

ORACLE

Oracle Multitenant Migration

Deep Dive and Secret Secrets

Mike Dietrich & Daniel Overby Hansen

Database Upgrade, Utilities and Patching





Daniel Overby Hansen

Senior Principal Product Manager
Cloud Migration

 <https://dohdatabase.com>

 [@dohdatabase](https://twitter.com/dohdatabase)

 [dohdatabase](https://www.linkedin.com/company/dohdatabase)



Mike Dietrich

Distinguished Product Manager
Database Upgrade
and Migrations

 <https://MikeDietrichDE.com>

 @MikeDietrichDE

 mikedietrich

NEW Episode 1

Release and Patching Strategy

105 minutes – Feb 4, 2021



NEW Episode 2

AutoUpgrade to Oracle Database 19c

115 minutes – Feb 20, 2021



NEW Episode 3

Performance Stability, Tips and Tricks and Underscores

120 minutes – Mar 4, 2021



NEW Episode 4

Migration to Oracle Multitenant

120 minutes – Mar 16, 2021



NEW Seminar 5

Migration Strategies – Insights, Tips and Secrets

120 minutes – Mar 25, 2021



NEW Seminar 6

Move to the Cloud – Not only for techies

115 minutes – Apr 8, 2021



NEW Episode 7

Cool Features – Not only for DBAs

110 minutes – Jan 14, 2021



NEW Episode 8

Database Upgrade Internals – and so much more



Recorded Web Seminars

<https://MikeDietrichDE.com/videos/>

<https://dohdatabase.com/webinars/>




https://MikeDietrichDE.com

https://DOHdatabase.com


Upgrade your Database - NOW!

[Blog](#) [Slides](#) [Hands-On Lab](#) [Events](#) [Papers / Docs](#) [Videos](#) [Scripts](#) [Links](#) [Oracle Documentation](#) [Privacy Policy](#) [About](#)


Upgrade to Oracle 19...




Test Transportable Ta...



User Group Seminars...





Which database version can be used with Grid Infrastructure 19c?

Posted on October 6, 2020 by Mike.Dietrich [ASM](#) / [RAC](#) / [GI](#)

This will be a very short blog post today. But a colleague asked me this question just a few minutes ago: Which database version can be used with Grid Infrastructure 19c? And I realized that I neither put it on the blog yet nor did I bookmark it. But I answered this question several times already.

Which database version

Databases Are Fun

dohdatabase.com

[Twitter](#) [LinkedIn](#) [RSS](#)

[Blog](#) [Categories](#) [About](#)

Blog

Debut on the Big Stage

In mid-October you can join our two webinars on database migration. One of them is exclusively about cloud migration... for techies only.

Follow Blog via Email

Enter your email address to follow this blog and receive notifications of new posts by email.

