



Multitenant – Ein Review

DEVK

Tim Hensel
Johannes Ahrends

DEVK
CarajanDB GmbH

Oracle DBA seit 2004

- Studium zum Dipl. Wirtschaftsinformatiker
- Diplomarbeit „Einführung Oracle 10g bei der Stadtverwaltung Köln“
- Oracle Certified Professional 11g
- Seit 6 Jahren bei den DEVK Versicherungen in Köln



Themenschwerpunkte:

- Oracle Hochverfügbarkeit (RAC und Data Guard)
- Oracle Backup & Recovery
- Oracle Security
- Andere Datenbanksysteme: PostgreSQL, Aurora (Cloud)



Hobbies:

- Zwei Kinder
- Fußball



Tim Hensel
Senior Datenbank-Administrator

GESAGT. GETAN. GEHOLFEN.

DEVK

Oracle Spezialist seit 1992

- 1992: Presales bei Oracle in Düsseldorf
- 1999: Projektleiter bei Herrmann & Lenz Services GmbH
- 2005: Technischer Direktor ADM Presales bei Quest Software GmbH
- 2011: Geschäftsführer CarajanDB GmbH



2011 → Ernennung zum Oracle ACE

Autor der Bücher:

- Oracle9i für den DBA, Oracle10g für den DBA, Oracle 11g Release 2 für den DBA



DOAG Themenverantwortlicher Datenbankadministration,
Standard Edition



Hobbies:

- Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
- Motorradfahren (nur draußen)
- Singen (überall)

