ORACLE

Migrate to Multitenant with Refreshable Clones using AutoUpgrade

OUGN 2023

Daniel Overby Hansen
Senior Principal Product Manager

DANIEL OVERBY HANSEN

Senior Principal Product Manager
Cloud Migration

https://dohdatabase.com

dohdatabase

@dohdatabase

Episode 1

Release and Patching Strategy

105 minutes – Feb 4, 2021

Episode 2

AutoUpgrade to Oracle Database 19c

115 minutes – Feb 20, 2021

Episode 3

Performance Stability, Tips and Tricks and Underscores

120 minutes – Mar 4, 2021

Episode 4

Migration to Oracle Multitenant

120 minutes – Mar 16, 2021

Episode 5

Migration Strategies – Insights, Tips and Secrets

120 minutes – Mar 25, 2021

Episode 6

Move to the Cloud – Not only for techies

115 minutes – Apr 8, 2021

Recorded Web Seminars

https://MikeDietrichDE.com/videos

More than 30 hours of technical content,
on-demand, anytime, anywhere

Introduction

Multitenant Architecture

Starting with Oracle Database 21c, installation of
non-CDB Oracle Database architecture is no longer supported.

The non-CDB architecture was deprecated in Oracle Database 12c.
It is desupported in Oracle Database 21c.

Database 21c, Upgrade Guide, chapter 10

Oracle Database 19c

Convert

Oracle Database 23c

```
--Use up to 3 PDBs without
--a license for Multitenant option

alter system set max_pdbs=3;
```

How to

MIGRATE

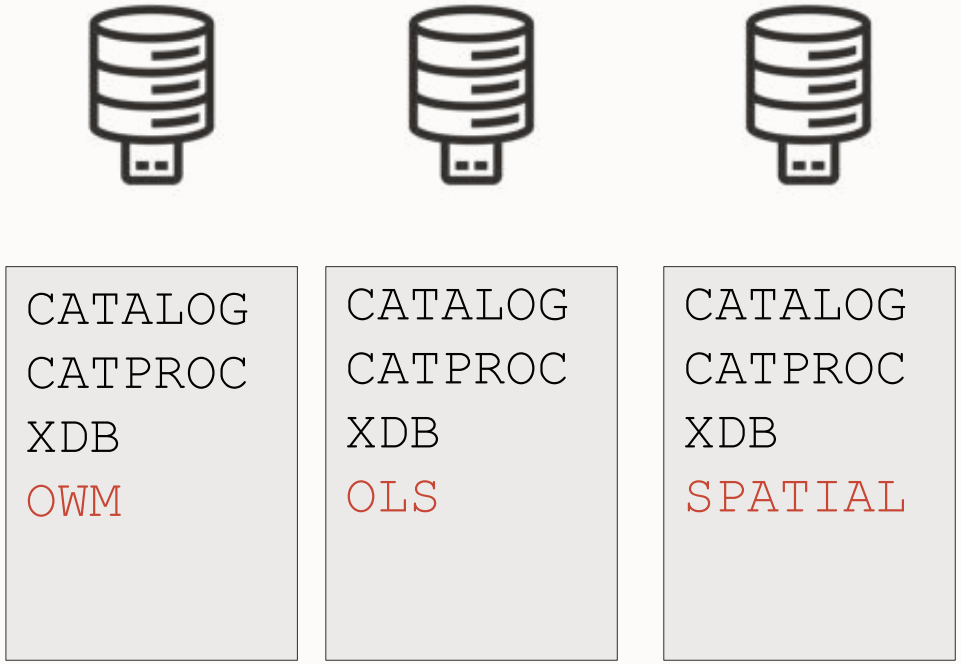
to multitenant architecture

Creating the container database

Always create the CDB with AL32UTF8 character set

Allows PDBs with different character set

Multitenant | Components



Install as many components as required.
But no more than that.

```
--Always set compatible to the default of a release
--Use three digits only
alter system set compatible='19.0.0' scope=spfile;


--Should I change compatible when patching?
--No, this is a bad idea
alter system set compatible='19.17.0' scope=spfile;
```

Multitenant | Silent Compatible Change

- On plug-in, a PDB adopts `COMPATIBLE` of CDB Silently and without confirmation
- Changing `COMPATIBLE` is irreversible
- Changing `COMPATIBLE` will prevent
  - Plugging back into original CDB
  - Downgrading to previous release

`compatible=12.2.0`

`compatible=19.0.0`

Keep `COMPATIBLE` at the default setting

Keep the same `COMPATIBLE` setting throughout your database landscape

Multitenant | Additional Information

Blog posts:

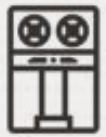
- https://mikedietrichde.com/2018/08/08/creating-cdbs-non-cdbs-with-less-options/
- https://mikedietrichde.com/2017/07/11/always-create-custom-database/
- https://mikedietrichde.com/2017/07/26/remove-clean-components-oracle-11-2-12-2/

Non-CDB to PDB conversion is irreversible

What are your fallback options?