[Follow](#) ***

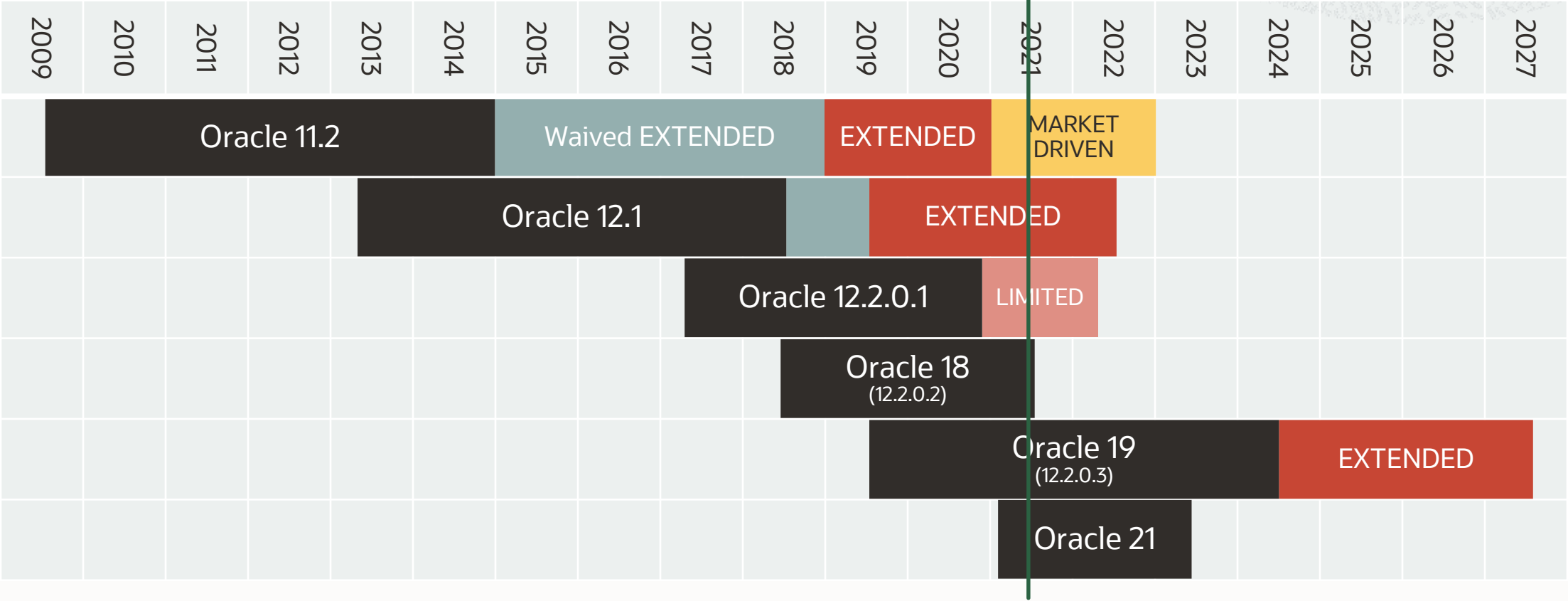


Chapter 1

Release Strategy



Lifetime Support Policy



Premier Support Waived Extended Support Paid Extended Support Market Driven Support Limited Error Correction



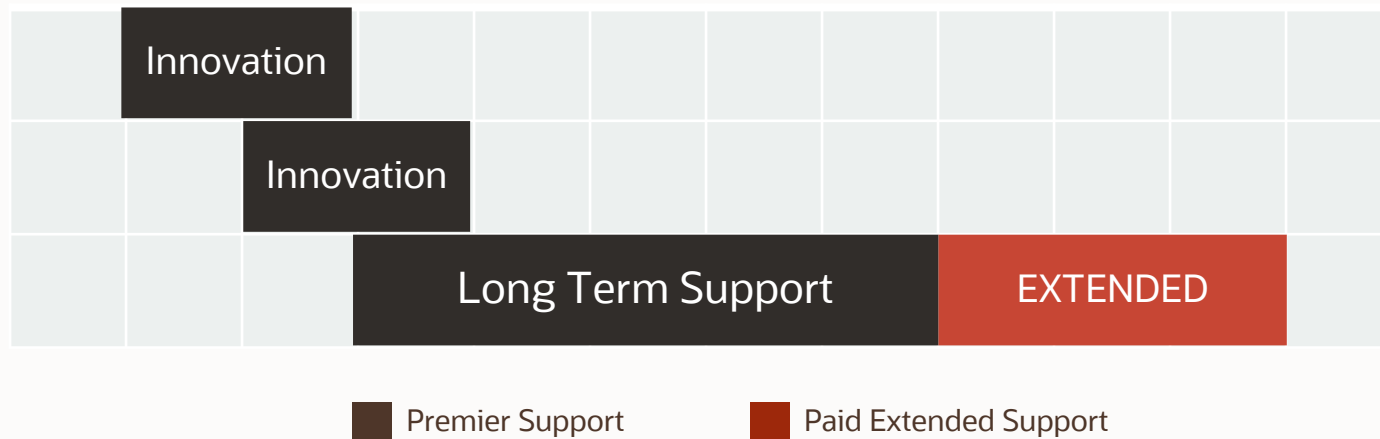
Release Types | Long Term Support vs Innovation Releases