Multitenant Database

Das Projekt

Vorgehensweise bei der Migration

Vorteile Multitenant

Nachteile Multitenant



Multitenant Database

DEVK

Non-CDB

- Architektur bis Oracle 11.2

Multitenant Architektur

- Alternative Architektur ab Version 12.1.1 für alle Editionen

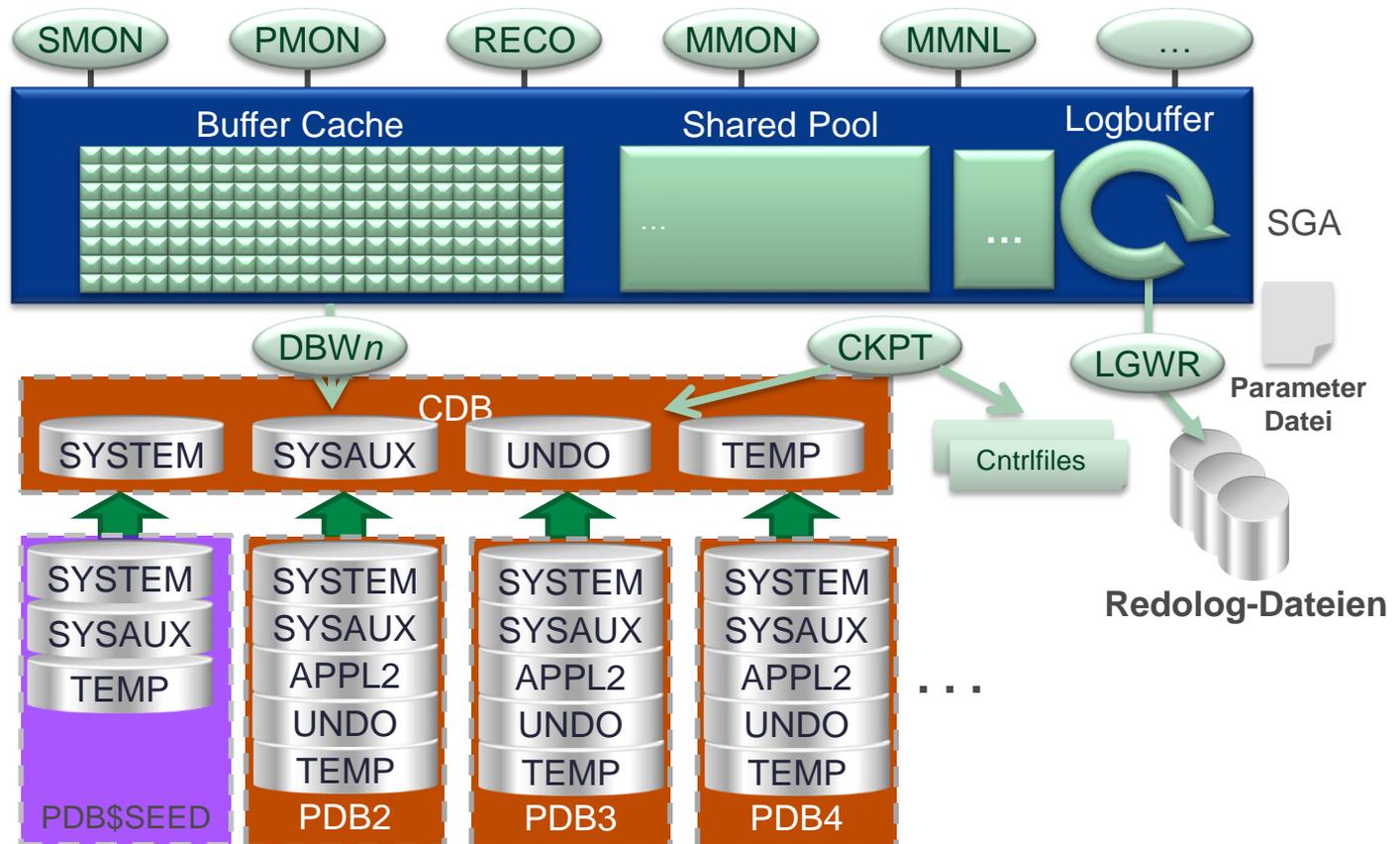
Multitenant Option

- Kostenpflichtig für die Enterprise Edition
- Bis zu 252 PDBs pro CDB (Exadata ab 12.2 4096 PDBs)

The non-CDB architecture was deprecated in Oracle Database 12c. It can be desupported and unavailable in a release after Oracle Database 12c Release

Oracle recommends use of the CDB architecture.

The non-CDB architecture will be desupported from Oracle 20c on (OOW 2019 Dominic Giles)



3 PDBs in jeder Edition

DEVK

Feature / Option / Pack	SE2	EE	EE-ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS	Notes
Oracle Multitenant - # of PDBs	3	252	4096	3	3	4096	4096	4096	<p>The number value in each column indicates the maximum number of pluggable databases (PDBs) that can be created for each offering.</p> <ul style="list-style-type: none"> • For all offerings, if you are not licensed for Oracle Multitenant, then you may have up to 3 PDBs in a given container database at any time. • EE: Extra cost option; if you are licensed for Oracle Multitenant, then you can create up to 252 PDBs. • EE-ES: Extra cost option; if you are licensed for Oracle Multitenant, then you can create up to 4096 PDBs. • DBCS EE-HP, DBCS EE-EP, and ExaCS: Included option; you can create up to 4096 PDBs.
CDB Fleet Management	N	N	Y	N	Y	Y	Y	Y	
PDB Snapshot Carousel	N	N	Y	N	Y	Y	Y	Y	
Refreshable PDB switchover	N	N	Y	Y	Y	Y	Y	Y	



Quelle: Oracle Database Database Licensing Information User Manual 19c E94254-10 September 2019