PDB Conversion

FALLBACK



Backup / restore

Ensure you have a recent backup and requires time to restore and recover

Copy data files

Requires time and disk space to hold a copy of the data files

PDB Conversion

FALLBACK

Backup / restore

Ensure you have a recent backup and requires time to restore and recover

Copy data files

Requires time and disk space to hold a copy of the data files

Refreshable clone

Requires ~~time and~~ disk space to hold a copy of the data files

Requires Oracle Database 12.2 or newer

Refreshable Clone

CREATE

Create PDB from non-CDB over a database link

REFRESH

Apply redo from non-CDB to keep PDB up-to-date

OUTAGE

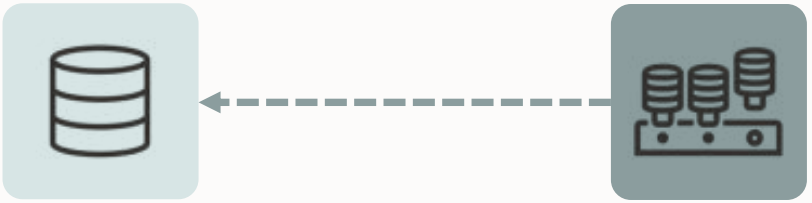
Disconnect users and refresh PDB for the last time

CONVERT

To become a proper PDB, it must be converted

Refreshable Clone

Source non-CDB	Target CDB

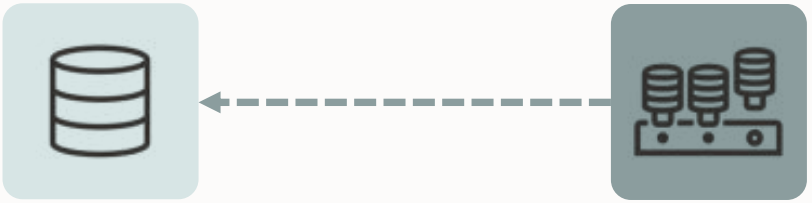


```
CREATE USER dblinkuser
       IDENTIFIED BY ... ;

GRANT CREATE SESSION,
      CREATE PLUGGABLE DATABASE,
      SELECT_CATALOG_ROLE TO dblinkuser;

GRANT READ ON sys.enc$ TO dblinkuser;
```

```
CREATE DATABASE LINK CLONEPDB
       CONNECT TO dblinkuser
       IDENTIFIED BY ...
       USING 'noncdb-alias';
```

Refreshable Clone

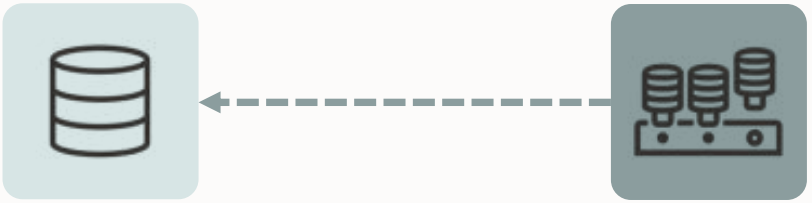
Source non-CDB	Target CDB



```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=NONCDB1
upg1.target_cdb=CDB1
upg1.source_dblink.NONCDB1=CLONEPDB
upg1.target_pdb_name.NONCDB1=PDB1
```

Refreshable Clone

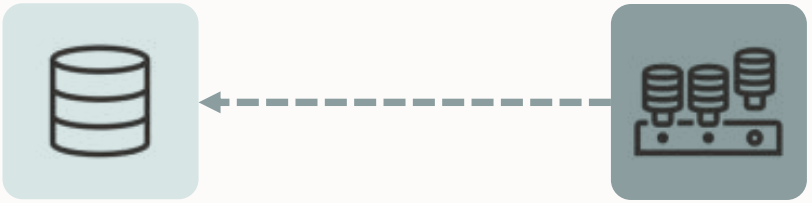
Source non-CDB	Target CDB



```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=NONCDB1
upg1.target_cdb=CDB1
upg1.source_dblink.NONCDB1=CLONEPDB 300
upg1.target_pdb_name.NONCDB1=PDB1
```

Refreshable Clone

Source non-CDB	Target CDB



```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=NONCDB1
upg1.target_cdb=CDB1
upg1.source_dblink.NONCDB1=CLONEPDB 300
upg1.target_pdb_name.NONCDB1=PDB1
upg1.start_time=29/04/2023 02:00:00
--Specify relative start time
--upg1.start_time=+1h30m
```

Refreshable Clone


```
autoupgrade.jar ... -mode deploy
```

```
upg1.start_time=29/04/2023 02:00:00
```

DEMO

- Upgrade to Oracle Database 19c
- Migrate from non-CDB to PDB
- Using Refreshable Clone PDBs