Long Term Support Release

- 5 years of Premier Support followed by 3 years of Extended Support

Innovation Release

- 2 years of Premier Support, but there is no Extended Support

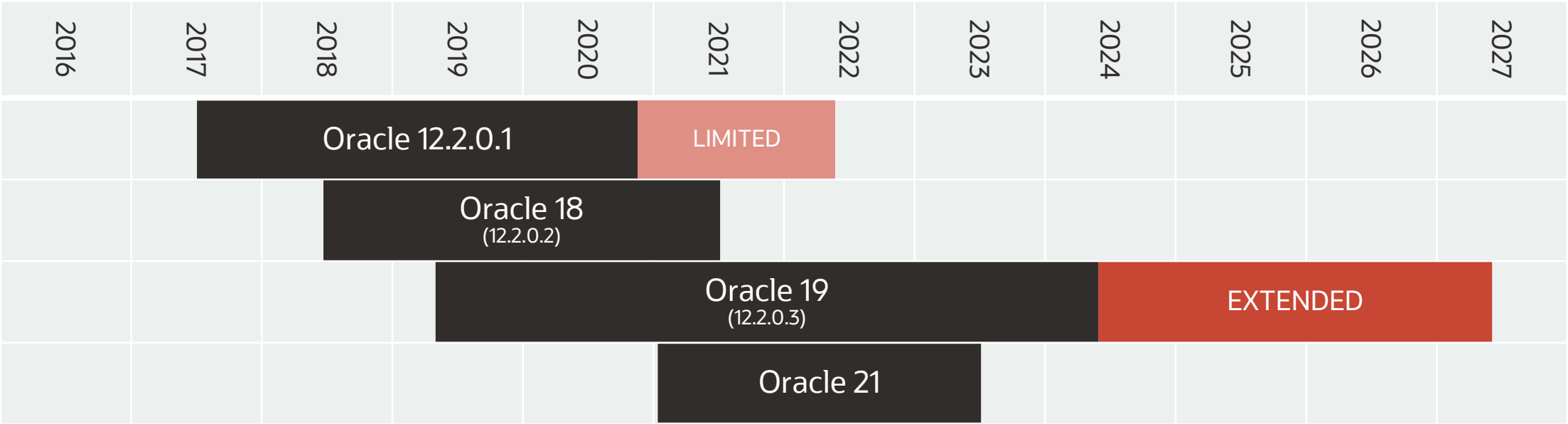


- [MOS Note: 742060.1 - Release Schedule of Current Database Releases](#)

Oracle 12.2 Release Family and beyond

Includes:

- Oracle 12.2.0.1, Oracle 18c (12.2.0.2), Oracle 19c (12.2.0.3)



- [MOS Note:742060.1](#) - The Single Source of Truth
- [MOS Note:161818.1](#) - Releases Support Status Summary



Multitenant | Support



”

Desupport of Non-CDB Oracle Databases

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](#)

Multitenant | Support

What does this mean?

1. Oracle Database 19c is the **last release** to support non-CDB architecture
2. **Before upgrade** to Oracle Database 21c or beyond, you must convert to the multitenant architecture



Pro tip: For further details see [Release Schedule of Current Database Releases \(Doc ID 742060.1\)](#)

Multitenant | License



Included in **all** offerings (SE2/EE)

12.1.0.2
12.2.0.1
18c

Single tenant
Max. **1** PDB

19c
21c

Multitenant
Max. **3** PDB

```
SQL> alter system set max_pdb=3;
```

Multitenant | License



Included in **Multitenant Option** (EE only)

On-prem Enterprise Edition	252 PDBs Extra cost
Engineered Systems Exadata / ODA	4096 PDBs Extra cost
ExaCS DBCS EE-HP DBCS EE-EP	4096 PDBs Included



Chapter 2

Multitenant Migration



How to

MIGRATE

to multitenant architecture



1.
Create

2.
Upgrade

3.
Plug In

4.
Convert

CDB | Components

- Create CDBs with fewer components
- DBCA and custom mode creation

Database Configuration Assistant - Create a database - Step 2 of 15

Select Database Creation Mode

19c ORACLE Database

Database Operation

Creation Mode

Deployment Type

Database Identification

Storage Option

Fast Recovery Option

Network Configuration

Database Options

Configuration Options

Management Options

User Credentials

Creation Option

Summary

Progress Page

Finish

Typical configuration

Global database name: CDB12

Storage type: File System

Database files location: {ORACLE_BASE}\oradata\{DB_UNIQUE_NAME} Browse...

Fast Recovery Area (FRA): BASE\fast_recovery_area\{DB_UNIQUE_NAME} Browse...

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

Administrative password: *****

Confirm password: *****

Create as Container database

Pluggable database name: PDB1

Advanced configuration

Database Configuration Assistant - Create a database - Step 3 of 15

Select Database Deployment Type

19c ORACLE Database

Database Operation

Creation Mode

Deployment Type

Database Identification

Storage Option

Fast Recovery Option

Network Configuration

Database Options

Configuration Options

Management Options

User Credentials

Creation Option

Summary

Progress Page

Finish

Select the type of database you want to create.

Database type: Oracle Single Instance database

Configuration type: Admin Managed

Select a template for your database.

Templates that include datafiles contain pre-created databases. They allow you to create a new database quickly. Use templates without datafiles only when necessary, such as when you need to change attributes like block size that cannot be altered after database creation.