Das Projekt

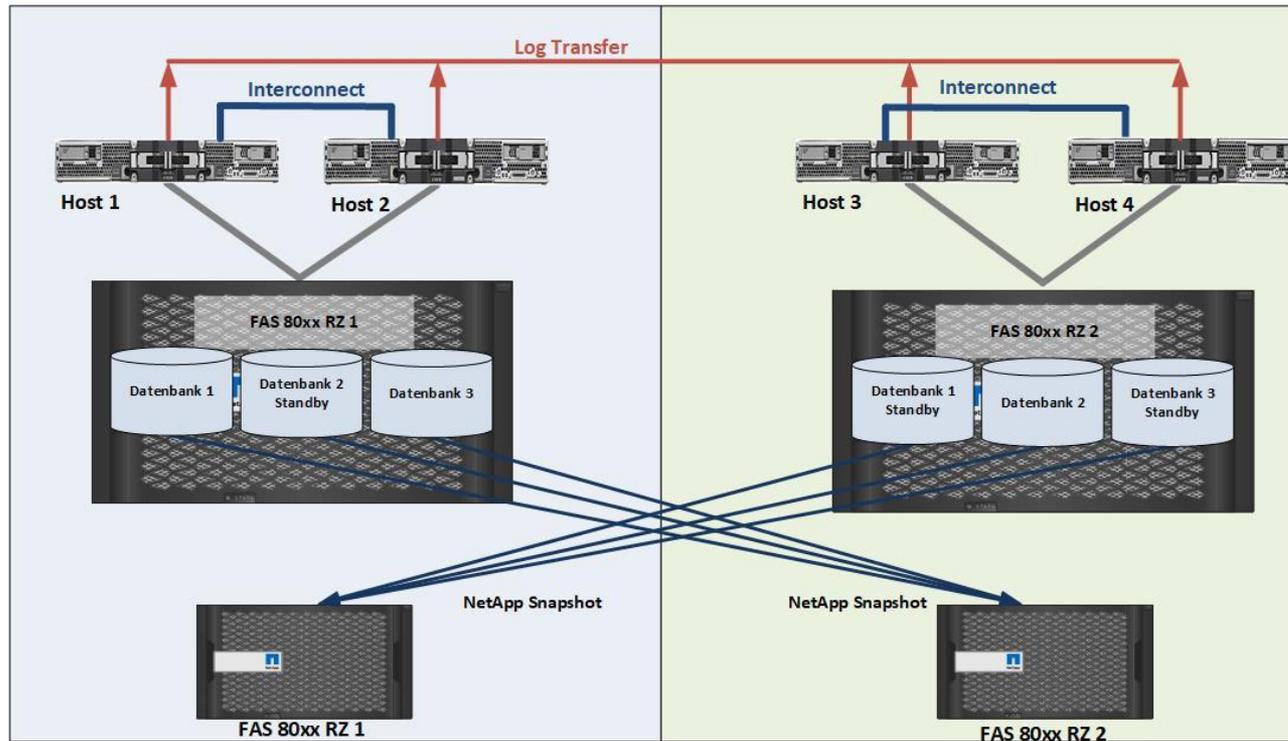
DEVK

Migration von 8 Oracle 11g Datenbank 6 Knoten Stretched RAC nach

- Oracle 12c (12.2.0.1)
- 2 Knoten RAC
- NetApp mit Direct NFS
- Data Guard
- Multitenant
- 4 Stages (Maintenance, Test, Vorproduktion, Produktion)
- Derzeit (Okt. 2019) 8 Datenbanken (CDBs)

Start des Projektes Migration August 2015

Start des Projektes Multitenant Januar 2016



Anzahl PDBs

- Test: 124
- Vorproduktion: 162
- Produktion: 107
- Insgesamt: 393



Migration / Upgrade

DEVK

Identifizieren des Projektes → Absprache mit Projekt-Verantwortlichen

- Wie heißt die Anwendung?
- Wie greift der Client darauf zu (Java, App-Server, Direkt, ...)?
- Gibt es Restriktionen?

