



Photo by Philipp Katzenberger on Unsplash

# Migrate to Multitenant with Refreshable Clones using AutoUpgrade

AOUG 2023





# Daniel Overby Hansen

Senior Principal Product Manager  
Cloud Migration



dohdatabase



@dohdatabase



<https://dohdatabase.com>



# Mike Dietrich

Senior Director Product Management  
Database Upgrade, Migration and Patching

---



MikeDietrich



@MikeDietrichDE



<https://MikeDietrichDE.com>

### 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_pdb=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



CATALOG  
CATPROC  
XDB  
OWM

CATALOG  
CATPROC  
XDB  
OLS

CATALOG  
CATPROC  
XDB  
SPATIAL



CATALOG  
CATPROC  
XDB  
OWM  
OLS  
SPATIAL





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='19.19.0' 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://mikedietchde.com/2018/08/08/creating-cdbs-non-cdbs-with-less-options/>
- <https://mikedietchde.com/2017/07/11/always-create-custom-database/>
- <https://mikedietchde.com/2017/07/26/remove-clean-components-oracle-11-2-12-2/>

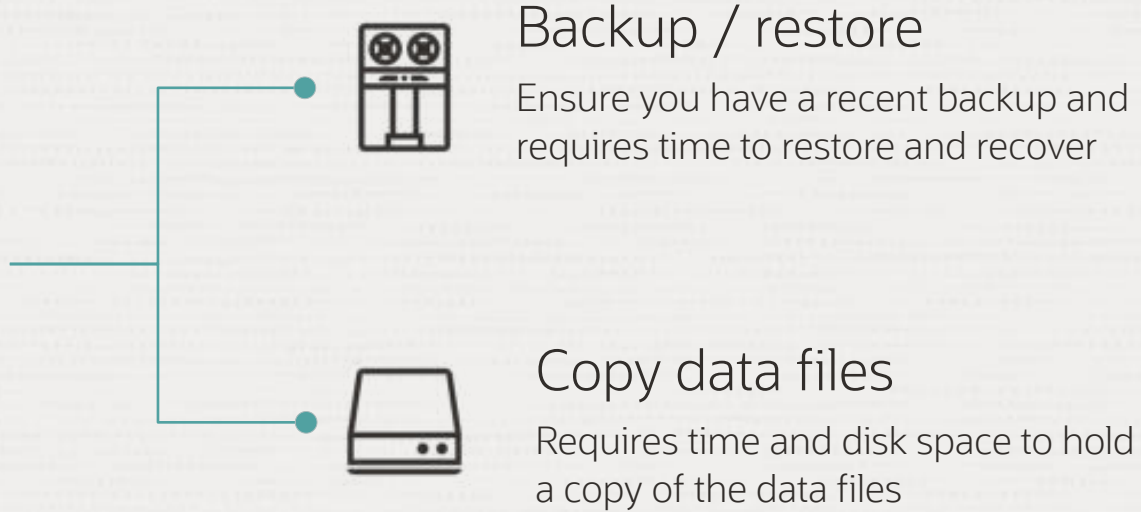


# Non-CDB to PDB conversion is irreversible

What are your fallback options?

# PDB Conversion

## FALLBACK



# 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



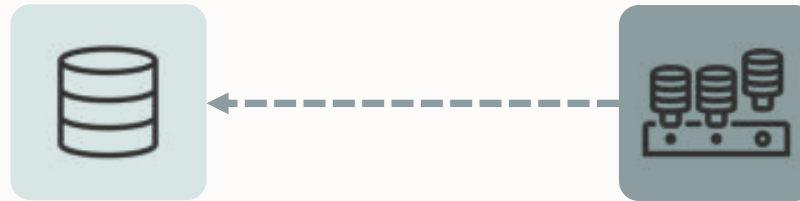
```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=NONCDB1
upg1.target_cdb=CDB1
upg1.source_dblink.NONCDB1=CLONEPDB
upg1.target_pdb_name.NONCDB1=PDB1
```



# 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

# Customer Case | Zürcher Kantonalbank

## Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

A reliable partner for over 150 years

- The bank for the people of Zurich since 1870
- With over 5'100 employees one of the largest employers in the canton of Zurich
- Globally networked full-service bank with strong regional and local roots



# Customer Case | Zürcher Kantonalbank

Customer

**Project**

Constraints

Preparation

Migration

Success?