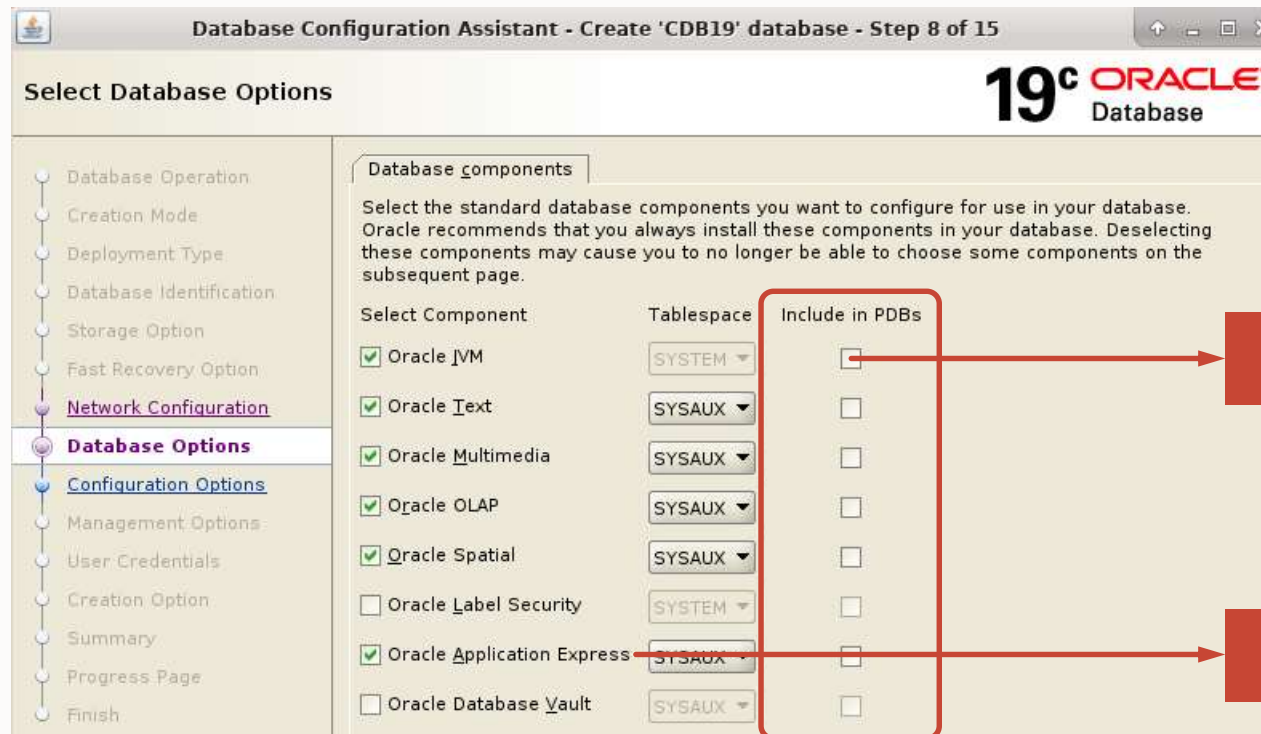
Template name	Include datafiles	Details
Data Warehouse	Yes	View details
Custom Database	No	View details
General Purpose OLTP Transaction Processing	Yes	View details

Template location: /u01/app/oracle/product/19/assistants/dbca/templates Change...



