



Datenbank-Virtualisierung & Instant Cloning ... und was macht Oracle 12c

Sebastian Winkler
CarajanDB GmbH

- **CarajanDB**
- **Klonen**
- **Möglichkeiten und Herausforderungen**
- **Oracle 11g CloneDB**
- **Oracle 12c Snapshot Copy**
- **Delphix**

- **Experten mit über 20 Jahren Oracle Erfahrung**
 - Firmensitz in Erftstadt bei Köln
- **Spezialisten für**
 - Backup & Recovery
 - Hochverfügbarkeit
 - Healthchecks
 - Performance Optimierung
 - Einsatz von Oracle Standard Edition
 - Oracle in virtuellen Umgebungen und in der Cloud
 - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
- **Schulung und Workshops (Oracle, Toad)**



Der Begriff "Klon" kommt aus dem Griechischen und heißt "Sprössling". Mit "klonen" bezeichnet man die künstliche Entwicklung von genetisch identischen Nachkommen eines einzigen Lebewesens. Anders als bei der normalen Zeugung, bei der die Erbanlagen des Vaters und der Mutter vererbt werden und jeder Nachkomme genetisch anders und einzigartig ist, **sind Klone exakte "Kopien" eines einzigen Wesens.**

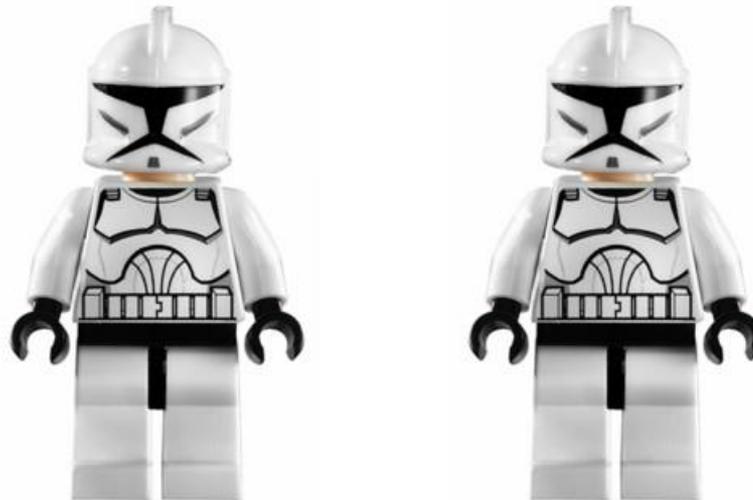
...

In Deutschland ist das Klonen von Menschen verboten.

Quelle: www.hanisauland.de



- Ziel sind möglichst identische Funktionen
- DB als möglichst identische Kopie der Produktionsumgebung für Test, Entwicklung, Reporting, Q&A ...



- Ziel sind möglichst identische Funktionen
- DB als möglichst identische Kopie der Produktionsumgebung für Test, Entwicklung, Reporting, Q&A ...
- Meist mehrere Kopien für unterschiedliche Anforderungen



Datenbanken klonen - Warum?

- Neue Datenbank(en) als Kopie einer bereits existierenden Datenbank erstellen
- Arbeiten mit „richtigen“ Daten aus der Produktion für Test- und Entwicklungsumgebungen
- Test von Backup & Recovery
- Aufbau Standby-Datenbank
- Performance Tests



Datenbanken klonen - Warum?

- Neue Datenbank(en) als Kopie einer bereits existierenden Datenbank erstellen
- Arbeiten mit „richtigen“ Daten aus der Produktion für Test- und Entwicklungsumgebungen
- Test von Backup & Recovery
- (Aufbau Standby-Datenbank)
- (Performance Tests)



- **Möglichkeiten eine Datenbank zu klonen bspw.:**
 - **Manuelles kopieren der Dateien und Neuerstellung der Datenbank**
 - **Data-Pump-Ex- und Import**
 - **Drittanbieter Lösungen**
 - **RMAN DUPLICATE (seit Version 10g)**



- Kopieren einer Oracle Datenbank aus einem Backup
- Ab Version 11.2.0.1 auch als direkte Kopie (FROM ACTIVE DATABASE)
- Ablauf:
 - Quelle:

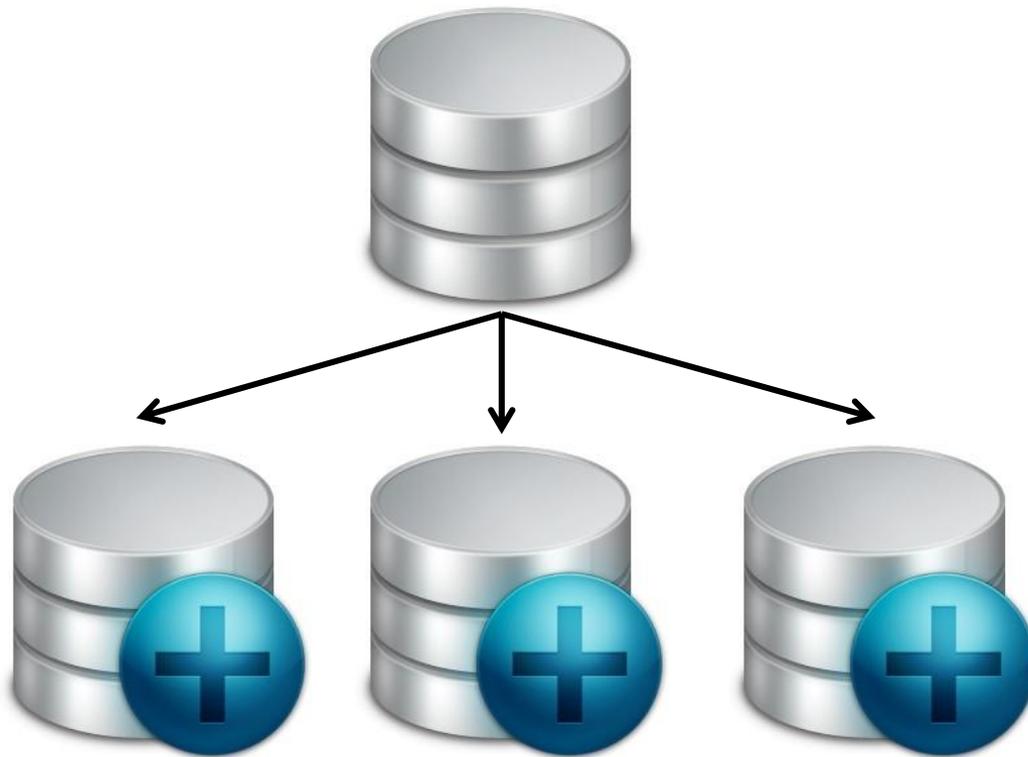
```
SQL> CREATE pfile='/tmp/initPROD.ora' from spfile;  
% scp /tmp/initPROD.ora entwicklung:/u01/app/.../dbs/initKLON.ora
```

- Ziel:

```
SQL> startup nomount  
% lsnrctl start listener  
% rman target sys/manager@PROD auxiliary sys/manager@KLON  
RMAN> DUPLICATE TARGET DATABASE FROM ACTIVE DATABASE  
NOFILENAMECHECK;
```

