

Keine Angst vor Multitenant Database!

Johannes Ahrends
CarajanDB GmbH
Erftstadt

Schlüsselworte

Oracle 12c Multitenant Database, Standard Edition

Einleitung

In der Oracle Dokumentation, genauer im Oracle 12c Database Upgrade Guide, Kapitel 8.1.1 steht es schwarz auf weiß:

"The non-CDB architecture is deprecated in Oracle Database 12c, and may be desupported and unavailable in a later Oracle Database release. Oracle recommends use of the CDB architecture."

Was ist CDB Architecture? Was bedeutet es, dass non-CDB deprecated ist? Sind mit der Umstellung zusätzliche Lizenzkosten verbunden? Was muss ich tun? Was passiert mit meinen Anwendungen? Können die überhaupt damit umgehen? Ist dann alles viel besser?

In dem Vortrag werden diese und weitere Fragen thematisiert und beantwortet, damit Sie bei der anstehenden Migration auf 12c gar nicht mehr überlegen, CDBs NICHT zu nutzen.

Begriffsklärung

Leider ist die Begriffsklärung gar nicht so einfach aber wenn man Sie einmal durchschaut hat, werden viele Fragen schon einmal geklärt sein.

Non-CDB: hierbei handelt es sich um die bis Version 11g gültige und in Oracle 12c immer noch mögliche Architektur: Es gibt eine Datenbank, die ein Data Dictionary hat und Tablespaces mit Daten. Alle Daten werden über dieses Data-Dictionary verwaltet. Ein Undo-Tablespace und die Redolog-Dateien runden die Datenbank ab.

CDB: das Gegenteil davon... Hier gibt es zunächst ein zentrales Data Dictionary, was im so genannten Root-Container verwaltet wird. Zusätzlich gibt es aber weitere Data Dictionaries in den einzelnen Anwendungsdatenbanken, den Pluggable Databases (PDB). Um eine Anwendung betreiben zu können, benötigt man hier also eine CDB und mindestens eine zusätzliche PDB mit den Anwendungsdaten und Tablespaces.

Single Tenant Database: Eine CDB in die exakt eine Pluggable Database eingehängt worden ist. Diese spezielle Ausführung ist ohne zusätzliche Lizenzkosten sowohl in der Standard Edition 2 als auch der Enterprise Edition erlaubt.

Multitenant Database: in diesem Fall sind in einer CDB mehrere PDBs eingehängt. Zunächst benötigt man dafür die Multitenant Database Option und natürlich ist das nur in der Enterprise Edition möglich. Dann bietet diese Architektur aber eine sehr flexible Infrastruktur für die Verwaltung von Datenbanken. Statt sich um viele kleine Datenbanken kümmern zu müssen, verwaltet man jetzt eine zentrale Datenbank, die CDB und damit automatisch alle eingehängten PDBs.