CDB | Components

Only *Advanced Configuration > Custom Database* gives you

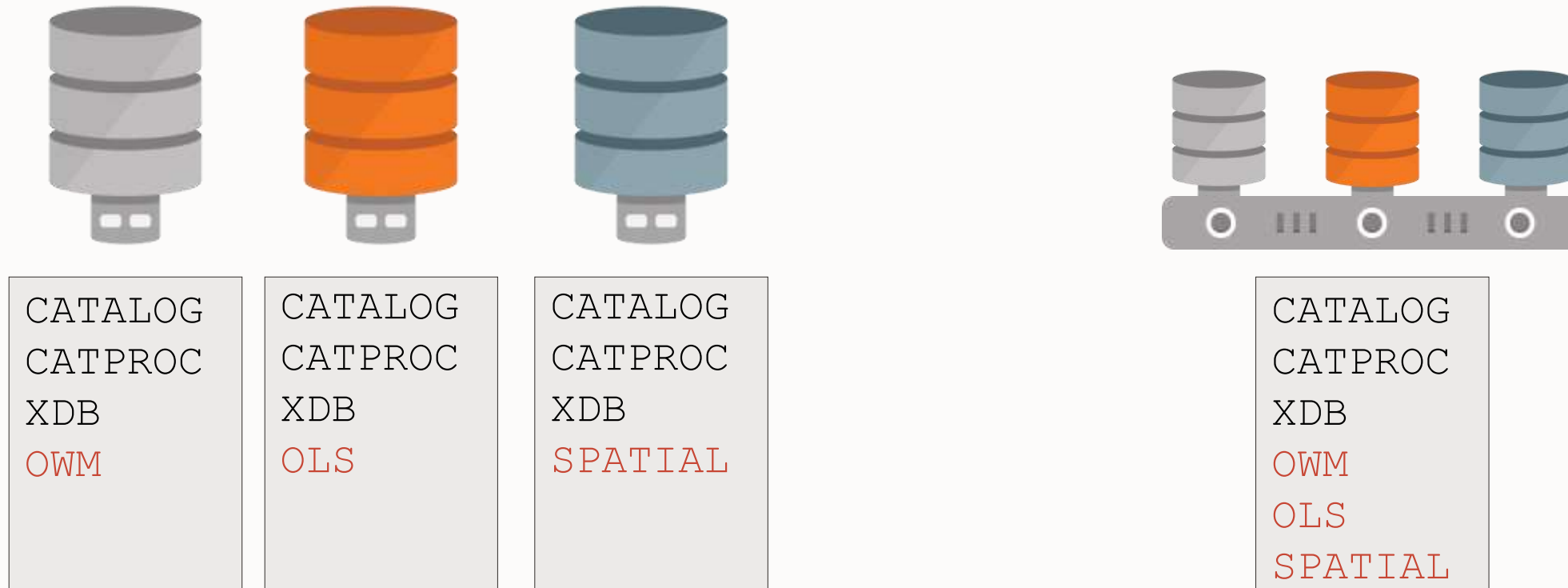


Be aware!

Be really aware!