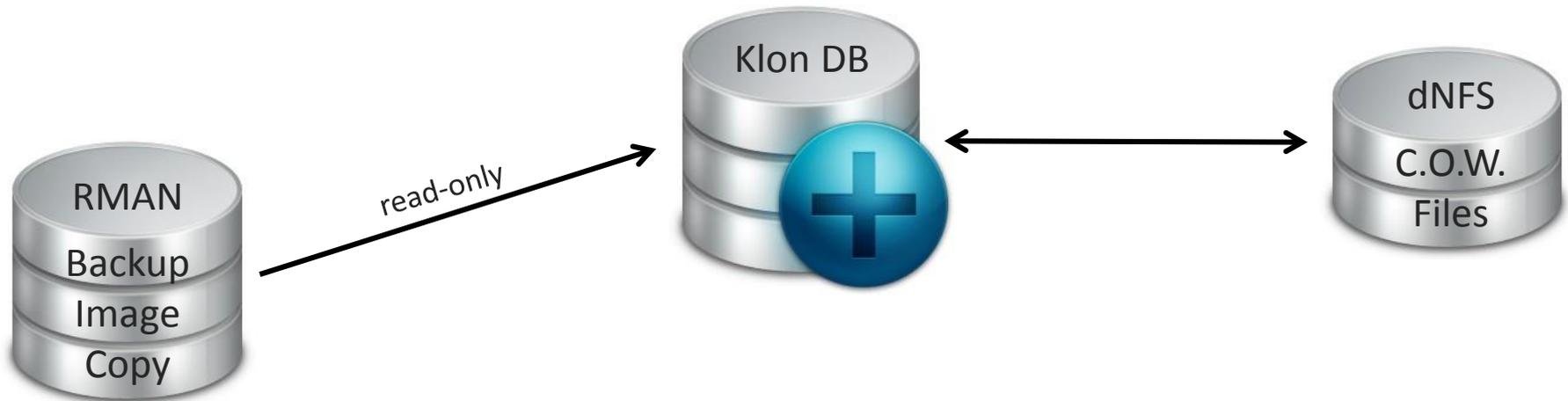
- **Viele Methoden erfordern eine Kopie bzw. Wiederherstellung der Daten auf der neuen Umgebung, was wiederum:**
 - zusätzlichen Speicherplatz und
 - Zeit zum ausführen der Kopie braucht
- **Anforderungen**
 - Kopie soll möglichst alle Daten enthalten
 - Erstellung soll möglichst schnell gehen
 - Eventuell müssen mehrere Kopien erstellt werden
 - Plattenplatz ist endlich!
 - DBAs haben andere Sorgen!

- Wie macht CloneDB das Klonen schneller, spart Zeit und Speicherplatz?



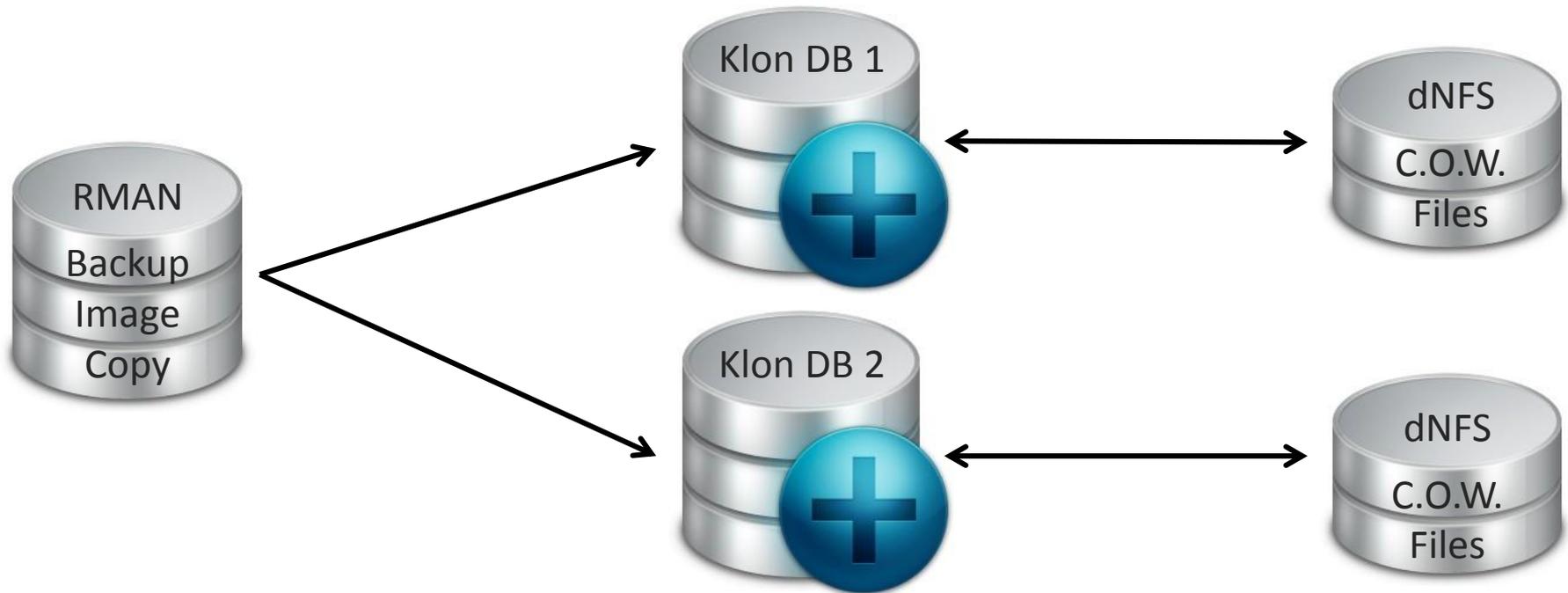
- In Version 11.1 fügt Oracle der Datenbank einen eigenen Direct NFS (dNFS) Client hinzu
- Mit dem 11.2.0.2 Patch Set wird dNFS - CloneDB als neues Feature hinzugefügt
- CloneDB nutzt das Copy-On-Write-Verfahren (C.O.W., Kopieren beim Schreiben) zum klonen
- Grundidee C.O.W.: Kopie wird erst dann angefertigt wenn das Original verändert wird – die Kopie erfolgt zunächst „virtuell“

CloneDB - Copy-On-Write



- RMAN Image-Kopie der Datafiles als Quelle - read-only
- NFS Storage für die C.O.W.-Datafiles
- Nur geänderte Blöcke werden in die C.O.W.-Files geschrieben

CloneDB - Copy-On-Write



- Mehrere Klone greifen auf die gleichen Quell-Daten zu
- modifizierte Blöcke werden in eigene C.O.W.-Datafiles geschrieben
- Platz- und Zeitersparnis ab dem 2ten Klon

- **Speicherplatzersparnis**
 - da jeder Klon die selben Quelldaten referenzieren kann
- **Zeitersparnis**
 - Backup-Quelldaten müssen zunächst nur einmal erstellt werden
 - Nach erstem Setup, ist jeder weitere Klon schnell aufgesetzt
- **Dokumentiert**
 - in My Oracle Support
 - Doc ID 1210656.1 - Clone your dNFS Production Database for Testing

- Für den Test haben wir folgende Server

NFS Server:

Name : nas1
IP Address : 172.16.0.2
NFS Share : /u01/nfshare/clonedb/klon

DB Server Produktion:

Name : prod
IP Address : 172.16.0.171
ORACLE_HOME: /u01/app/oracle/product/11.2.0.4/db_1
ORACLE_SID : PROD

DB Server CloneDB:

Name : klon
IP Address : 172.16.0.172
ORACLE_HOME: /u01/app/oracle/product/11.2.0.4/db_1
ORACLE_SID : KLON
NFS Mount : /u01/app/oracle/oradata/TEST

- **Verzeichnis für die Copy-On-Write Files erstellen**

```
# mkdir -p /u01/nfshare/clonedb/klon
```

