Overkill“.
  - b. Der Nutzer erhält keine vollständige Datenbank, sondern lediglich ein oder mehrere Datenbank Schema(s) in einer zuvor für diesen Zweck eingerichteten und in der Regel gemeinschaftlich mit anderen Nutzern verwendeten Datenbank. Diese Variante wird als „*Schema as a Service*“ bezeichnet und ist im Gegensatz zur zuvor genannten wesentlich schonender im Umgang mit Ressourcen. Damit einhergehend existieren aber natürlich auch die bekannten Einschränkungen einer Schema-Konsolidierung, wie z.B. die Tatsache dass es pro Datenbank bestimmte Benutzer- und Schemanamen nur einmal geben kann. Auch die Isolation zwischen den Nutzern eines solchen „Schema Dienstes“ ist weniger gut möglich als bei einer eigenständigen Datenbank pro Nutzer.
  - c. Der Nutzer erhält eine „*Pluggable Database*“ (*PDB*) innerhalb einer „Container Database“ (*CDB*). Diese neue Möglichkeit basiert auf der sog. „Oracle Multitenant Option“, welche mit dem *Datenbankrelease 12c* eingeführt wurde und funktioniert daher im Gegensatz zu den beiden o.g. Varianten natürlich auch nur mit diesem. Pluggable Databases stellen aus Sicht eines Datenbank-Anwenders eine vollwertige Datenbank dar, benötigen aber aufgrund der gemeinsamen Nutzung von Speicher und Hintergrundprozessen weniger Ressourcen als es eine gleiche Anzahl Datenbanken „konventioneller Architektur“ auf ein und demselben System tun würde. Auch die Verwaltung der Datenbanken aus Sicht des Betreibers ist effizienter, da er viele Aufgaben gemeinschaftlich auf Ebene einer Container-Datenbank erledigen kann.  
„*Pluggable Database as a Service*“ (*PDBaaS*) vereinbart daher in sich die Vorteile einzeln provisionierter Datenbanken mit denen der Schemakonsolidierung und ist somit für den Aufbau von Datenbank Clouds prädestiniert.  
(Weitere Informationen über die Eigenschaften und Funktionsweise von Oracle Multitenant finden sich z.B. auf der unter [1] aufgeführten Webseite.)
2. Ein Self-Service Nutzer benötigt einen **Klon einer bereits bestehenden Datenbank** (zu Testzwecken)
  - a. Für Performance-intensive Tests (z.B. Lasttests) wird eine vollständige und unabhängige Kopie einer Datenbank bereitgestellt
  - b. Für einfache, eher funktionale Tests wird ein sog. „*Snap Clone*“ einer Datenbank bereitgestellt. Dieser belegt (sofern der zugrundeliegende Storage dazu in der Lage ist) dank „Copy-on-Write“ Snapshots nur einen Bruchteil des Speicherplatzes einer vollen Kopie, kann aber ohne das „Original“ nicht verwendet werden

In den meisten Fällen wird unabhängig vom Szenario außerdem zu Zwecken der (internen oder externen) Verrechnung eine Messung der vom Nutzer tatsächlich verwendeten Ressourcen in der Datenbank Cloud erfolgen („Metering & Chargeback“). Die hierfür verwendbaren Metriken umfassen neben grundlegenden Eigenschaften einer Datenbank (wie z.B. Version, etwaige Anzahl RAC Knoten) auch detaillierte nutzungsbezogene Größen wie z.B. die Anzahl ausgeführter SQL Operationen oder der CPU Verbrauch.