Erstellen einer PDB für die Anwendung

Data Pump Export des/der Schemata

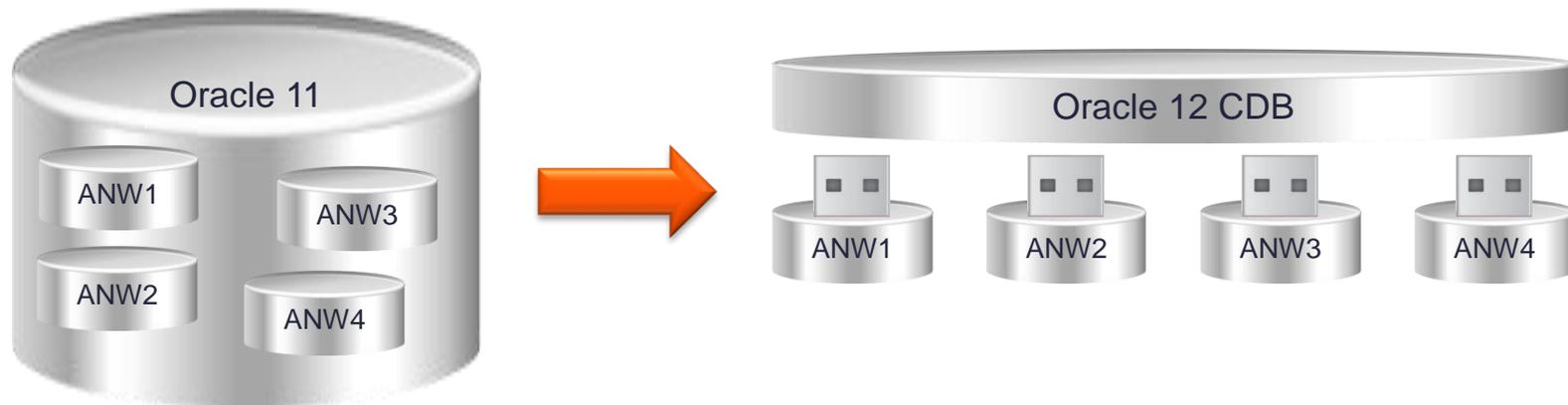
Data Pump Import des/der Schemata

Logdaten der Migration müssen 10 Jahre aufbewahrt werden

Quelle: 8 Datenbanken mit ca. 120 Anwendungen

(Schemakonsolidierung)

Ziel: 7 Datenbanken mit ca. 120 Pluggable Databases



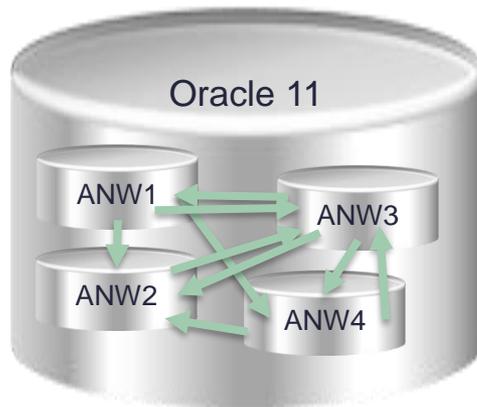
Anwendung 1: Wir brauchen Oracle 12.1

Anwendung 2: Wir brauchen 12.2 aber nicht Oracle 18

Anwendung 3: Wir brauchen unbedingt WE8ISO8859P15

Anwendung 4: Wir brauchen unbedingt Unicode

Anwendung 5: Wir benötigen einen Oracle 10 Client (nicht bewusst!)



ANW1 greift auf Dateien von ANW3 zu
ANW3 benötigt Daten von ANW2

...

Ein User ändert Daten in ANW1 und
ANW4

Ein User liest Daten aus ANW2 und ANW4

Schemanamen enthalten den Stage (z.B. ANW1_T, ANW1_V, ANW1_P)

- Oracle kennt kein Rename Schema (offiziell)

➔ Kein Cloning der PDB über Stagegrenzen möglich

Anwendung arbeitet mit SID statt Servicennamen

- Ist bei PDBs nicht möglich

Anwendung benutzt Serveridentifikation (OPPS\$-Account)

- OPS\$ muss in der CDB angelegt werden (C##OPS\$)
- Logon Trigger für Redirect auf die PDB