- **Eintragen des Verzeichnisses in “/etc/exports”**

```
/u01/nfshare/clonedb/klon  
*(rw,sync,no_wdelay,insecure,insecure_locks,no_root_squash)
```

- **Sicherstellen, dass der NFS Service läuft**

```
# chkconfig nfs on  
# service nfs restart
```

- Erstellen eines Image-Kopie Backups der Datenbank. Die Kopie kann ebenfalls auf einem Share (CIFS/NFS) abgelegt werden oder auf dem lokalen Filesystem des Klon-Servers

```
$ rman target = /
```

```
RMAN> configure controlfile autobackup off;  
RMAN> sql 'alter database begin backup';  
RMAN> backup as copy database format '/u03/orabackup/prod/%U';  
RMAN> sql 'alter database end backup';
```

- **BACKUP AS COPY** erstellt eine Byte-für-Byte Kopie der Datenbank

- **Erstellen einer Kopie des PFILE der Produktion als SYS User**

```
$ sqlplus / as sysdba
```

```
SQL> CREATE PFILE='/u03/orabackup/prod/initKLON.ora' FROM SPFILE;
```

- **Bearbeiten des PFILE: alle Referenzen zu Original-SID und Datenbankname ändern entsprechend der Klon-Datenbank**
 - Ab Datenbankversion 11.2.0.3 muss zusätzlich der Parameter CLONEDB im PFILE für den KLON auf TRUE gesetzt werden.

```
*.clonedb=true
```

- **Einschalten von Direct NFS** im ORACLE_HOME auf dem CloneDB Server

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk dnfs_on
```

- **Erstellen aller notwendigen Verzeichnisse** zum Mounten des NFS und um die Instanz zu starten

```
# su - oracle
$ mkdir -p $ORACLE_BASE/oradata/KLON
$ mkdir -p $ORACLE_BASE/fast_recovery_area/KLON
$ mkdir -p $ORACLE_BASE/admin/KLON/adump
$ mkdir -p $ORACLE_BASE/admin/KLON/dpdump
```

- **Hinzufügen des NFS mount in “/etc/fstab”**
 - Damit wird das Verzeichnis beim Reboot automatisch verbunden

```
nas1:/u01/nfshare/clonedb/test /u01/app/oracle/oradata/KLON nfs
rw,bg,hard,nointr,tcp,vers=3,timeo=600,rsize=32768,wsiz=32768,actimeo=0 0 0
```

- **Mounten des NFS-Verzeichnisses als „oracle“ OS User**
 - ggf. Ownership und Permission prüfen

```
# mount /u01/app/oracle/oradata/KLON
```

- **Setzen der notwendigen Umgebungsvariablen**

```
$ export ORACLE_SID=KLON
$ export MASTER_COPY_DIR=/u03/orabackup/prod
$ export CLONE_FILE_CREATE_DEST=/u01/app/oracle/oradata/KLON
$ export CLONEDB_NAME=KLON
```

- **clonedb.pl Skript ausführen**

- Das erzeugte und bearbeitete PFILE angeben und die Namen für die zu erzeugenden Skripte angeben (in \$ORACLE_HOME/rdbms/install/clonedb.pl oder MOS 1210656.1)

```
$ perl clonedb.pl /u01/app/oracle/product/11.2.0.4/db_1/dbs/initKLON.ora
crtddb.sql dbren.sql
```

- **Mit SQL*Plus als SYS anmelden und die Skripte ausführen**

```
$ sqlplus / as sysdba
Connected to an idle instance.

@crtddb.sql
@dbren.sql
```

```
SQL> @crtddb.sql
SQL> SET FEEDBACK 1
SQL> SET NUMWIDTH 10
SQL> SET LINESIZE 80
SQL> SET TRIMSPOOL ON
SQL> SET TAB OFF
SQL> SET PAGESIZE 100
SQL>
SQL> STARTUP NOMOUNT PFILE=/u01/app/oracle/oradata/KLON/initKLON.ora
ORACLE instance started.

Total System Global Area  839282688 bytes
Fixed Size                  2257880 bytes
Variable Size              494930984 bytes
Database Buffers          339738624 bytes
Redo Buffers                2355200 bytes
SQL> CREATE CONTROLFILE REUSE SET DATABASE KLON RESETLOGS
 2      MAXLOGFILES 32
 3      MAXLOGMEMBERS 2
 4      MAXINSTANCES 1
 5      MAXLOGHISTORY 908
 6 LOGFILE
 7  GROUP 1 '/u01/app/oracle/oradata/KLON/KLON_log1.log' SIZE 100M BLOCKSIZE 512,
 8  GROUP 2 '/u01/app/oracle/oradata/KLON/KLON_log2.log' SIZE 100M BLOCKSIZE 512
 9 DATAFILE
10  '/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-EXAMPLE_FNO-5_0eom14gg',
11  '/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-SYSAUX_FNO-2_0bom14rf',
12  '/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-SYSTEM_FNO-1_0aom14po',
13  '/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-UNDOTBS1_FNO-3_0com14ss',
14  '/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-USERS_FNO-4_0dom14tl'
15 CHARACTER SET WE8DEC;

Control file created.
```

```
SQL> @dbren.sql
SQL> REM Rename clonedb files to local copy from backup
SQL> declare
  2  begin
  3  dbms_dnfs.clonedb_renamefile('/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-EXAMPLE_FNO-5_0eom14qg' ,
'/u01/app/oracle/oradata/KLON/ora_data_KLON0.dbf');
  4  dbms_dnfs.clonedb_renamefile('/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-SYSAUX_FNO-2_0bom14rf' ,
'/u01/app/oracle/oradata/KLON/ora_data_KLON1.dbf');
  5  dbms_dnfs.clonedb_renamefile('/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-SYSTEM_FNO-1_0aom14po' ,
'/u01/app/oracle/oradata/KLON/ora_data_KLON2.dbf');
  6  dbms_dnfs.clonedb_renamefile('/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-UNDOTBS1_FNO-3_0com14ss' ,
'/u01/app/oracle/oradata/KLON/ora_data_KLON3.dbf');
  7  dbms_dnfs.clonedb_renamefile('/u03/orabackup/prod/data_D-PROD_I-4197193115_TS-USERS_FNO-4_0dom14t1' ,
'/u01/app/oracle/oradata/KLON/ora_data_KLON4.dbf');
  8  end;
  9  /
PL/SQL procedure successfully completed.

SQL>
SQL> REM open clone database
SQL> WHENEVER SQLERROR CONTINUE
SQL> alter database open resetlogs;

Database altered.