Neben den größtenteils bereits oben in der Beschreibung genannten, spezifischen Voraussetzungen für diese unterschiedlichen DBaaS-Szenarien müssen außerdem in jedem Fall folgende allgemeine Bedingungen erfüllt sein:

- Enterprise Manager Agenten müssen auf den zu verwendenden Servern installiert sein
- Die Oracle Datenbank Software (bei Verwendung von RAC außerdem die Grid Infrastructure) muss ebenfalls bereits installiert sein. Außerdem muß natürlich auch jeweils ein Listener konfiguriert sein.
- Die *Software Library* in Enterprise Manager muß konfiguriert sein (um je nach gewünschtem Szenario mit den Vorlagen der zu provisionierenden Datenbanken oder Schemas bestückt zu werden)
- Da es sich bei DBaaS um Funktionalität handelt, welche im sog. *“Cloud Management Pack for Oracle Database”* enthalten ist, muss dieses (inkl. dem *„Database Lifecycle Management Pack“* als dessen Voraussetzung) lizenziert sein.

*Hinweis:* Sämtliche in diesem Vortrag behandelten DBaaS Features funktionieren übrigens vollkommen unabhängig vom verwendeten Betriebssystem und auch davon, ob die zugrundeliegenden Server tatsächlich „physisch vorhanden“ sind (also „Bare Metal“), oder als VMs (z.B. mit Oracle VM) bereitgestellt wurden.

## Einrichten der DBaaS Funktionalität in Enterprise Manager Cloud Control

Wie bereits aus dem vorigen Kapitel ersichtlich wurde, sind zunächst einige Voraussetzungen zu erfüllen, bevor man DBaaS mit Cloud Control nutzen kann. Die dazu nötigen Vorbereitungen und administrativen Tätigkeiten werden in Enterprise Manager durch zwei Rollen abgedeckt: Den „Cloud Administrator“ (EM\_CLOUD\_ADMINISTRATOR) und den „Self-Service Administrator“ (EM\_SSA\_ADMINISTRATOR) – wobei rein technisch natürlich nichts dagegen spricht, dass dies in der Praxis die gleiche Person ist.

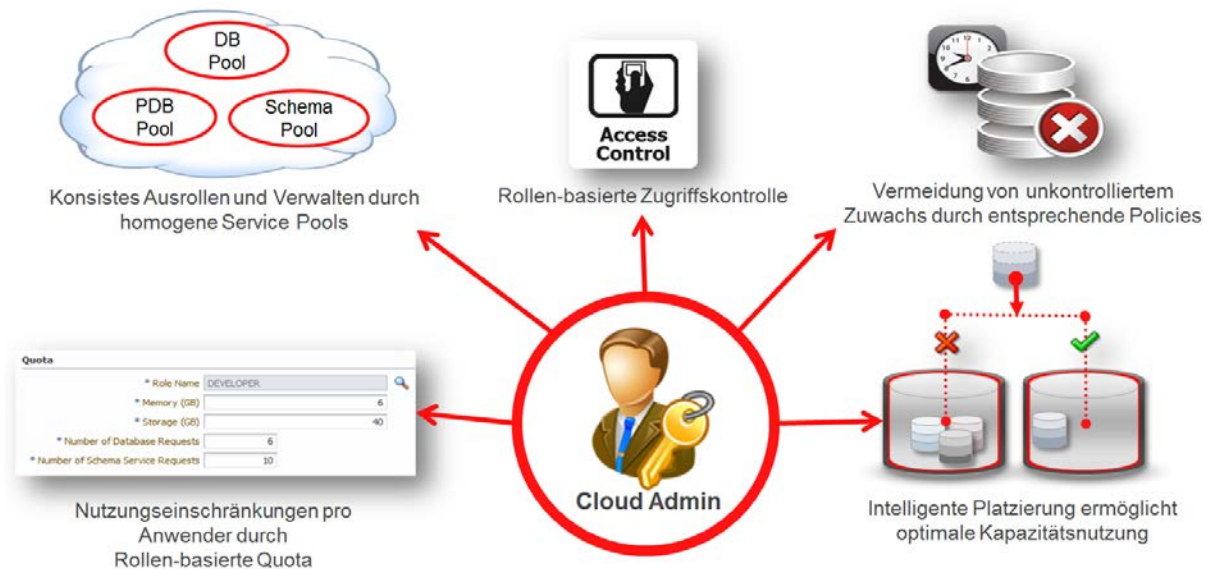


Abbildung 2: Der Cloud Administrator hat viele Möglichkeiten den DBaaS-Betrieb zu konfigurieren