Anwendung hat nur eine maximale Zeichenlänge für Connect (hier 30 Zeichen)

- Anwendung kann nicht hochverfügbar sein (Alternative Adresse)

Oracle 11g:

Zugriffe auf andere Schemata / Anwendungen durch GRANT der Privilegien

Oracle 12c mit Multitenant:

Zugriffe auf andere PDBs nur über Database Link möglich

- Database Link erlauben keine Synonyme
 - Erstellen von Views für Remote Datenbank Objekte
 - Synonyme auf Views
 - Vollqualifizierter Zugriff nicht mehr möglich (Dummyuser erforderlich)

Je Competence Center (3) eine eigene CDB

Kritische Statistik Datenbank alleine (eigener Server in Produktion)

Eine CDB für Java Anwendungen (Patching Problematik)

Eine CDB für Unicode (derzeit default noch WE8ISO)

Eine CDB für Oracle 12.2

Zusätzlich eine CDB für Management (OEM, RMAN Katalog, etc.)

Minimierung der Downtime für Patching und Upgrades

Flexible Patchzeiten

Jede Anwendung kann eigene Public Synonyme haben

Mehrere Versionen einer Anwendung in Produktion (Mandanten)

DBA Berechtigungen

Zurücksetzen (Flashback) einzelner Anwendungen



Arbeiten mit PDBs

DEVK

Eine PDB wird beim Starten der Instanz nicht automatisch geöffnet

Daher immer:

```
SQL> CREATE PLUGGABLE DATABASE <PDBNAME>  
  
SQL> ALTER PLUGGABLE DATABASE <PDBNAME> OPEN INSTANCES=ALL;  
  
SQL> ALTER PLUGGABLE DATABASE <PDBNAME> SAVE STATE;
```

SAVE STATE: sichert den derzeitigen Status der PDB

Nachinstallation einer Funktion, z.B. Context (catctx.sql)

- Keine Verwendung von SQL*Plus möglich
- Statt dessen perl und catcon.pl

```
$ORACLE_HOME/perl/bin/perl -u SYS -n 8 -d $ORACLE_HOME/ctx/admin ... catctx.sql
```

Sorgt dafür, dass das Package in allen PDBs ausgeführt wird (hier parallel 8)

Zitat:

„The PDB name must be unique in the CDB, and it must be unique within the scope of all the CDBs whose instances are reached through a specific listener.“

Erstellen von NetApp Snapshots

- Snapshot PDBs haben den gleichen Namen wie die originalen
- Standard Services können nicht zugeordnet werden

IMMER MIT ANWENDUNGSSPEZIFISCHEN SERVICES ARBEITEN!

Example 19-7 Showing the Data Files for Each PDB in a CDB

This example queries the `DBA_PDBS` and `CDB_DATA_FILES` views to show the name and location of each data file for all of the PDBs in a CDB, including the PDB seed.

```

COLUMN PDB_ID FORMAT 999
COLUMN PDB_NAME FORMAT A8
COLUMN FILE_ID FORMAT 9999
COLUMN TABLESPACE_NAME FORMAT A10
COLUMN FILE_NAME FORMAT A45

SELECT p.PDB_ID, p.PDB_NAME, d.FILE_ID, d.TABLESPACE_NAME, d.FILE_NAME
FROM DBA_PDBS p, CDB_DATA_FILES d
WHERE p.PDB_ID = d.CON_ID
ORDER BY p.PDB_ID;

```

Sample output:

PDB_ID	PDB_NAME	FILE_ID	TABLESPACE	FILE_NAME
2	PDB\$SEED	6	SYSAUX	/disk1/oracle/dbs/pdbseed/cdbl_ax.f
2	PDB\$SEED	5	SYSTEM	/disk1/oracle/dbs/pdbseed/cdbl_db.f
3	HRPDB	9	SYSAUX	/disk1/oracle/dbs/hrpdb/hrpdb_ax.f
3	HRPDB	8	SYSTEM	/disk1/oracle/dbs/hrpdb/hrpdb_db.f
3	HRPDB	13	USER	/disk1/oracle/dbs/hrpdb/hrpdb_usr.dbf
4	SALESPDB	15	SYSTEM	/disk1/oracle/dbs/salespdb/salespdb_db.f
4	SALESPDB	16	SYSAUX	/disk1/oracle/dbs/salespdb/salespdb_ax.f
4	SALESPDB	18	USER	/disk1/oracle/dbs/salespdb/salespdb_usr.dbf

Default: PDB\$SEED wird bei allen CDB-Views ignoriert

Laut Oracle muss das so sein (warum auch immer)

Ausschalten mit:

- 12.1: `exclude_seed_cdb_view = FALSE`
- 12.2: `_exclude_seed_cdb_view = FALSE`

Setting this parameter to `FALSE` would return results for the seed database when querying against the CDB views

```
ALTER SYSTEM SET "_exclude_seed_cdb_view"=FALSE;
SELECT con_id, tablespace_name,
       trunc(sum(bytes)/1024/1024) Mbytes,
       trunc(sum(decode(maxbytes,0,bytes,maxbytes)/1024/1024)) maxmbytes
FROM   cdb_data_files
GROUP BY con_id, tablespace_name
ORDER BY con_id;
```

CON_ID	TABLESPACE_NAME	MBYTES	MAXMBYTES
1	USERS	5	32767
1	UNDOTBS1	920	32767
1	SYSTEM	800	32767
1	SYSAUX	870	32767
2	SYSTEM	250	32767
2	SYSAUX	550	32767
3	USERS	100	5120
3	SYSAUX	600	32767
3	SYSTEM	260	32767
4	USERS	300	1024
4	SYSTEM	260	32767
4	SYSAUX	600	32767

Connect mit ORACLE_SID

→ Kein Connect an die PDB
möglich

The screenshot shows the Oracle Logon dialog box with the following fields and options:

- Database:** A dropdown menu.
- User ID:** A text input field.
- Password:** A text input field with masked characters (dots).
- Logon:** A button to attempt the connection.
- Exit:** A button to close the dialog.
- Configuration Name:** A dropdown menu.
- Database Type:** A dropdown menu set to "Oracle".
- Host name or IP:** A text input field containing "<HOST>".
- Database Port:** A text input field containing "1521".
- Database Name:** A text input field containing "<SID>".
- Database Alias:** An empty text input field.
- Operator Mode:** A checked checkbox.
- Table Owner:** A text input field.
- Test:** A button to test the connection.
- Save:** A button to save the configuration.
- Remove:** A button to remove the configuration.
- Import/Exp...:** A button for import/export operations.
- Help:** A button for help.

Listener Parameter

```
USE_SID_AS_SERVICE_listener=ON
```

Der „SID“ Parameter in der Connect Beschreibung wird als Servicename interpretiert

Alle Tablespaces werden als Bigfile Tablespaces angelegt

Ausnahme: USERS Tablespace

```
CREATE PLUGGABLE DATABASE <PDBNAME> ...  
DEFAULT TABLESPACE users ...;
```

```
SELECT con_id, tablespace_name, bigfile FROM cdb_tablespaces;
```

CON_ID	TABLESPACE_NAME	BIG
1	SYSTEM	YES
1	SYSAUX	YES
1	UNDOTBS1	YES
1	TEMP	YES
3	SYSTEM	YES
3	SYSAUX	YES
3	UNDOTBS1	YES
3	TEMP	YES
3	USERS	NO

Lösung laut Oracle Support:

- Der Users-Tablespace muss beim Anlegen der CDB schon für die PDB\$SEED mit angegeben werden (editieren der CREATE Database Skripte)

```
create database CDB12201
  SET DEFAULT BIGFILE TABLESPACE
  datafile '/u01/data/CDB12201/system.dbf' size ...
  sysaux datafile '/u01/data/CDB12201/sysaux.dbf' size ...
  BIGFILE undo tablespace undotbs1 datafile '/u01/data/CDB12201/untotbs.dbf' size ...
  BIGFILE default temporary tablespace temp tempfile '/u01/data/CDB12201/temp.dbf' size ...
  BIGFILE default tablespace users datafile '/u01/data/CDB12201/users.dbf' size ...;
logfile
...
ENABLE PLUGGABLE DATABASE
SEED FILE_NAME_CONVERT=('/u01/data/CDB12201','/u01/data/CDB12201/pdbseed');
```



Patches und Upgrades

DEVK



New Paradigms for Rapid Patching and Upgrades

The investment of time and effort to patch one multitenant container database results in patching all of its many pluggable databases. To patch a single pluggable database, you simply unplug/plug to a multitenant container database at a different Oracle Database software version.

To upgrade all hosted pluggable databases in a container database, simply upgrade the container database and all hosted pluggable databases are upgraded 'in-place'.

To upgrade a single PDB, you simply unplug/plug the pluggable database in to a container database at a higher version and upgrade the pluggable database as described in the [Database Upgrade Guide](#).

ALTERNATIVE FAKTEN

Die Investition von Zeit und Aufwand für das Patchen einer Multitenant-Containerdatenbank führt dazu, dass all die vielen zugehörigen plug-fähigen Datenbanken gepatcht werden. Um eine einzelne plug-fähige Datenbank zu patchen, trennen Sie einfach mit einer anderen Oracle Database-Softwareversion die Verbindung zu einer Multitenant-Containerdatenbank oder stellen sie her.

Quelle: <https://www.oracle.com/de/database/technologies/multitenant.html>

Prinzipiell: Out-Of-Place Patching

- Datenbank kann im laufenden Betrieb gepatcht werden (kein Upgrade Modus)
- Ausnahme: OJVM nur mit Einschränkung

Konkretes Projekt

- RAC Rolling Upgrade
- Standby First Apply

Optimizer Fixes werden über das Package `dbms_optim_bundle` aktiviert

- `GETBUGSFORBUNDLE` → Zeigt die Patchids an, die damit gefixt sind
- `ENABLE_OPTIM_FIXES` → Aktiviert die Fixes

Funktioniert auch für Oracle 12.1

```
SQL> execute dbms_optim_bundle.enable_optim_fixes('ON','BOTH','YES');
```

Funktioniert nicht für Standby Datenbank!

- Entweder nach Switchover / Failover explizit ausführen
- Besser:

```
SQL> ALTER SYSTEM SET _fix_control='patch_nr1','patch_nr2',...
```

RAC Rolling Upgrade

- Alle Anwendungen kommen damit klar
 - Einige Anwendungen benötigen einen Reconnect

Standby Rolling Upgrade

- Anwendungen bekommen Fehler
 - Switchoverzeit zu lang (5 Minuten)
 - Services werden erst sehr spät gestartet

Installation RU 190716

- Datapatch geht in einer CDB schief → Kein Problem, ist ja die Standby Datenbank
- Switchover auf die ursprüngliche Primart → Datapatch funktioniert

Aber:

- Durch den fehlerhaften datapatch sind etliche PDBs im Zustand „Restricted“
- Fehlerbehebung: CLOSE PDB / OPEN PDB
- Unschön, weil die Anwendung dadurch für ca. 15 Minuten nicht zur Verfügung stand

Lösung:

- Immer Status der PDBs abfragen (show pdbs im SQL*Plus)



Ressourcenverwaltung

DEVK

Jede PDB eigene SGA

- SGA_TARGET
- DB_CACHE_SIZE
- SHARED_POOL_SIZE
- PGA_AGGREGATE_TARGET
- PGA_AGGREGATE_LIMIT

```
SQL> ALTER SESSION SET CONTAINER=suzanne;
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
sga_max_size	big integer	2512M
sga_min_size	big integer	0
sga_target	big integer	0