Watch on YouTube

The source non-CDB stays intact to allow fallback

Works for unplug-plug upgrades as well


Zürcher
Kantonalbank

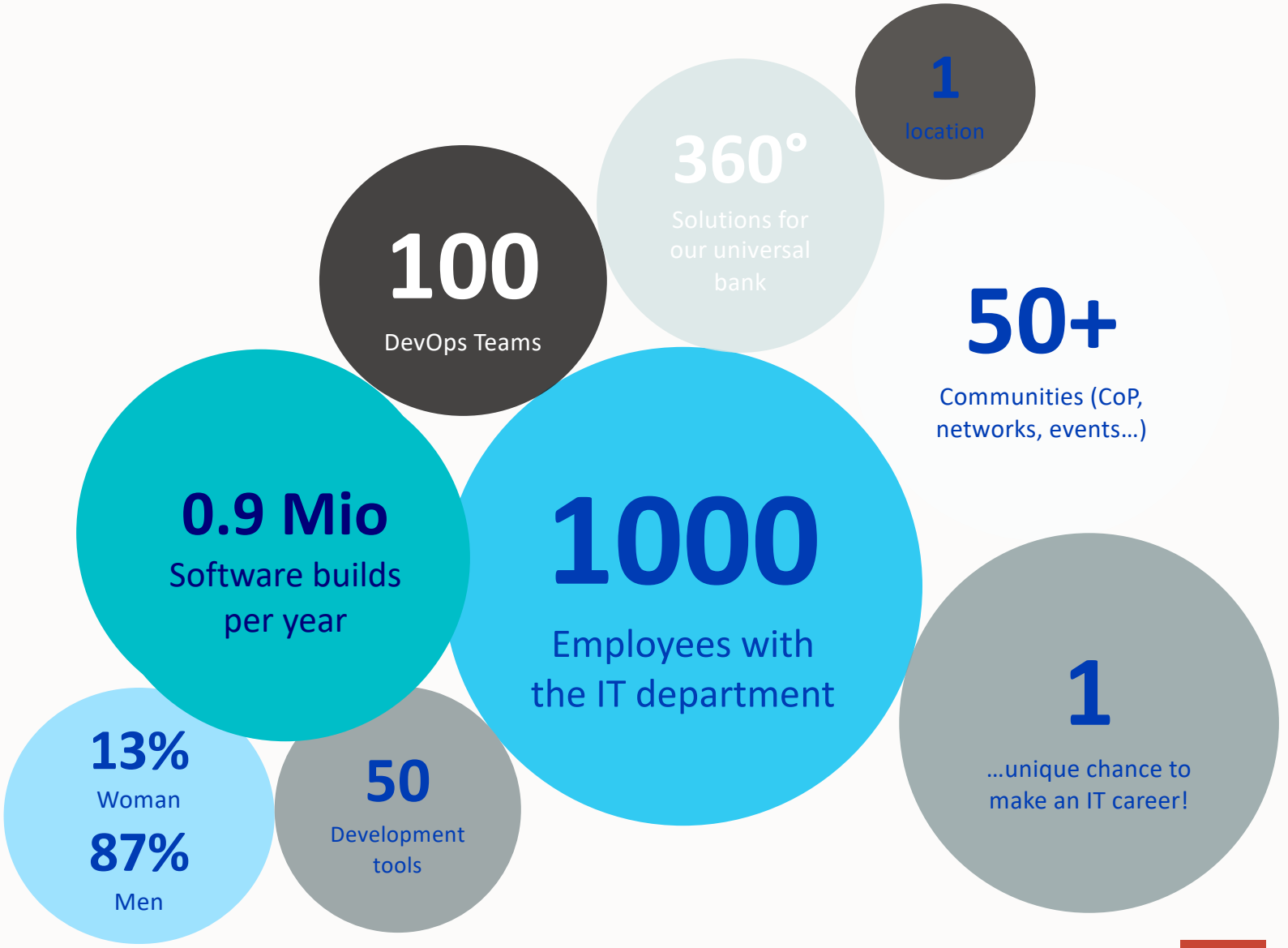
we have been a reliable partner for almost 150 years

- the bank for the **people of Zurich** since 1870
- with over **5’100 employees** one of the biggest employers in the canton of Zurich
- a globally networked **full-service bank** with strong regional and local roots

IT made in Chreis 5

650

+applications like eBanking, Mobile Banking and ZKB Twint (mobile payment)



about me

- 45 years old, married, 8-year-old daughter
- working with Oracle since 1997
- DBA at Zürcher Kantonalbank since 2011
- now DBA & Solution architect for Oracle topics
- Organizer of Oracle Beer ZH Meetup Events

www.linkedin.com/in/christian-marquardt-ch

@krischanCH

https://oracle-beer.blog

current situation

- Oracle databases on old OS and on Oracle Exadata
- till end of 2023: migrating all databases from old OS to Exadata compute nodes
- starting 2024: consolidating all databases to Multitenant
  and Oracle's next-generation long-term release