Remarks

Current situation

- Oracle databases on old OS and on Oracle Exadata
- 2023:
  - Migrate everything to Exadata until end of 2023
  - Consolidation to Multitenant and to the next long-term support release

Planned solution: AutoUpgrade



# Customer Case | Zürcher Kantonalbank

Customer

Project

Constraints

**Preparation**

Migration

Success?

Remarks

## Test setup

- 3 non-CDB databases of different size

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

- Exadata X6-2 compute node
- 7 storage cells (2x X6-2L / 3x X7-2L / 2x X8-2L)
- Oracle Database 19.15.0
- No additional options





# Customer Case | Zürcher Kantonalbank

Customer

Cloning user

Project

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

Constraints

Permissions

**Preparation**

```
grant CONNECT, RESOURCE, CREATE PLUGGABLE DATABASE,  
      SELECT_CATALOG_ROLE to dblinkuser;  
grant ALL ON SYS.ENC$ to dblinkuser;
```

Migration

Success?

Database link

Remarks

```
create database link TEST42.DOMAIN connect to dblinkuser  
identified by oracle_4uoracle_4u using 'test42.domain';
```

# Customer Case | Zürcher Kantonalbank

Customer

Project

Constraints

Preparation

**Migration**

Success?

Remarks

Migration in progress

```

Job#|DB_NAME| STAGE|OPERATION| STATUS|START_TIME| UPDATED| MESSAGE|
-----|-----|-----|-----|-----|-----|-----|-----|
| 106| | CLONEFDB|EXECUTING| RUNNING| 09:50:39|!!!17381s ago|Creating pluggable database|
| 107| | |COMPLETED| STOPPED|FINISHED| 09:50:40| |
| 108| | |COMPLETED| STOPPED|FINISHED| 09:50:40| |
-----|-----|-----|-----|-----|-----|-----|-----|
Total jobs 3

The command lsj is running every 60 seconds. PRESS ENTER TO EXIT

Job#|DB_NAME| STAGE|OPERATION| STATUS|START_TIME| UPDATED| MESSAGE|
-----|-----|-----|-----|-----|-----|-----|-----|
| 106| | CLONEFDB|EXECUTING| RUNNING| 09:50:39|!!!17441s ago|Creating pluggable database|
| 107| | |COMPLETED| STOPPED|FINISHED| 09:50:40| |
| 108| | |COMPLETED| STOPPED|FINISHED| 09:50:40| |
-----|-----|-----|-----|-----|-----|-----|-----|
Total jobs 3

The command lsj is running every 60 seconds. PRESS ENTER TO EXIT

Job#|DB_NAME| STAGE|OPERATION| STATUS|START_TIME| UPDATED| MESSAGE|
-----|-----|-----|-----|-----|-----|-----|-----|
| 106| | CLONEFDB|EXECUTING| RUNNING| 09:50:39|!!!17501s ago|Creating pluggable database|
| 107| | |COMPLETED| STOPPED|FINISHED| 09:50:40| |
| 108| | |COMPLETED| STOPPED|FINISHED| 09:50:40| |
-----|-----|-----|-----|-----|-----|-----|-----|
Total jobs 3

The command lsj is running every 60 seconds. PRESS ENTER TO EXIT
    
```

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



# Customer Case | Zürcher Kantonalbank

Customer

Project

Constraints

Preparation

**Migration**

Success?