SQL>
SQL> REM create any temp tablespaces needed
SQL> drop tablespace TEMP;
drop tablespace TEMP
*
ERROR at line 1:
ORA-12906: cannot drop default temporary tablespace
```

```
SQL> create temporary tablespace TEMP;
create temporary tablespace TEMP
*
ERROR at line 1:
ORA-01543: tablespace 'TEMP' already exists
```

```
SQL>
```

- **ERROR TEMP Tablespace**

da das Skript diesen Punkt nicht ordnungsgemäß abwickelt:

- **Wiederherstellen des Datafiles für den TEMP-Tablespace**

```
SQL> select * from dba_temp_files;
```

```
no rows selected
```

```
SQL> alter tablespace TEMP add tempfile '/u01/app/oracle/oradata/KLON/temp01.dbf' size 200M;
```

```
Tablespace altered.
```

- Sobald die Skripte vollständig ausgeführt sind erhält man einen funktionierenden Klon

```
SELECT t.name AS tbs,  
       d.name AS dbf,  
       status  
FROM   v$datafile d  
       JOIN v$tablespace t ON t.ts# = d.ts#  
ORDER BY t.name;
```

TBS	DBF	STATUS
EXAMPLE	/u01/app/oracle/oradata/KLON/ora_data_KLON0.dbf	ONLINE
SYSAUX	/u01/app/oracle/oradata/KLON/ora_data_KLON1.dbf	ONLINE
SYSTEM	/u01/app/oracle/oradata/KLON/ora_data_KLON2.dbf	SYSTEM
UNDOTBS1	/u01/app/oracle/oradata/KLON/ora_data_KLON3.dbf	ONLINE
USERS	/u01/app/oracle/oradata/KLON/ora_data_KLON4.dbf	ONLINE

- Die Copy-On-Write Location enthält jetzt nur die geänderten Blöcke und keine vollständige Kopie der Datafiles

```
$ du -k /u03/orabackup/prod/*
162452    /u03/orabackup/prod/data_D-PROD_I-4197193115_TS-EXAMPLE_FNO-5_0eom14qg
614412    /u03/orabackup/prod/data_D-PROD_I-4197193115_TS-SYSAUX_FNO-2_0bom14rf
716812    /u03/orabackup/prod/data_D-PROD_I-4197193115_TS-SYSTEM_FNO-1_0aom14po
312332    /u03/orabackup/prod/data_D-PROD_I-4197193115_TS-UNDOTBS1_FNO-3_0com14ss
5128      /u03/orabackup/prod/data_D-PROD_I-4197193115_TS-USERS_FNO-4_0dom14tl
```

```
$ du -k /u01/app/oracle/oradata/KLON/*.dbf
20        /u01/app/oracle/oradata/KLON/ora_data_KLON0.dbf
55        /u01/app/oracle/oradata/KLON/ora_data_KLON1.dbf
89        /u01/app/oracle/oradata/KLON/ora_data_KLON2.dbf
201       /u01/app/oracle/oradata/KLON/ora_data_KLON3.dbf
18        /u01/app/oracle/oradata/KLON/ora_data_KLON4.dbf
9851     /u01/app/oracle/oradata/KLON/KLON_ctl.dbf
```

- **Ermöglicht sehr schnelles Klonen**
- **Spart potentiell viel Speicherplatz ab der zweiten Klon-DB**
 - Je mehr geänderte Blöcke desto geringer die Einsparung
 - Nutzt man Image-Kopien als Backup?
- **Setzt mindestens Version 11.2.0.2 voraus**
- **Benötigt NFS Storage als Copy-On-Write Location**
 - Problem im gegebenen IT-Umfeld?
- **Besonders geeignet für kurzlebige Klone, die vor allem für Funktionstest benötigt werden**
 - unterschiedliche Storage-Technologien machen Performance-Tests problematisch

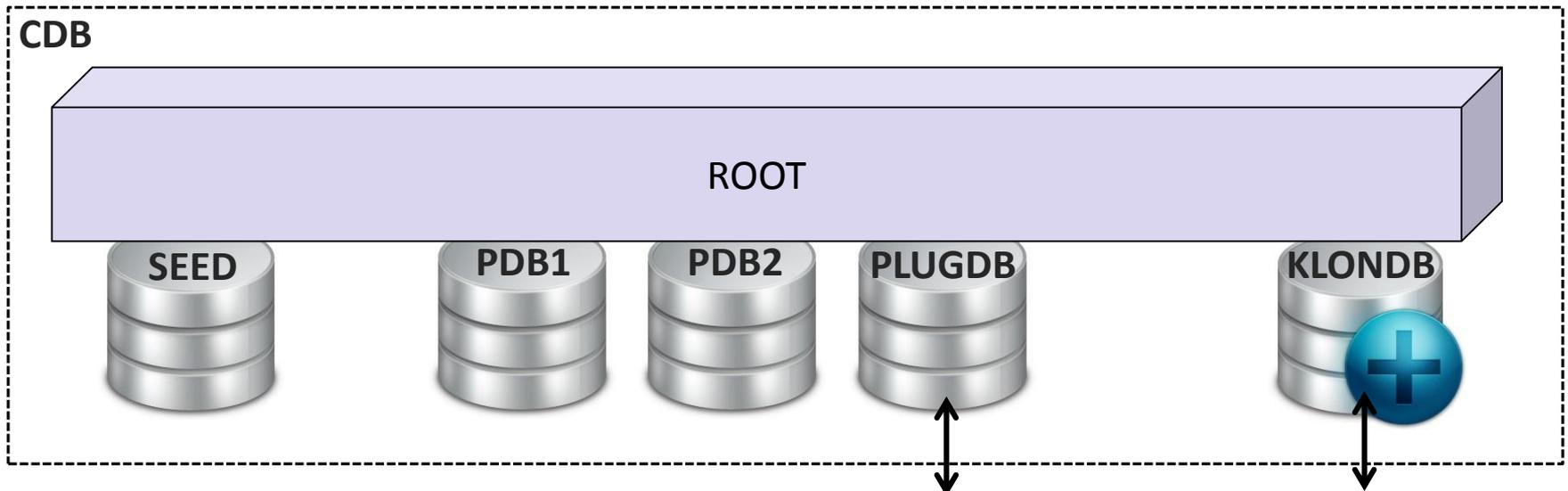
CloneDB - Oracle 12c

- ... und was macht 12c ?
- CloneDB ist unter 12c ebenfalls nutzbar aber ...

- ... und was macht 12c ?
- **CloneDB ist unter 12c ebenfalls nutzbar aber nur mit einer NON-CDB**
 - Keine Pluggable Database Features

- Erstellen einer PDB durch Klonen einer existierenden PDB