- planned solution: **AutoUpgrade** utility

test setup

3 test databases

- Exadata X6-2 compute node
- 7 Storage-Cells (2x X6-2L / 3x X7-2L / 2x X8-2L)
- Oracle RDBMS 19.15
- no additional options

Source	Size / GB
TEST40 (108)	165
TEST42 (107)	555
TEST41 (106)	18’496

preparation

create the DB-Link-User

```
create user dblinkuser identified by Oracle_4UOracle_4U;
```

grant permissions

```
grant CONNECT, RESOURCE, CREATE PLUGGABLE DATABASE,
select_catalog_role to dblinkuser;
grant all on sys.enc$ to dblinkuser;
```

create DB link on destination DB

```
create database link test42.domain connect to dblinkuser identified
by oracle_4uoracle_4u using ‘test42.domain’;
```

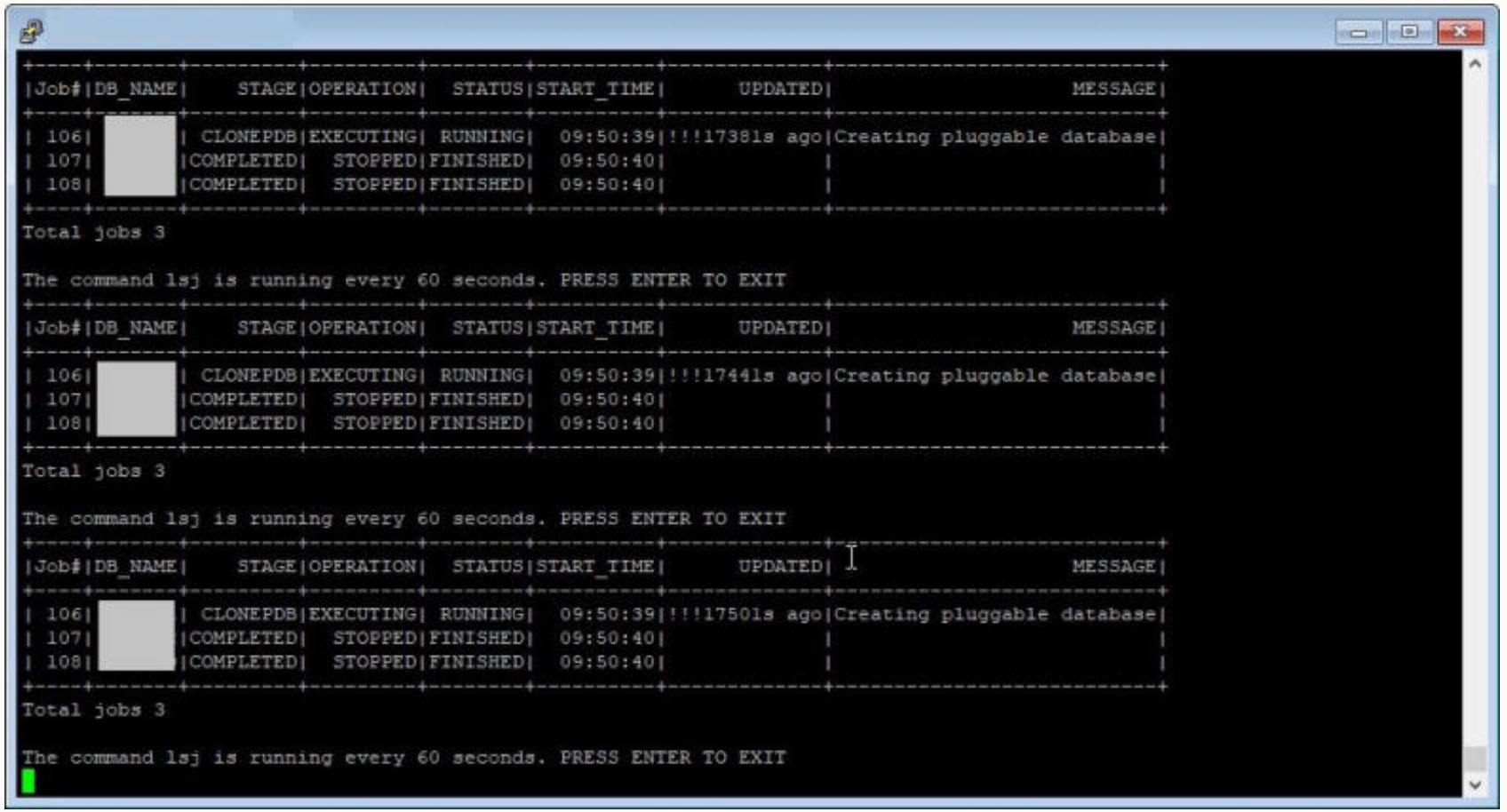

migration - the config file

```
> cat migrateTESTDBs.cfg
test40.log_dir=/home/oracle/logs_TEST40
global.autoupg_log_dir=/home/oracle/AULOG
test40.source_home=/app/oracle/product/19.15.0.0.220419
test40.target_home=/app/oracle/product/19.15.0.0.220419
test40.sid=TEST40
test40.target_cdb=CDBTST02
test40.source_dblink.TEST40=TEST40.domain 3600