Remarks

Migration completed

```

The command is running every 60 seconds, PRESS ENTER TO EXIT
-----
|Job#|DB_NAME| STAGE|OPERATION| STATUS| START_TIME|UPDATED| MESSAGE|
-----
| 194| | CLONEPDB| STOPPED| ERROR|Oct-23 09:50| | UPG-4014|
| 197| | (COMPLETED) STOPPED|FINISHED|Oct-23 09:50| | |
| 198| | (COMPLETED) STOPPED|FINISHED|Oct-23 09:50| | |
-----
Total jobs 3

The command is running every 60 seconds, PRESS ENTER TO EXIT
-----
|Job#|DB_NAME| STAGE|OPERATION| STATUS| START_TIME|UPDATED| MESSAGE|
-----
| 104| | CLONEPDB| STOPPED| ERROR|Oct-23 09:50| | UPG-4014|
| 107| | (COMPLETED) STOPPED|FINISHED|Oct-23 09:50| | |
| 108| | (COMPLETED) STOPPED|FINISHED|Oct-23 09:50| | |
-----
Total jobs 3

The command is running every 60 seconds, PRESS ENTER TO EXIT

rpg>
rpg> exit
There is 1 job in progress. If you exit it will stop.
Are you sure you wish to leave? [y/n] y
-----
Final Summary
-----
Number of databases      [ 3 ]
Jobs finished            [ 2 ]
Jobs failed              [ 1 ]
    
```

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



# Customer Case | Zürcher Kantonalbank

Customer

First non-CDBs migrated successfully

Project

- Project is ongoing

Constraints

Preparation

Migration

**Success?**

Remarks

# Customer Case | Zürcher Kantonalbank

Customer

Project

Constraints

Preparation

Migration

Success?

**Remarks**

For large databases, make sure archives aren't cleaned up

- Solution: restore archivelogs from backup

User profile with IDLE\_TIME lead to kill of the session

- Solution: assign a different profile to the clone user



## Summary

- Very comfortable to use
  - Everything happens automatically
  - Does not require user interaction
- Simple syntax
- No license costs associated
- Perfect for pre-migration test
  
- Very Stable



# 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 3<sup>rd</sup> 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';
```

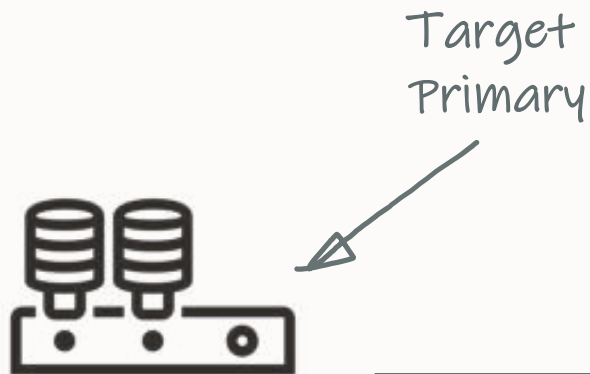
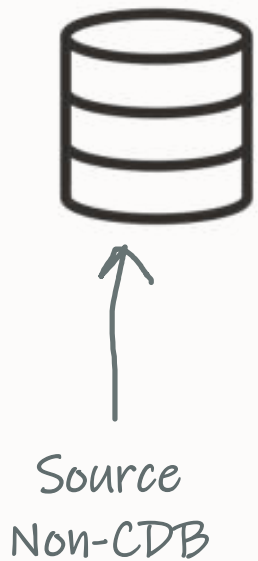
```
SQL_TEXT
/* SQL Analyze(256,0) */ SELECT /*+PARALLEL(4) NO_STATEMENT_QUEUEING */ * FROM X$KXFTASK /*kpdbfParallelCopyOrMove,PDB_FILE_COPY*/
```



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`



```
RMAN> restore pluggable database ...  
from service ... ;  
  