```
SQL> create pluggable database KLONDB from PLUGDB
```



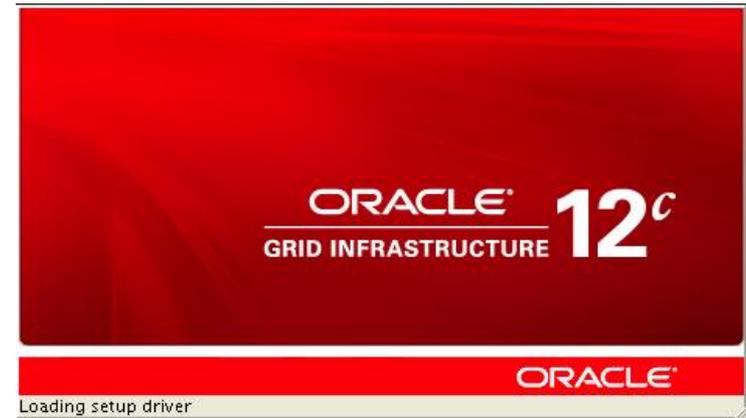
- Ausgangsproblem:



12c Snapshot Copy Feature

- **Lösung: Snapshot Copy Feature**
- **Voraussetzungen**
 - Multitenant-Architektur
 - Storage muss ACFS oder Direct NFS Client sein
- **Einschränkungen**
 - Quell-PDB muss für das Klonen im Open Read-Only Mode sein
 - Quell-PDB und Klon-PDB müssen in der selben CDB liegen
 - Kein unpluggen oder droppen der Quell-PDB, solange der Klon existiert

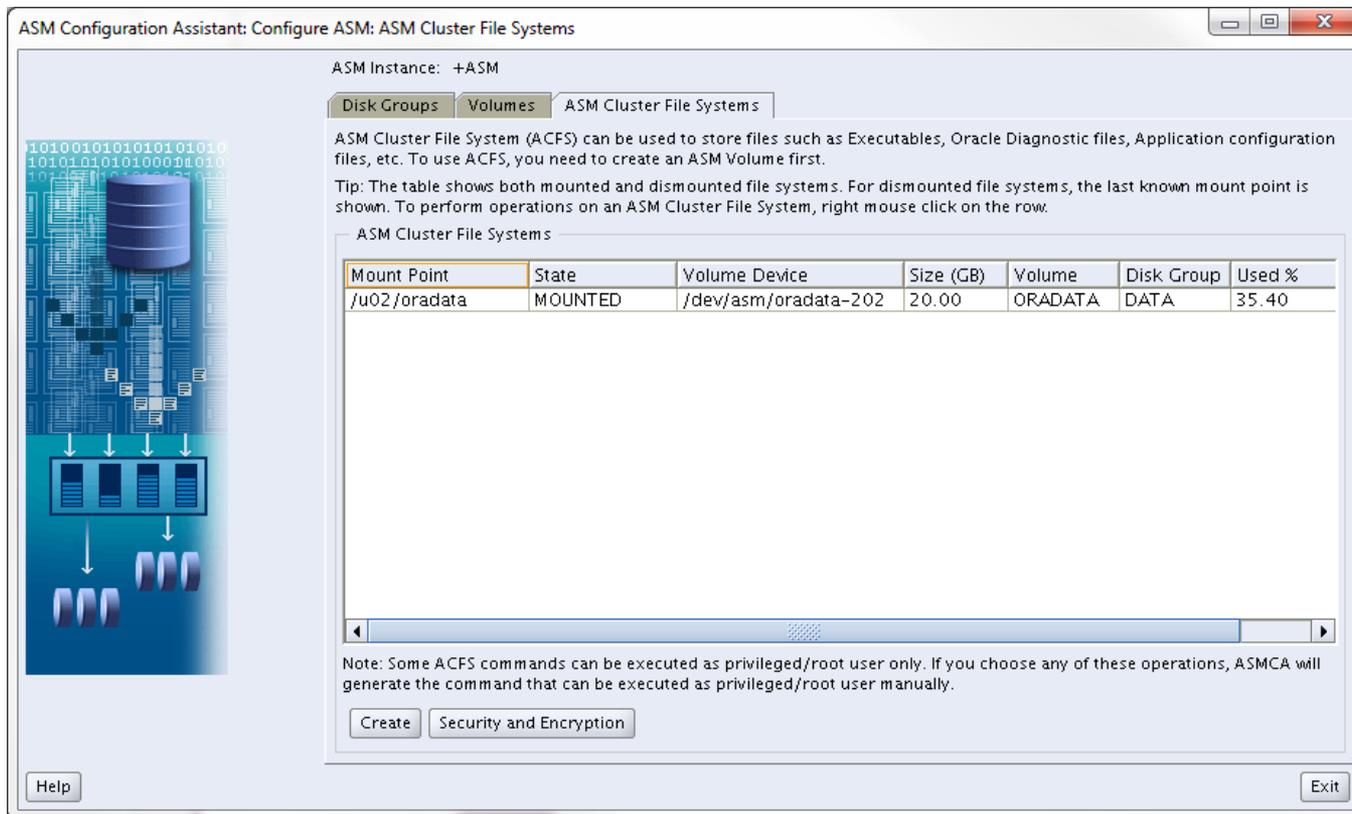
- **Erstellen des ACFS** (Oracle Automatic Storage Management Cluster File Systems)
- **Voraussetzung**
 - Oracle 12c Grid Infrastructure



- **Ab Oracle 12c darf Datenbank im ACFS liegen**

ASM Cluster File System

- per ASMCA
 - *ASM Volume erstellen und Datafile-Verzeichnis mounten*



ASM Configuration Assistant: Configure ASM: ASM Cluster File Systems

ASM Instance: +ASM

Disk Groups | Volumes | **ASM Cluster File Systems**

ASM Cluster File System (ACFS) can be used to store files such as Executables, Oracle Diagnostic files, Application configuration files, etc. To use ACFS, you need to create an ASM Volume first.

Tip: The table shows both mounted and dismounted file systems. For dismounted file systems, the last known mount point is shown. To perform operations on an ASM Cluster File System, right mouse click on the row.