test40.target_pdb_name.TEST40=TEST40X
test40.target_pdb_copy_option.TEST40=file_name_convert=NONE
test42.log_dir=/home/oracle/logs_TEST42
test42.source_home=/app/oracle/product/19.15.0.0.220419
test42.target_home=/app/oracle/product/19.15.0.0.220419
test42.sid=TEST42
test42.target_cdb=CDBTST02
test42.source_dblink.TEST42=TEST42.domain 3600
test42.target_pdb_name.TEST42=TEST42X
test42.target_pdb_copy_option.TEST42=file_name_convert=NONE
test41.log_dir=/home/oracle/logs_TEST41
test41.source_home=/app/oracle/product/19.15.0.0.220419
test41.target_home=/app/oracle/product/19.15.0.0.220419
test41.sid=TEST41
test41.target_cdb=CDBTST02
test41.source_dblink.TEST41=TEST41.domain 3600
test41.target_pdb_name.TEST41=TEST41X
test41.target_pdb_copy_option.TEST41=file_name_convert=NONE
```

migration - autoupgrade start

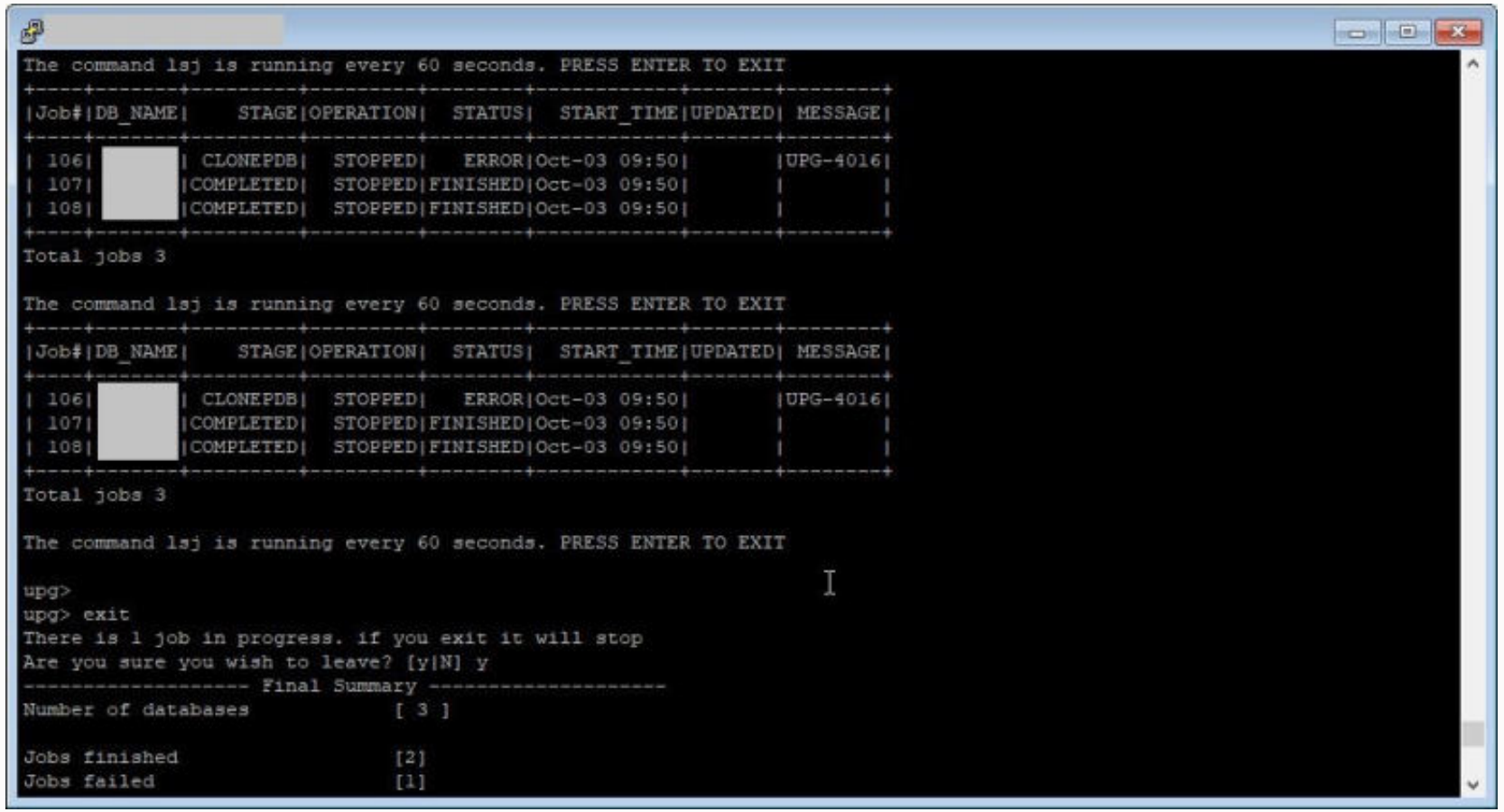
```
> java -jar /app/oracle/product/19.15.0.0.220419/rdbms/admin/autoupgrade.jar -config [redacted].cfg -mode deploy
AutoUpgrade 22.5.220913 launched with default internal options
Processing config file ...
```