CDB | Components

CDB\$ROOT must be a **superset** of all PDBs



CDB | Components

Recommendation

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

Number of components have **big effect** on upgrade duration

Components (e.g., JAVAVM) may require patch regular activity

CDB | Compatible

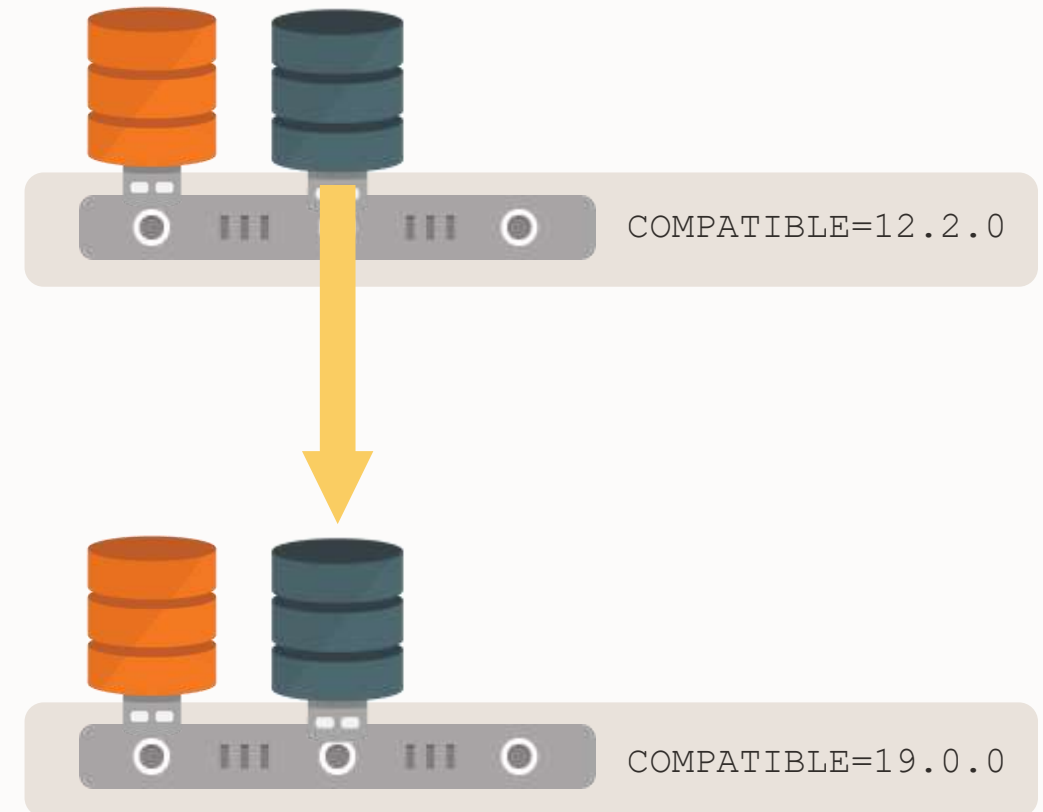
Recommendation

- **Always** use default of a given version
 - Example 19.0.0
 - Always use three digits only
- Should you change COMPATIBLE after applying a Release Update?
 - Example 19.10.0
 - **Never**

CDB | Silent Compatible Change

Be aware

- When you unplug and plugin a PDB, it will adopt `COMPATIBLE` of the CDB automatically
- Changing `COMPATIBLE` will prevent re-plug and downgrade



How to

MIGRATE

to multitenant architecture



1.
Create

2.
Upgrade

3.
Plug In

4.
Convert

Migrate | Upgrade First

Recommendation

1. Upgrade (if needed)
2. Convert to PDB

Fallback: Upgrade can be reverted
Convert is **irreversible**, re-runnable from 12.2

Pro tip: Plugging in a PDB makes changes to the file headers of the datafiles that cannot be undone - even by flashback database



How to

MIGRATE

to multitenant architecture



1.
Create

2.
Upgrade

3.
Plug In

4.
Convert

Plug In | Compatibility Check

Is my database compatible with this CDB?

1. In source, generate manifest file

```
SQL> exec dbms_pdb.describe('/tmp/DB19.xml');
```

2. In CDB, check compatibility

```
set serveroutput on

BEGIN
  IF dbms_pdb.check_plug_compatibility('/tmp/DB19.xml') THEN
    dbms_output.put_line('PDB compatible? ==> Yes');
  ELSE
    dbms_output.put_line('PDB compatible? ==> No');
  END IF;
END;
/
```

Pro tip: You can [generate a manifest file of a remote database](#) via a database link



Plug In | Compatibility Check

3. Always check the details

```
SQL> select type, message
      from   PDB_PLUG_IN_VIOLATIONS
      where  name='DB19' and status<>'RESOLVED';
```

TYPE	MESSAGE
ERROR	'19.9.0.0.0 Release_Update' is installed in the CDB but no release updates are installed in the PDB
ERROR	DBRU bundle patch 201020: Not installed in the CDB but installed in the PDB
ERROR	PDB's version does not match CDB's version: PDB's version 12.2.0.1.0. CDB's version 19.0.0.0.0.
WARNING	CDB parameter compatible mismatch: Previous '12.2.0' Current '19.0.0'
WARNING	PDB plugged in is a non-CDB, requires noncdb_to_pdb.sql be run.

Pro tip: Errors will prevent you from opening the PDB in unrestricted mode

Plug In | Compatibility Check

What is a **manifest** file

- Data files
- Components
- Parameters
- Services
- Patch level
- Time zone
- ... and more

```
<?xml version="1.0" encoding="UTF-8"?>
<PDB>
  <xmlversion>1</xmlversion>
  <pdbname>DB12</pdbname>
  <cid>0</cid>
  <byteorder>1</byteorder>
  <vsns>203424000</vsns>
  <vsns>
    <vsnum>12.2.0.1.0</vsnum>
    <cdbcomp>12.2.0.0.0</cdbcomp>
    <pdbscomp>12.2.0.0.0</pdbscomp>
    <vslibnum>0.0.0.0.24</vslibnum>
    <vsnsql>24</vsnsql>
    <vsnsbv>8.0.0.0.0</vsnsbv>
  </vsns>
  <dbid>1852833295</dbid>
  <ncdb2pdb>1</ncdb2pdb>
  <cdbid>1852833295</cdbid>
  <guid>86D5DC2587337002E0532AB2A8C0A57C</guid>
  <uscnbas>4437941</uscnbas>
  <uscnwrp>0</uscnwrp>
  <undosc>8</undosc>
  <rdba>4194824</rdba>
  <tablespace>
    <name>SYSTEM</name>
    <type>0</type>
    <tsn>0</tsn>
    <status>1</status>
    <issft>0</issft>
    <isnft>0</isnft>
    <encts>0</encts>
    <flags>0</flags>
    <bmunitsize>8</bmunitsize>
    <file>
      <path>/u02/oradata/DB12/system01.dbf</path>
      <afn>1</afn>
      <rfn>1</rfn>
    </file>
  </tablespace>
</PDB>
```