ASM Cluster File Systems

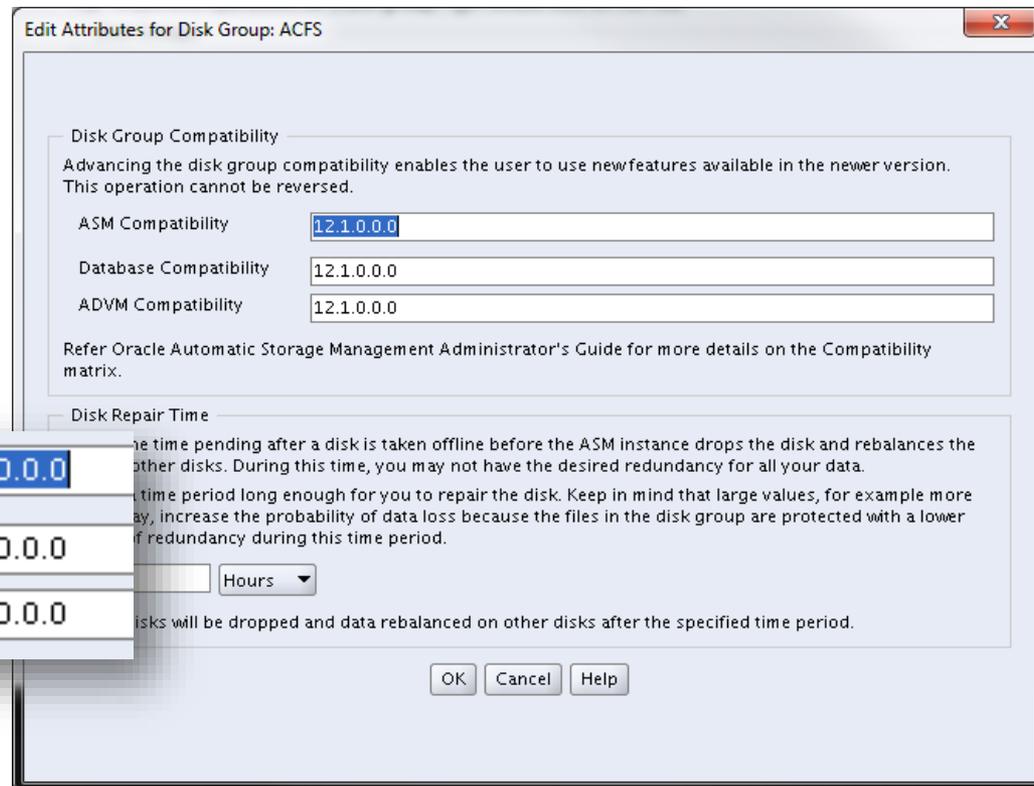
Mount Point	State	Volume Device	Size (GB)	Volume	Disk Group	Used %
/u02/oradata	MOUNTED	/dev/asm/oradata-202	20.00	ORADATA	DATA	35.40

Note: Some ACFS commands can be executed as privileged/root user only. If you choose any of these operations, ASMCA will generate the command that can be executed as privileged/root user manually.

Create Security and Encryption

Help Exit

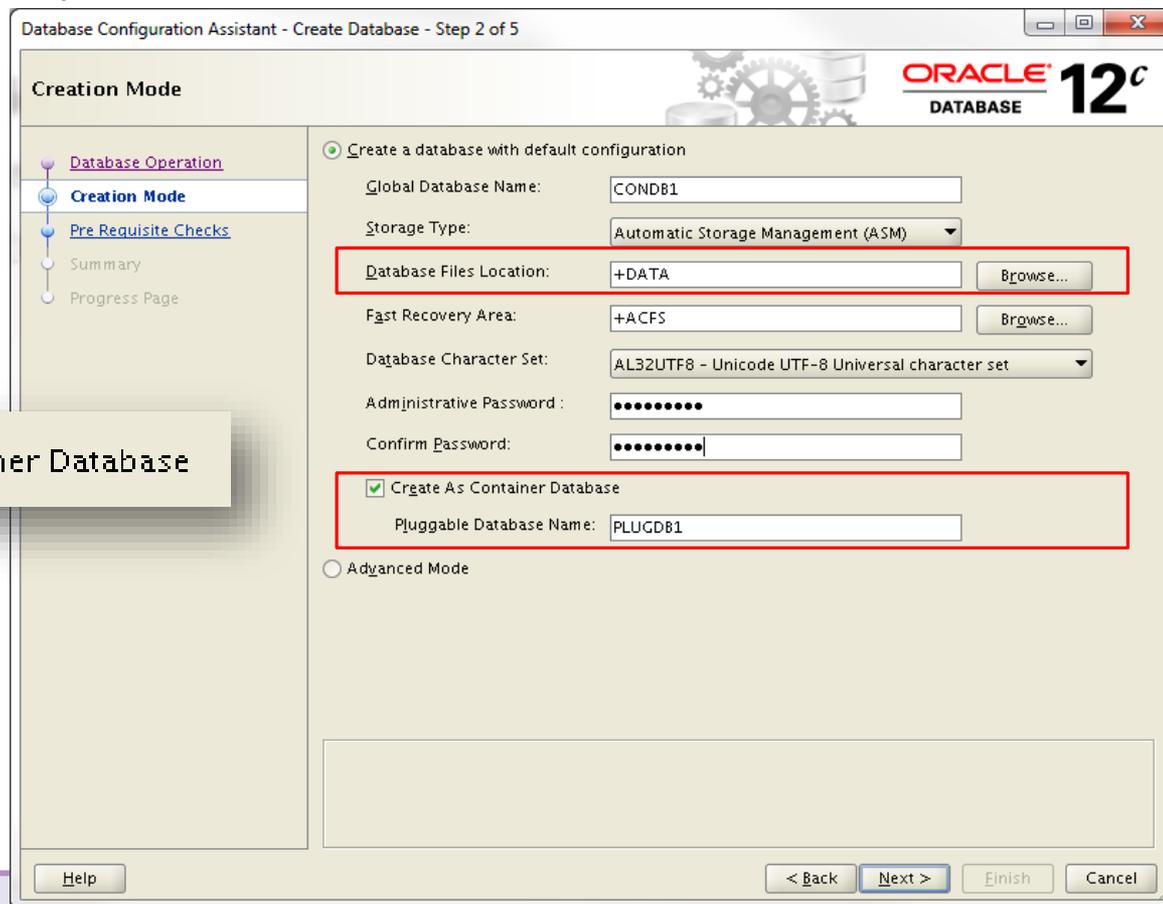
- **Disk Group Compatibility**
 - *Kompatibilität auf aktuelle Version 12.1.0.0.0*



ASM Compatibility	12.1.0.0.0
Database Compatibility	12.1.0.0.0
ADVM Compatibility	12.1.0.0.0

Multitenant Datenbank aufsetzen

- Container Datenbank und Pluggable Datenbank
 - Erstellen per DBCA



Database Configuration Assistant - Create Database - Step 2 of 5

Creation Mode

Database Operation

Creation Mode

Pre Requisite Checks

Summary

Progress Page

Create a database with default configuration

Global Database Name: CONDB1

Storage Type: Automatic Storage Management (ASM)

Database Files Location: +DATA

Fast Recovery Area: +ACFS

Database Character Set: AL32UTF8 - Unicode UTF-8 Universal character set

Administrative Password:

Confirm Password:

Create As Container Database

Pluggable Database Name: PLUGDB1

Advanced Mode

Help

Create As Container Database

- *Datenbank in den Read Only Mode versetzen*