migration – in progress



Source	Runtime/Min
TEST40 (108)	26
TEST42 (107)	ongoing
TEST41 (106)	ongoing

migration – “finished”



Source	Runtime/Min
TEST40 (108)	26
TEST42 (107)	226 (~3.5h)
TEST41 (106)	1770 (29h)

errors during migration

1st error: archivelogs on source DB missing during refresh of the 18TB DB

solution: restore archivelogs needed from backup

2nd error: user profile with idle_time defined; an idle session was killed because of idle_time setting

solution: different profile for sourcedb_user configured and migration restarted

3rd error: ORA-600 [4293]

still under investigation

Tip: think about this kind of errors before starting a long-running migration

summary

comfortable to use (does everything automatically)

without extra license costs

very stable

easy syntax

perfect for pre-migration tests

Pro Tips and Details

Get there faster and smarter

Multitenant Conversion | How Long Does It Take?

DOWNTIME

Requires downtime.

RUNTIME

Typically 10-30 minutes.
Depends mostly on the number of objects.
Not the physical size of the database.

PROCESS

Only need to run it once.
The process is irreversible.
Rerunnable in case of errors.

Ensure archive logs are available on disk during migration

Cloning

CLONING

AutoUpgrade uses `CREATE PLUGGABLE DATABASE` statement with `PARALLEL` clause which clones the database using multiple parallel processes

PARALLEL

Based on system resources and current utilization the database automatically determines a proper parallel degree

TRANSFER

A new file transfer protocol that can bypass several layers in the database to achieve very high transfer rates

NETWORK

Watch out for network saturation. Optionally, use 3rd party tools like traffic control (`tc`) to limit network usage

```
SQL> select message, sofar, totalwork,time_remaining as remain, elapsed_seconds as ela
     from v$session_longops
     where opname='kpdbfCopyTaskCbk' and sofar != totalwork;
```

MESSAGE	SOFAR	TOTALWORK	REMAIN	ELA
kpdbfCopyTaskCbk: /u01/app/oracle/oradata/CDB2/EDA 3: 643199 out of 1310720 Blocks done	643199	1310720	134	129
kpdbfCopyTaskCbk: /u01/app/oracle/oradata/CDB2/EDA 3: 443007 out of 1310720 Blocks done	443007	1310720	213	109
kpdbfCopyTaskCbk: /u01/app/oracle/oradata/CDB2/EDA 3: 436351 out of 1310720 Blocks done	436351	1310720	216	108
kpdbfCopyTaskCbk: /u01/app/oracle/oradata/CDB2/EDA 3: 370431 out of 1310720 Blocks done	370431	1310720	256	101

```
SQL> select sql_text
     from v$sql s, v$session_longops l
     where s.sql_id=l.sql_id and l.opname='kpdbfCopyTaskCbk';
```

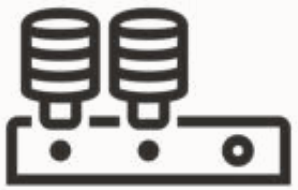
```
SQL_TEXT
/* SQL Analyze(256,0) */ SELECT /*+PARALLEL(4) NO_STATEMENT_QUEUING  */ * FROM X$KXFTASK /*kpdbfParallelCopyOrMove,PDB_FILE_COPY*/
```

If CDB is configured with Data Guard
special attention is needed

```
--Default config file setting
--PDB is not fully created on standbys
--Recovery is needed

upg1.manage_standbys_clause=standbys=none
```