„Was bedeutet es, dass NON-CDB deprecated ist?“

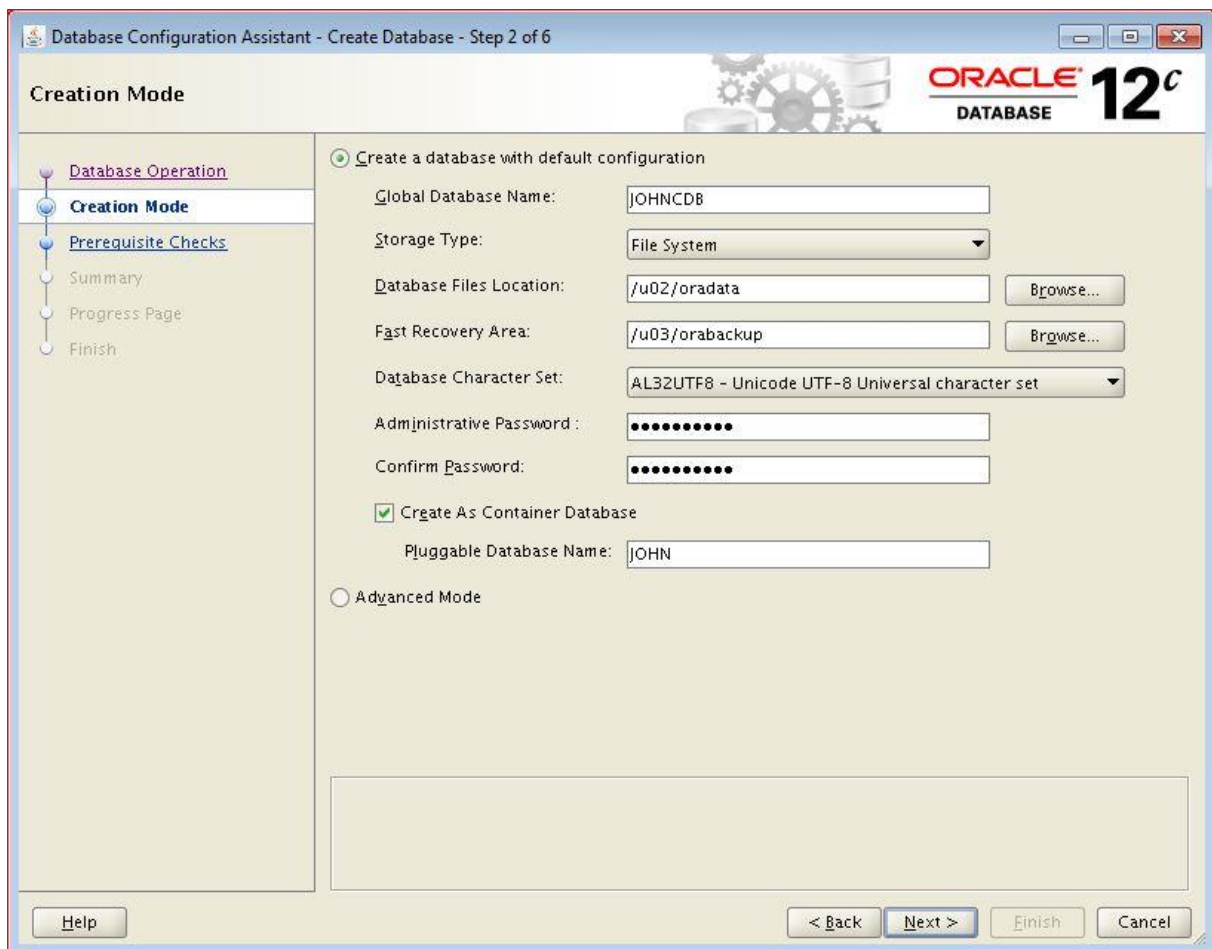
„Non-CDB is deprecated“ bedeutet zunächst einmal nichts anderes, als dass neue Funktionen eventuell nicht mehr mit der bisherigen Architektur funktionieren. Derzeit ist es aber eher umgekehrt. Wer z.B. das neue Information Lifecycle Management (ILM) verwenden möchte, muss dafür eine NON-CDB aufbauen. Es gibt meines Wissens kein Feature, das mit einer CDB funktionieren würde aber mit einer NON-CDB nicht. Aber warten wir es ab! Im nächsten Release ist damit zu rechnen, dass die Features erweitert werden, die dann nur noch in einer CDB funktionieren.

„Sind mit der Umstellung zusätzliche Lizenzkosten verbunden?“

Nein! Solange eine CDB mit nur einer PDB betrieben wird, benötigt man keine Multitenant Option.

„Was muss ich tun?“

Ganz einfach: bei der Erstellung einer Datenbank mit dem dbca einfach die Option „Create As Container Database“ anwählen und den Namen der gewünschten Pluggable Database angeben.



Für den Anfang empfiehlt es sich nach meiner Erfahrung, als globalen Datenbanknamen das Kürzel „CDB“ als Pre- oder Postfix anzugeben (im Beispiel JOHNCDB). Der Name der PDB ist dann vielleicht der bisherige Datenbankname (z.B. JOHN).

„Was passiert mit meinen Anwendungen?“ „Können die überhaupt damit umgehen?“

Mit den Anwendungen passiert nichts. Sie greifen weiterhin über einen TNS-Alias auf die Datenbank zu, benutzen als Ziel jetzt aber die PDB. Das ist alles. Ähnlich wie bei Tablespaces, von denen die Anwendung ja auch nichts weiß, ist eine PDB für sie transparent. Empfehlen würde ich allerdings, als Servicennamen nicht den Namen der PDB zu verwenden (im obigen Beispiel also „JOHN“) sondern für jede Anwendung einen eigenen Service zu definieren. Das erlaubt die höchstmögliche Flexibilität beim Verbindungsaufbau.

„Ist dann alles viel besser?“

Wo Licht ist, ist auch Schatten. Sicherlich gibt es in der aktuellen Version noch einige Einschränkungen. Der wesentliche Unterschied von NON-CDB zu CDB besteht aber darin, dass ich mich nicht mehr einfach mit der ORACLE_SID an die PDB anmelden kann. Alle Skripte, die lokal auf dem Server gestartet werden, müssen also angepasst werden. Und man hat bei der Single Tenant Architektur zwei SYSTEM, SYSAUX und TEMP Tablespaces. Das kann schon einmal für Verwirrung sorgen. Aber ich bin sicher, dass in den kommenden Releases speziell das „Plugin“ und „Unplug“ vereinfacht werden und dann wird die Verwaltung von Datenbanken, Patching und Upgrades um ein vielfaches einfacher. Also nicht „Abwarten und Tee trinken“!

Kontaktadresse:

Johannes Ahrends
CarajandB GmbH
Siemensstraße 25
D-50374 Erftstadt

Telefon: +49 (22 35) – 1 70 91 84
Fax: +49 (22 35) – 1 70 89 79
Mobil: +49 (1 70) - 4 05 69 36
E-Mail: johannes.ahrends@carajandb.com
Internet: www.carajandb.com