Aus Platzgründen sollen an dieser Stelle nur grob die Schritte genannt werden, welche der Cloud Administrator bzw. der Self-Service Administrator durchführen müssen:

1. Einrichten mindestens einer sog. „PaaS Infrastructure Zone“. Hierbei werden die zu verwendenden Datenbank-Server mit den dafür vorgesehenen Self-Service Rollen verknüpft und bekommen außerdem die benötigten „Host Credentials“ zugewiesen. Auch das Einstellen von „Placement Policies“ – also von Restriktionen, wie viele Datenbanken überhaupt auf einen bestimmten Host provisioniert werden dürfen - wird an dieser Stelle vorgenommen. Die Einrichtung unterschiedlicher Zonen kann z.B. aufgrund geografischer oder organisatorischer Vorgaben erfolgen. Das Zuweisen zu verschiedenen „Datenbankklassen“ bzw. „Service Leveln“ wird dadurch ebenfalls ermöglicht, was später bei Verwaltung und Monitoring auch für mehr Übersicht sorgt.
2. Erzeugen von einem oder mehreren „Database Pools“. Ein Datenbank Pool ist eine Organisationseinheit innerhalb einer PaaS Infrastructure Zone, welche durch Hinzufügen von Cloud Control bereits bekannten Oracle Homes erzeugt wird. Diese liegen auf den Hosts, welche im vorigen Schritt bereits einer Infrastructure Zone zugewiesen wurden. (Hinweis: Jeder Host kann nur Mitglied in einem einzigen Datenbank Pool sein.)

3. Einstellen von Quota (also Nutzungsbeschränkungen für die Cloud Anwender) sowie der sog. „Request Settings“. Letztere bestimmen z.B. für welchen Zeitraum eine Datenbank durch einen Self-Service Nutzer beantragt und genutzt werden kann.
4. Erstellen und Zuweisen von sog. „Service Templates“. Diese stellen die konkrete Basis für die von den Self-Service Nutzern beantragbaren Datenbankdienste dar. Was ein Service Template genau umfasst unterscheidet sich je nach verwendetem DBaaS-Szenario: Es kann sich hierbei z.B. um ein DBCA Template (für die Erstellung vollständiger, neuer Datenbanken) oder um eine Vorlage für die Erzeugung eines Datenbank Schemas (Schema as a Service) handeln. Die Abbildung 3 zeigt als Beispiel den zusammenfassenden letzten Schritt der Definition eines Service Templates für PDBaaS.
5. Optional: Einstellen von Regeln für die nutzungsbezogene Abrechnung.

Die Dokumentation erläutert das Einrichten der für DBaaS notwendigen Infrastruktur sowie des Self-Service Portals im Detail unter [2], [3] und [4].

*Hinweis: Die beiden im Folgenden gezeigten Screenshots stammen von einer Vorversion des für PDBaaS notwendigen DB Plugins (12.1.0.5). Das Aussehen der jeweiligen Bildschirme in der finalen Version kann davon u.U. leicht abweichen.*