Source
Non-CDB

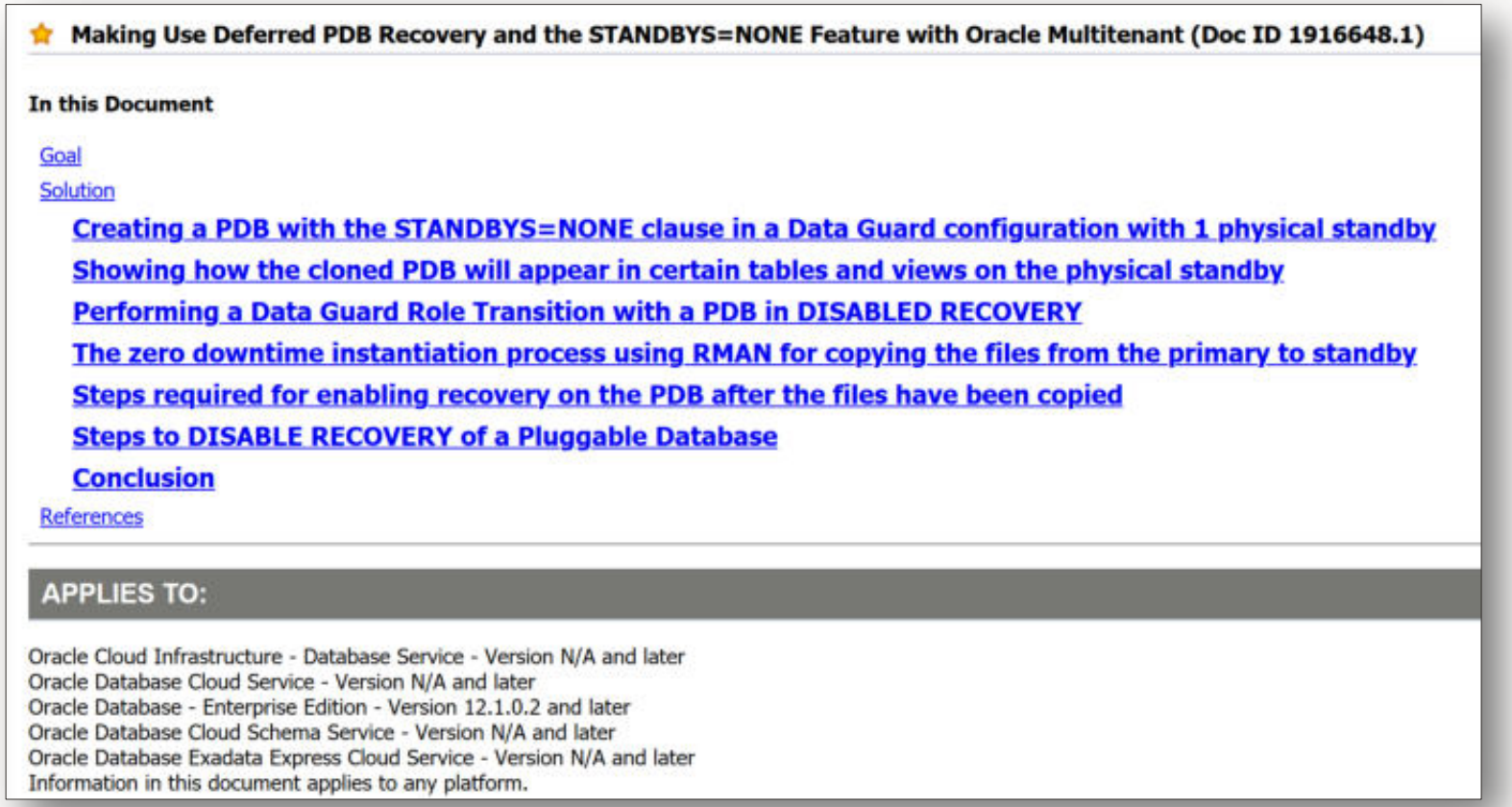
Target
Primary



```
RMAN> restore pluggable database ...
      from service ... ;

SQL> alter pluggable database
     enable recovery;
```

PDB created
Datafiles missing

Data Guard | Plug-in



Making Use Deferred PDB Recovery and the STANDBYS=NONE Feature with Oracle Multitenant (Doc ID 1916648.1)

In this Document

APPLIES TO:

Oracle Cloud Infrastructure - Database Service - Version N/A and later
Oracle Database Cloud Service - Version N/A and later
Oracle Database - Enterprise Edition - Version 12.1.0.2 and later
Oracle Database Cloud Schema Service - Version N/A and later
Oracle Database Exadata Express Cloud Service - Version N/A and later
Information in this document applies to any platform.

Making Use Deferred PDB Recovery and the STANDBYS=NONE Feature with Oracle Multitenant (Doc ID 1916648.1)

```
--To change the default behavior
--Take care - don't break your standby database
--Works only for databases without ASM or OMF

upg1.manage_standbys_clause=standbys=all
```

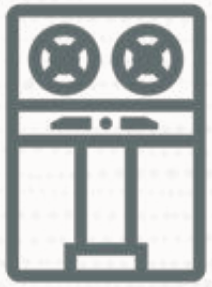
Data Guard | Additional Information

Data Guard Impact on Oracle Multitenant Environments (Doc ID 2049127.1)

The physical standby database and redo apply will normally expect a new PDB's datafiles to have been pre-copied to the standby site and be in such a state that redo received from the primary database can be immediately applied. The standby database ignores any file name conversion specification on the CREATE PLUGGABLE DATABASE statement and relies solely on the standby database's initialization parameter settings for DB_CREATE_FILE_DEST and DB_FILE_NAME_CONVERT for locations and file naming.