SQL> alter pluggable database  
enable recovery;
```

~~PTD~~ Targeted  
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**

[Goal](#)

[Solution](#)

- [Creating a PDB with the STANDBYS=NONE clause in a Data Guard configuration with 1 physical standby](#)
- [Showing how the cloned PDB will appear in certain tables and views on the physical standby](#)
- [Performing a Data Guard Role Transition with a PDB in DISABLED RECOVERY](#)
- [The zero downtime instantiation process using RMAN for copying the files from the primary to standby](#)
- [Steps required for enabling recovery on the PDB after the files have been copied](#)
- [Steps to DISABLE RECOVERY of a Pluggable Database](#)

[Conclusion](#)

[References](#)

---

**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

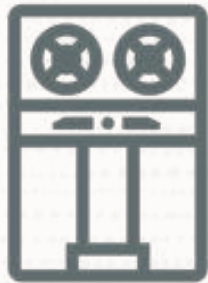
## [Data Guard Impact on Oracle Multitenant Environments \(Doc ID 2049127.1\)](#)

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

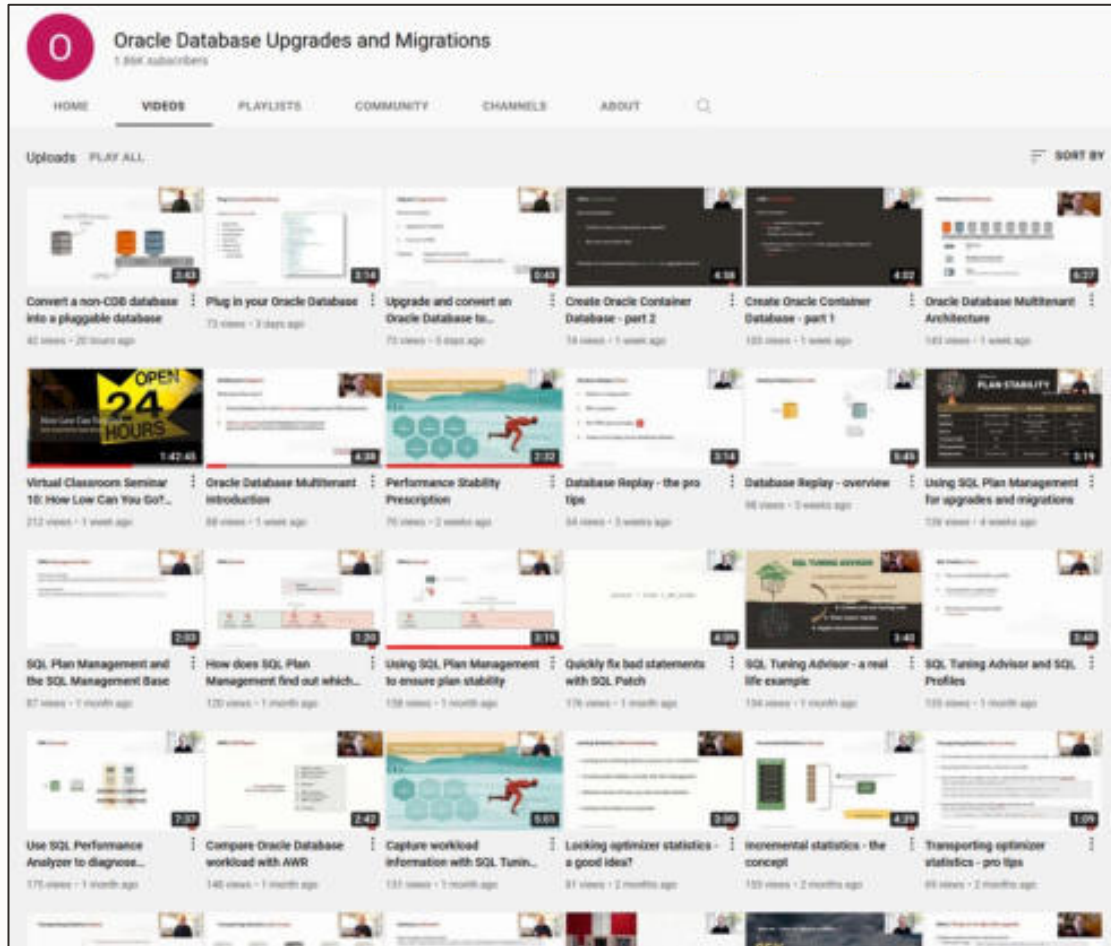
You can also restore with pre-plugin backups

# Wrapping up



Final Words

# YouTube | Oracle Database Upgrades and Migrations



[Link](#)

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



# 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**