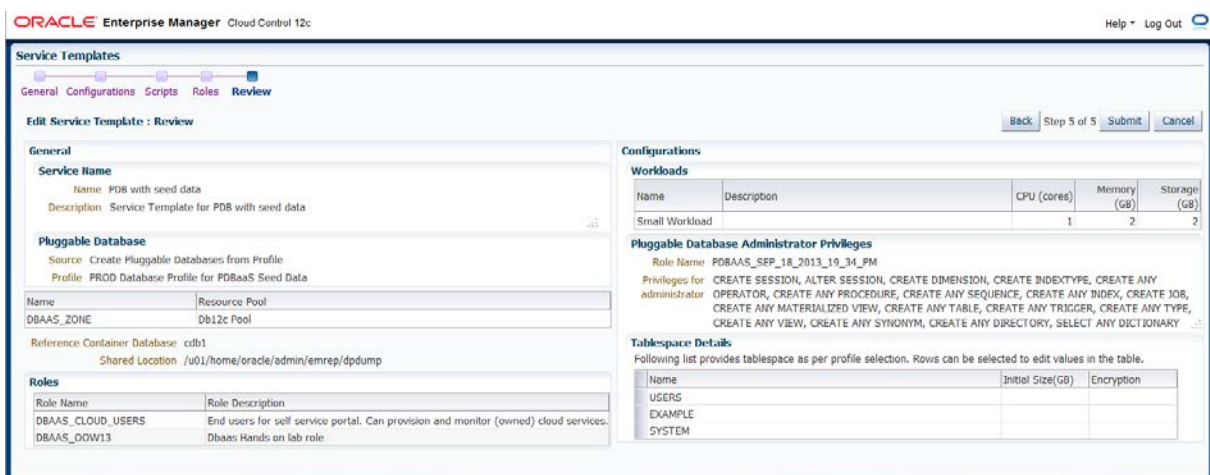


Abbildung 3: Screenshot am Ende der Erstellung eines Service Templates für PDBaaS

## Die Sicht des Nutzers – Beantragen einer Datenbank im Self-Service Portal

Ein Nutzer, welcher mittels des in Enterprise Manager enthaltenen Self-Service Portals eine Datenbank im Sinne von DBaaS beantragen will, benötigt dazu die Rolle EM\_SSA\_USER.

Durch Zuweisen ausschließlich dieses Rechts können Nutzer erzeugt werden, welche sich dann zwar am Enterprise Manager anmelden und das Self-Service Portal benutzen können, aber darüber hinaus keinerlei andere Rechte haben. D.h. obwohl es sich um ein- und dieselbe EM-Installation handelt, welche z.B. auch ein DBA oder der Administrator einer WebLogic-Farm benutzen würde, kann dieser eingeschränkte Nutzer nicht einmal „sehen“, dass es neben dem Portal zum Beantragen von Datenbanken überhaupt noch andere Bereiche im EM gibt.

Nach der Anmeldung kann der Anwender – je nach ihm zugewiesener Quota – beliebig neue Datenbanken beantragen. Der entsprechende Request läuft dann asynchron im Hintergrund und benötigt je nach Szenario und Geschwindigkeit der verwendeten Storage- und Serversysteme nur wenige Minuten.

Sofern es entsprechend konfiguriert wurde, erhält der Nutzer nach Beendigung des Requests eine entsprechende eMail von Oracle Enterprise Manager zugeschickt. Das gleiche gilt sinngemäß, falls ein automatisches „Ablaufen“ der provisionierten Datenbank eingestellt wurde – d.h. wenn diese gelöscht oder archiviert wird.

The screenshot displays the Oracle Database Cloud Self Service Portal interface. The top navigation bar includes the Oracle logo, user information (EDWARDMUTZ), and a 'Log Out' button. The main content area is divided into several sections:

- Notifications:** Shows 'Databases Due to Expire in Next 7 Days: 0' and 'Service Templates Published in Last 7 Days: 0'.
- Usage:** Contains three progress bars: 'Databases: 0' (range 0-4), 'Schema Services: 0' (range 0-20), and 'Pluggable Databases: 3' (range 0-5). The 'Pluggable Databases: 3' bar is highlighted with a red rectangle.
- Memory:** Shows 'Memory: 4 GB' with a progress bar (range 0-10).
- Storage:** Shows 'Storage: 4 GB' with a progress bar (range 0-20).
- Databases:** A table listing existing databases. A red arrow points to the 'Instance Type' column. The table contains two rows:

Name	Instance Type	Instance Status	Start	End	Zone Name
cdb1_EMEP_002	Pluggable Database	↑	27-Sep-2013 22:0...	Indefinitely	DBAAS_ZONE
cdb1_EMEP_003	Pluggable Database	↑	27-Sep-2013 22:1...	Indefinitely	DBAAS_ZONE