For these cases, Oracle recommends deferring recovery of the PDB using the STANDBYS=NONE clause on the CREATE PLUGGABLE DATABASE statement. Recovery of the PDB can be enabled at some point in the future once the PDB's data files have been copied from the primary database to the standby database in a manner similar to that documented in Document 1916648.1.

Confused? We explain all the details in our AutoUpgrade 2.0 webinar



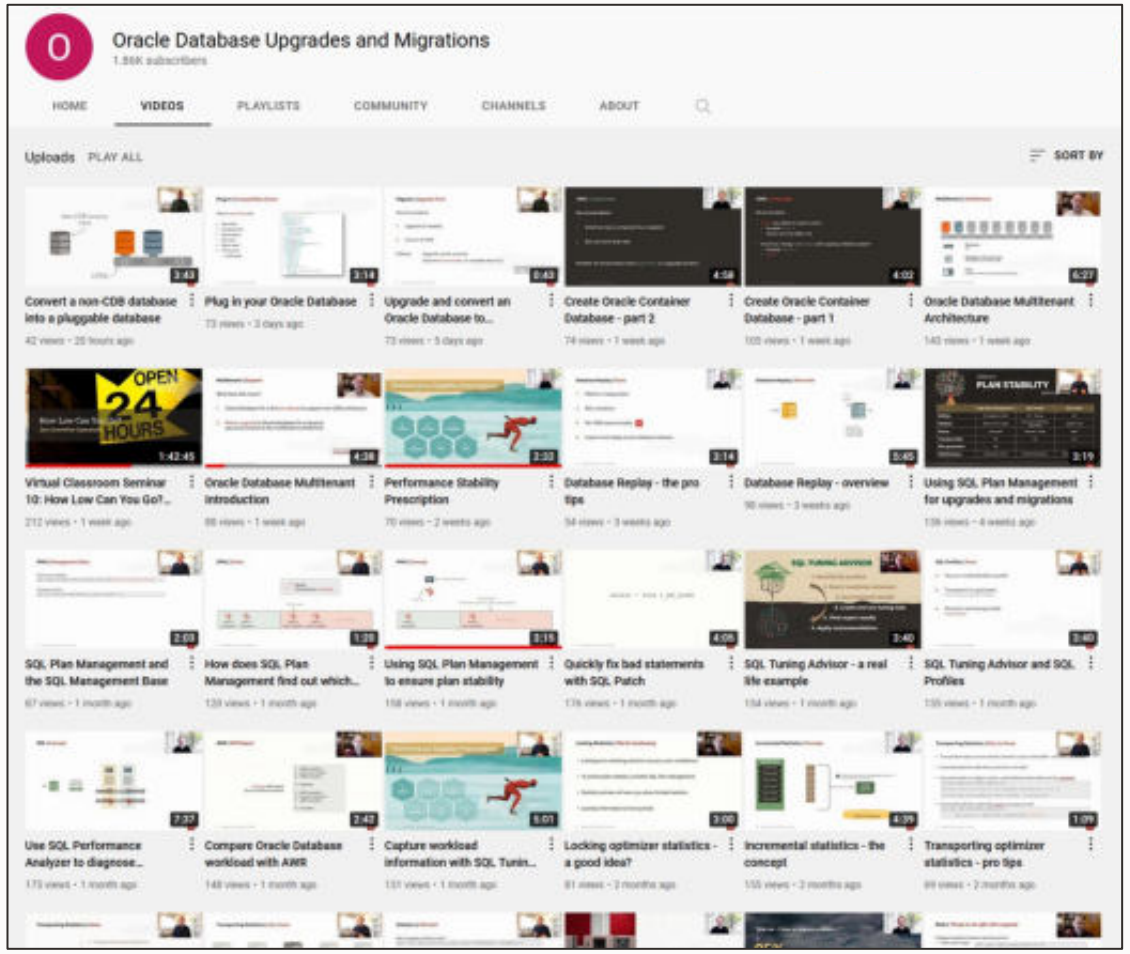
Remember a level 0 backup after migration

You can also restore with pre-plugin backups

Wrapping up

Final Words

YouTube | Oracle Database Upgrades and Migrations



Link

- 300+ videos
- New videos every week
- No marketing
- No buzzwords
- All tech

SCAN ME

THANK YOU

Visit our blogs:

https://MikeDietrichDE.com

https://DOHdatabase.com

https://www.dbarj.com.br/en

THANK YOU

Webinars:

https://MikeDietrichDE.com/videos

YouTube channel:

@UpgradeNow

THANK YOU