```
SQL> ALTER PLUGGABLE DATABASE PLUGDB1 CLOSE IMMEDIATE;  
SQL> ALTER PLUGGABLE DATABASE PLUGDB1 OPEN READ ONLY;
```

- *Jetzt kann die Klon-Datenbank erstellt werden*

```
SQL> create pluggable database KLONDB from PLUGDB1  
      Storage ( Maxsize unlimited  
              Max_shared_temp_size unlimited)  
      File_name_convert = NONE  
      Path_prefix = NONE  
      Tempfile reuse  
      snapshot copy;
```

```
SQL> ALTER PLUGGABLE DATABASE KLONDB OPEN READ WRITE;
```

- *Wiederherstellen der Quell-Datenbank nach dem Klonen nicht vergessen*

```
SQL> ALTER PLUGGABLE DATABASE PLUGDB1 CLOSE IMMEDIATE;  
SQL> ALTER PLUGGABLE DATABASE PLUGDB1 OPEN READ WRITE;
```

- *Anzeigen der Snapshots mit ACFSUTIL*

```
# acfsutil snap info /u02/oradata
```

```
snapshot name:                E98F6E0720E036E4E043181E10ACBE06
RO snapshot or RW snapshot:   RW
parent name:                  /u02/oradata
snapshot creation time:       Fri Oct 25 14:11:37 2013
```

```
snapshot name:                EA576455FF753723E043181E10AC6E75
RO snapshot or RW snapshot:   RW
parent name:                  /u02/oradata
snapshot creation time:       Mon Nov 4 12:08:07 2013
```

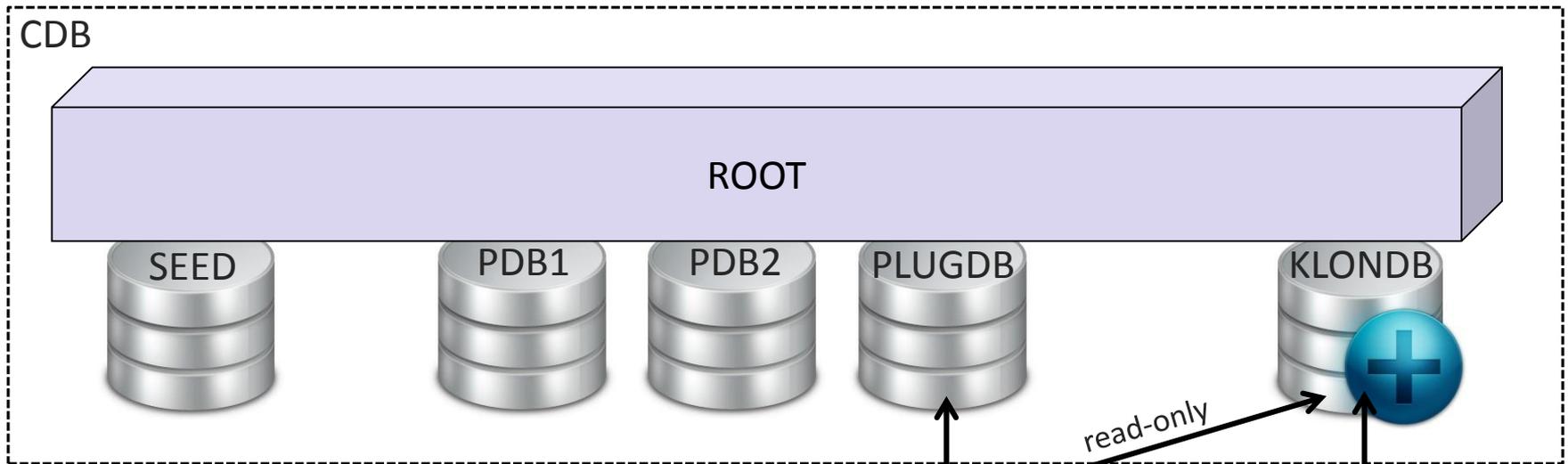
```
number of snapshots: 2
snapshot space usage: 2339905536
```

- *Erstellen und Löschen des Klons inklusive Daten in wenigen Minuten*

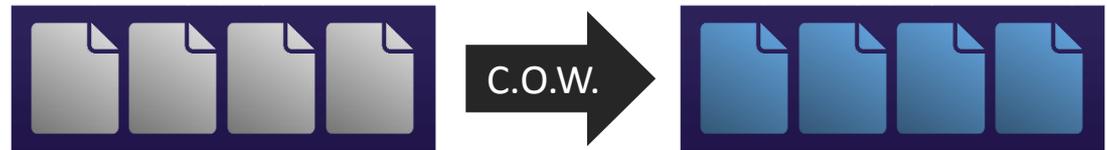
```
SQL> drop pluggable database KLONDB including datafiles;
```

- Erstellen einer PDB durch Klonen einer existierenden PDB im ACFS