- Requests:** A table showing the status of database creation requests. The table contains three rows:

Name	Status	Service Instance	Request Type	Submission Date	Begin Date
EDWARDMUTZ - Fri Sep 27 22:05:43 EDT 2013...	Success	cdb1_EMEP_002	Create Service Ins...	27-Sep-2013 22:0...	27-Sep-2013 22:0...
EDWARDMUTZ - Fri Sep 27 22:13:08 EDT 2013...	Success	cdb1_EMEP_003	Create Service Ins...	27-Sep-2013 22:1...	27-Sep-2013 22:1...
CLOUDUSER Request - PDBaaS_CREATE_32_50	Running		Create Service Ins...	14-Oct-2013 05:3...	14-Oct-2013 05:3...

Abbildung 4: Homepage eines PDBaaS Nutzers im Self-Service Portal

Wie der Screenshot in Abbildung 4 zeigt, werden sämtliche in diesem Vortrag genannten Varianten von DBaaS gleichberechtigt dargestellt und bedient – soweit möglich und sinnvoll also auch unabhängig von der zugrundeliegenden Technik bzw. Implementierung. In diesem konkreten Fall bemerkt der Nutzer z.B. lediglich an der Spalte „Instance Type“ sowie an der entsprechenden

grafischen Darstellung seiner Quota links im Bild, dass es sich bei den von ihm zuvor beantragten Datenbanken um „Pluggable Databases“ handelt und nicht um eigenständige („konventionelle“) Datenbanken.

In der Oberfläche des Self-Service Portals kann der Nutzer für die von ihm beantragten Datenbanken grundlegende Monitoring- und Verwaltungstätigkeiten durchführen sowie für die Nutzung relevanten Daten einsehen (z.B. den Connection String.)

Wenn es vom Cloud Administrator entsprechend eingerichtet wurde, ermöglicht das Self-Service Portal außerdem die Kontrolle über Backup und Restore der entsprechenden Datenbank.

## **Kontaktadressen:**

Manuel Hoßfeld

Oracle Deutschland B.V. & Co. KG

Robert-Bosch-Str. 5

D-63303 Dreieich

Telefon: +49 (0) 6103-397 494

E-Mail [Manuel.Hossfeld@oracle.com](mailto:Manuel.Hossfeld@oracle.com)

Ralf Durben

Oracle Deutschland B.V. & Co. KG

Rudolfstr. 5

D-59556 Lippstadt

Telefon: +49 (0) 211-74839 461

E-Mail [Ralf.Durben@oracle.com](mailto:Ralf.Durben@oracle.com)

---

[1] Seite im Oracle Technology Network zu Pluggable Databases mit Oracle Multitenant:  
<http://www.oracle.com/technetwork/database/multitenant/overview/index.html>

[2] Kapitel 9 des Enterprise Manager Cloud Administration Guide: “Setting up the PaaS Infrastructure”  
[http://docs.oracle.com/cd/E24628\\_01/doc.121/e28814/cloud\\_paas\\_setup.htm](http://docs.oracle.com/cd/E24628_01/doc.121/e28814/cloud_paas_setup.htm)

[3] Kapitel 11 des Enterprise Manager Cloud Administration Guide: “Setting up a DBaaS Cloud”  
[http://docs.oracle.com/cd/E24628\\_01/doc.121/e28814/cloud\\_db\\_setup.htm](http://docs.oracle.com/cd/E24628_01/doc.121/e28814/cloud_db_setup.htm)

[4] Kapitel 12 des Enterprise Manager Cloud Administration Guide: “Setting up the DBaaS Self-Service Portal”  
[http://docs.oracle.com/cd/E24628\\_01/doc.121/e28814/cloud\\_db\\_portal.htm](http://docs.oracle.com/cd/E24628_01/doc.121/e28814/cloud_db_portal.htm)