```
SQL> ALTER SYSTEM SET sga_target=500M;
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
sga_max_size	big integer	2512M
sga_min_size	big integer	0
sga_target	big integer	500M

```
SQL> ALTER SESSION SET CONTAINER=CDB$ROOT;
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
sga_max_size	big integer	2512M
sga_min_size	big integer	0
sga_target	big integer	2512M

Einschränkung von Datenbank Optionen

- Partitioning
- Database Queuing (warum auch immer)

Datenbank Features

- AWR
- OS_ACCESS
- usw.

SQL Befehle

- ALTER DATABASE
- ALTER PLUGGABLE DATABASE
- ALTER SYSTEM
- ALTER SESSION

Bestimmte Optimizerparameter werden auf PDB-Ebene gesetzt

- optimizer_index_cost_adj
- optimizer_features_enabled

Lockdown Profiles werden nicht verwendet

- zu kompliziert

Ressourcenlimitierung wird nicht verwendet

- Statt dessen Services nur auf bestimmten Knoten
- Aktive Nutzung aller Server
- Zusätzlicher Server nur für Statistik Datenbank



Data Guard

DEVK

Rechenzentrum 1 (Primary):

```
CDB: /u03/oradata/MC01/MC01_RZ/datafile  
PDB: /u03/oradata/MC01/pdbshare0/MC01_RZ/54AA536C9C006B7E053261E10AC2048/datafile
```

Rechenzentrum 2 (Standby):

```
CDB: /u03/oradata/MC01/MC01_RR/datafile  
PDB: /u03/oradata/MC01/pdbshare0/MC01_RZ/54AA536C9C006B7E053261E10AC2048/datafile
```

Nicht brauchbar, weil nicht konsistent!

Kein Einsatz von Oracle Managed Files mit Multitenant und Data Guard

Lösung:

- Keine Oracle Managed Files (Dateinamen voll qualifizieren) verwenden
- Trennung von PDBs und CDB für separate Mount-Points

```
→CDB Datafiles: /u03/oradata/<DBNAME>/CDB
```

```
→PDB Datafiles: /u03/oradata/<DBNAME>/pdbshared/<PDBNAME>/datafile
```

Anlegen von PDB schlägt auf der Standby Seite fehl!

- Verzeichnis wird auf der primären Seite angelegt
- Auf der Standby Seite fehlt diese

Daher Skripting erforderlich

Generell: Tempfiles werden auf der Standby Seite nicht angelegt

Wird derzeit nicht eingesetzt

Gründe:

- Ca. 80% aller PDBs noch Version 12.1
- Man will in der Regel nicht die gesamte PDB zurücksetzen

→ Flashback auf die Standby Datenbank

→ Snapshot Copy von der NetApp

- Aufwändig: Welche User / Schemata gehören zur Anwendung
- Schwierig: Zugriff auf andere Schemata / Anwendungen
- Herausfordernd: Arbeiten mit der Multitenant Architektur
- Flexibel: „Mal eben eine PDB anlegen“
- Robust: Multitenant mit RAC und Data Guard
- Ausbaufähig: Multitenant und Data Guard

DOAG Konferenz und Ausstellung 19 bis 23. November

- 20.11 11:00 Uhr Oracle 18 XE „Mehr als eine Spieldatenbank“
- 20.11 13:00 Uhr Geliebter Feind „Der DBA und DevOps“
- 21.11 12:00 Uhr Datenbank-Monitoring für das nächste Jahrzehnt
- 21.11 13:00 Uhr Alternativen zur Enterprise Edition der Oracle DB
- 22.11 ganztägig PostgreSQL für Oracle DBAs

Frankfurter IT Tage 9. bis 12. Dezember in Frankfurt

- 12.12 09:00 Uhr Alternativen zur Oracle Enterprise Edition

DOAG Noon2Noon „Oracle auf VMware“ 23. bis 24 Januar 2020

München