Plug In | Compatibility Check

- Optional
- Possible when source database is 12.1 or newer
- Enables you to see which plug-in violations will occur
- If you plan on renaming the PDB, check the new name

```
dbms_pdb.check_plug_compatibility('/tmp/DB19.xml', 'SALES')
```

Pro tip: Default PDB name in manifest file is `DB_NAME`



Plug In | Create PDB



1. Restart database in read-only mode

```
SQL> shutdown immediate  
SQL> startup mount  
SQL> alter database open read only;
```

2. Generate manifest file and shut down

```
SQL> exec dbms_pdb.describe('/tmp/DB19.xml');  
SQL> shutdown immediate;
```

3. In CDB, create PDB from manifest file

```
SQL> create pluggable database DB19  
      using '/tmp/DB19.xml' nocopy tempfile reuse;
```

Pro tip: CREATE PLUGGABLE DATABASE has many options

How to

MIGRATE

to multitenant architecture



1.
Create

2.
Upgrade

3.
Plug In

4.
Convert

Convert | Create PDB



1. Open PDB

```
SQL> alter pluggable database DB19 open;  
SQL> alter session set container=DB19;
```

2. Convert and restart

```
SQL> @?/rdbms/admin/noncdb_to_pdb.sql
```

3. Restart PDB

```
SQL> alter pluggable database DB19 close;  
SQL> alter pluggable database DB19 open;
```

Pro tip: As of 12.2 the script
`noncdb_to_pdb.sql` is re-runnable

Convert | Create PDB



4. Check plug-in violations

```
SQL> select type, message  
       from   pdb_plug_in_violations  
       where  name='DB19' and status<>'RESOLVED';
```

5. Purge

```
SQL> exec dbms_pdb.clear_plugin_violations(pdb_name => 'DB19');
```

6. Ensure PDB is open READ WRITE and unrestricted

```
SQL> select open_mode, restricted from v$pdb;
```

7. Configure PDB to auto-start

```
SQL> alter pluggable database DB19 save state;
```

Pro tip: For RAC databases control PDB startup through Clusterware

Convert | **noncdb_to_pdb.sql**

- Requires downtime
- Runtime varies - typically 10-30 min
 - Fix for Bug 25809128 is included since 19.9.0 and adds a **significant** improvement
- Runs **only once** in the life of a database
- Irreversible
- Re-runnable from 12.2

Pro tip: If you want more details refer to [blog post](#)