```
SQL> create pluggable database KLONDB from PLUGDB snapshot copy;
```

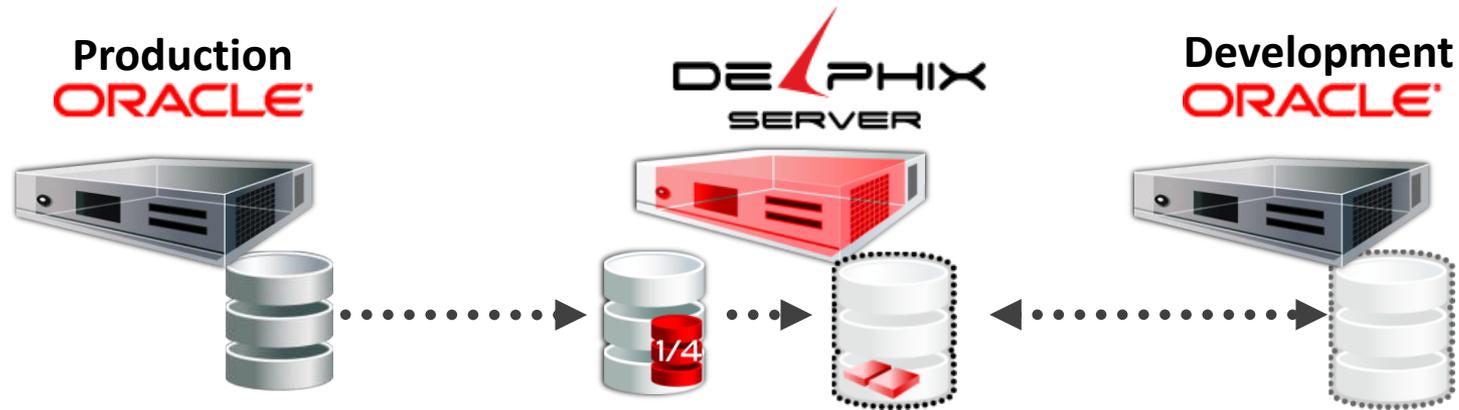


- Lösung C.O.W.-Files:



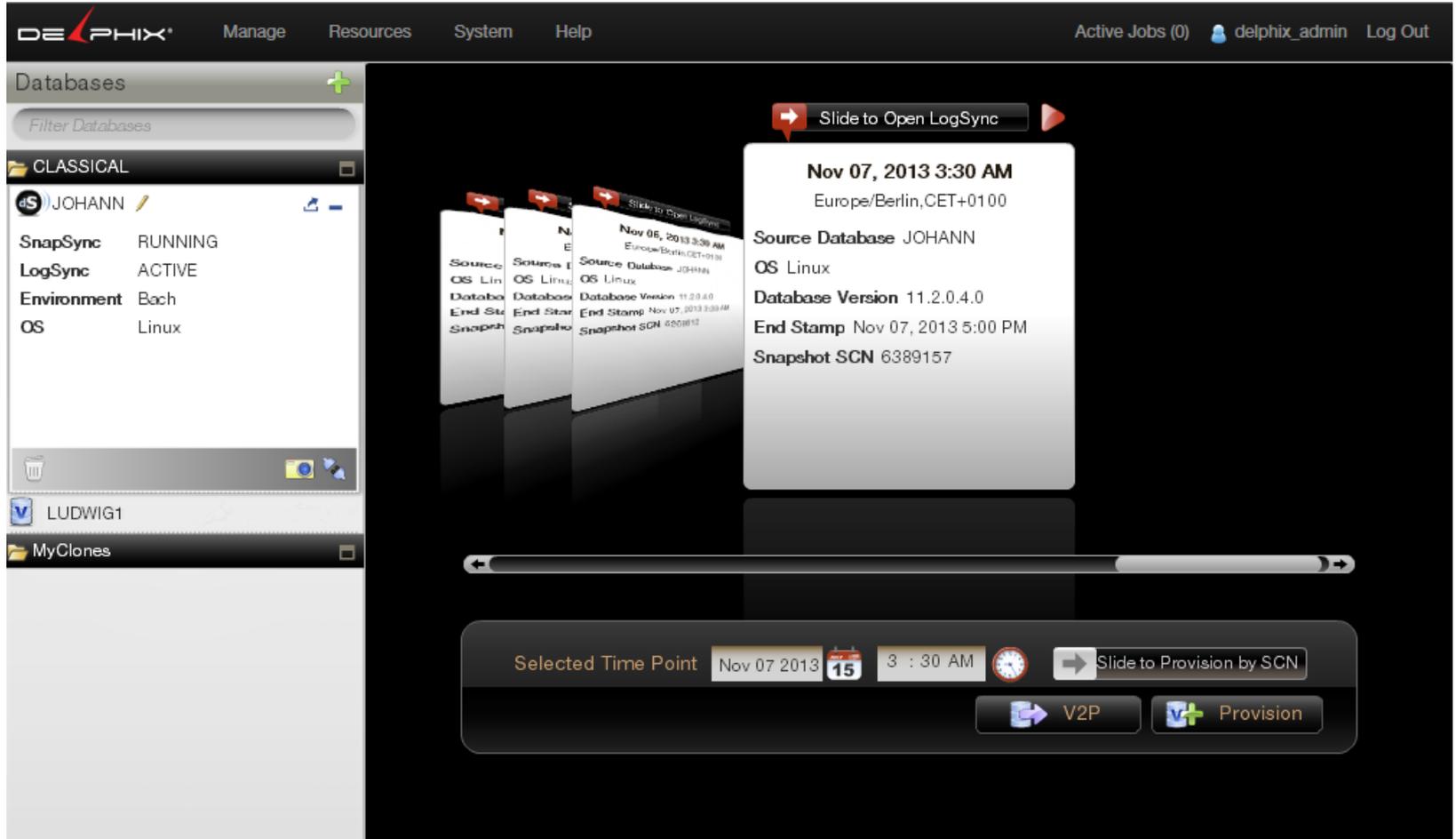
- **Einstiegshürden in Form von**
 - 12c Multitenant Datenbank
 - Bereitstellung von Storage über ASM oder NFS
 - Grid Infrastructure
- **Einmal eingerichtet, ermöglicht es**
 - sehr einfaches und schnelles Klonen
- **Was in dieser Version noch fehlt**
 - die Quell-PDB muss „Read-Only“ geöffnet sein, was ein schnelles Kopieren behindert
- **Keine inkrementellen Snapshots möglich**

- Clonen von Oracle oder SQL-Server Datenbanken
- Basiert auf eigenem Filesystem (DxFS)
- Management über Delphix Appliance



- **Startpunkt: Full Backup**
 - Komprimiertes RMAN Full Backup nach SBT_TAPE
- **Inkrementelle Backups für neue Synchronisationspunkte**
 - Komprimiertes inkrementelles RMAN Backup nach SBT_TAPE
- **Zwischenstufen über archivierte Redolog-Dateien**
- **Datenbank-Verzeichnisse werden auf dem Zielsystem als NFS Shares zur Verfügung gestellt**

Management Console



The screenshot displays the Delphix Management Console interface. The top navigation bar includes 'DEPHIX', 'Manage', 'Resources', 'System', and 'Help'. On the right, it shows 'Active Jobs (0)', the user 'delphix_admin', and a 'Log Out' button.

The left sidebar, titled 'Databases', contains a search filter and a list of databases. The 'CLASSICAL' folder is expanded, showing a database named 'JOHANN'. Its properties are listed as follows:

SnapSync	RUNNING
LogSync	ACTIVE
Environment	Bach
OS	Linux

Below the database list, there are icons for trash, refresh, and other actions. The 'LUDWIG1' folder and 'MyClones' folder are also visible.

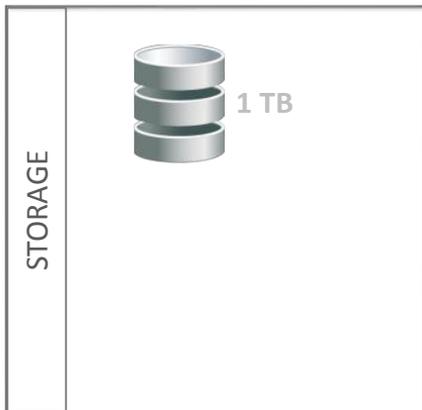
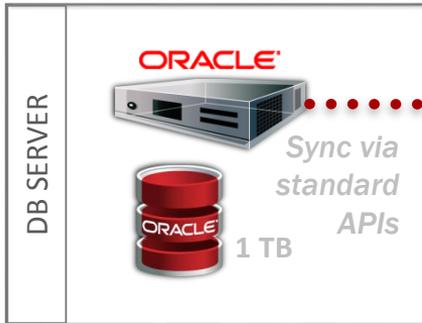
The main content area shows a series of snapshots for the 'JOHANN' database. A 'Slide to Open LogSync' control is visible. A detailed view of a snapshot is shown, with the following information:

- Nov 07, 2013 3:30 AM**
Europe/Berlin,CET+0100
- Source Database JOHANN
- OS Linux
- Database Version 11.2.0.4.0
- End Stamp Nov 07, 2013 5:00 PM
- Snapshot SCN 6389157

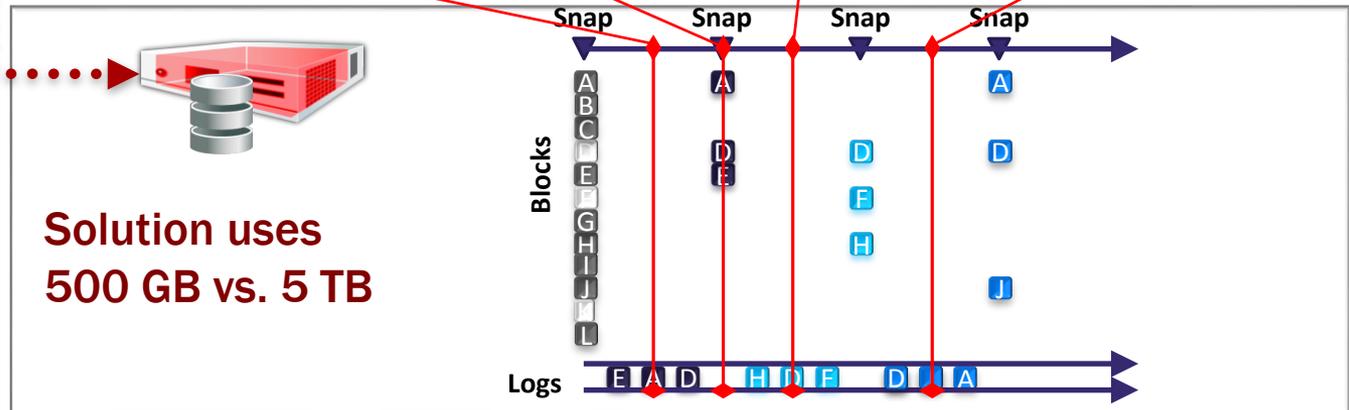
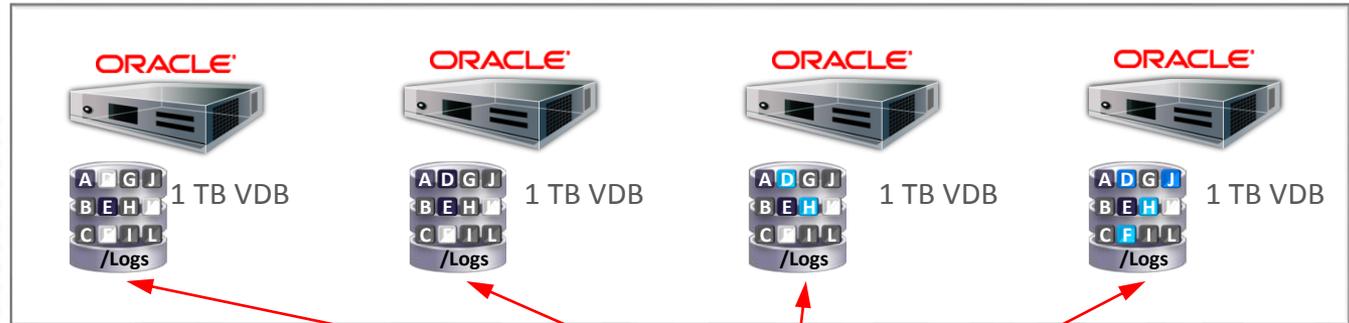
At the bottom, a 'Selected Time Point' section allows users to choose a date and time (Nov 07 2013, 3:30 AM) and includes a 'Slide to Provision by SCN' control. Below this are buttons for 'V2P' and 'Provision'.

Schnelle Provisionierung

Produktion



Test und Entwicklung



- **Delphix Server**
 - VMware (ESX Server 3.x, vSphere 4.x, vSphere 5.x)
 - VM Guest Container Profile: Solaris 10 64-bit
 - Virtual CPUs: mind. 8 vCPUs
 - Memory: mind. 16 GB
- **Source und Target Servers**
 - 9.2.0.8, 10g R2, 11g R1, 11g R2
 - Solaris, RedHat, OEL, SUSE, AIX, HP/UX

- **Eintägiger Workshop zu Oracle 12c**
 - Überblick über viele neue Funktionen
 - Multitenant Database
 - Information Lifecycle Management
 - High Availability
 - Datenbank Administration
 - Application Development
 - Database Express

- **Weitere Informationen: info@carajandb.com**



Fragen

www.carajandb.com
sebastian.winkler@carajandb.com