other

MIGRATION

options

GoldenGate

Transportable

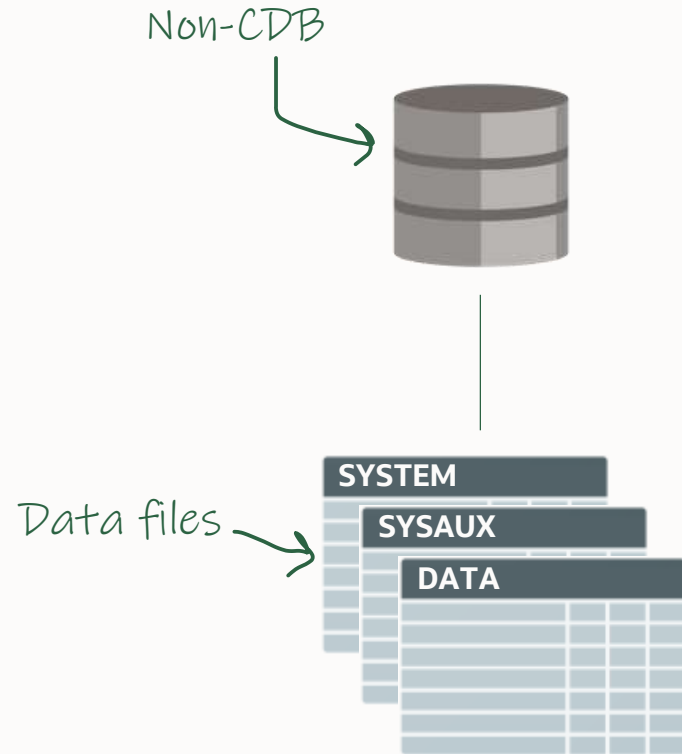
Data Pump

Plug-in Copy

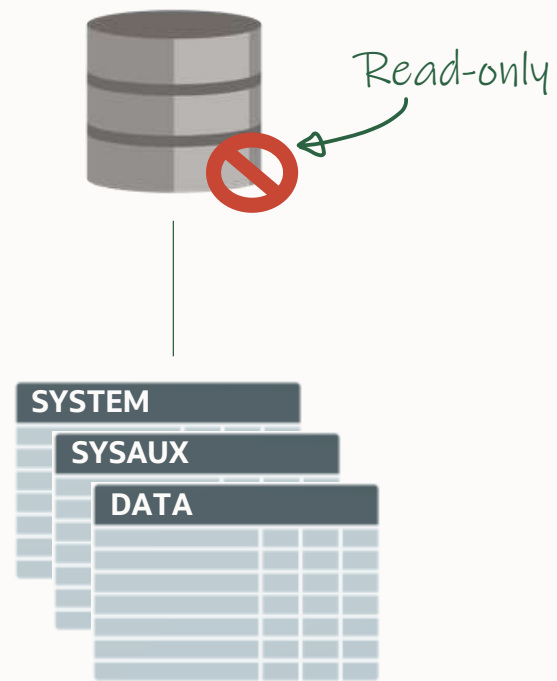
Plug-in NoCopy



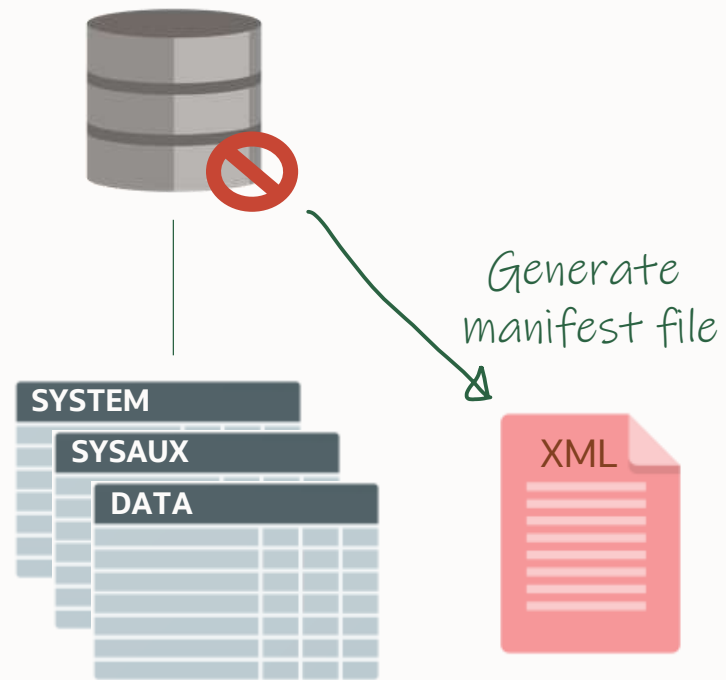
Plug-in NoCopy | Concept



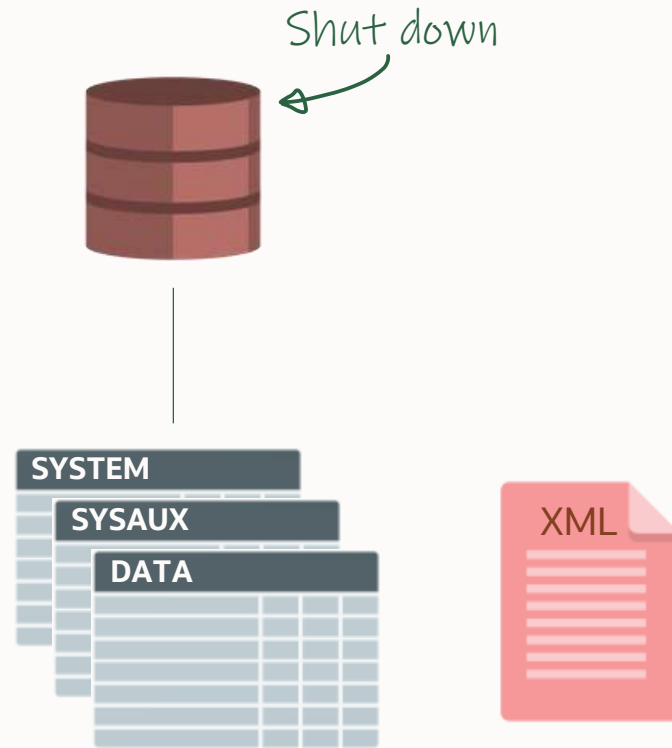
Plug-in NoCopy | Concept



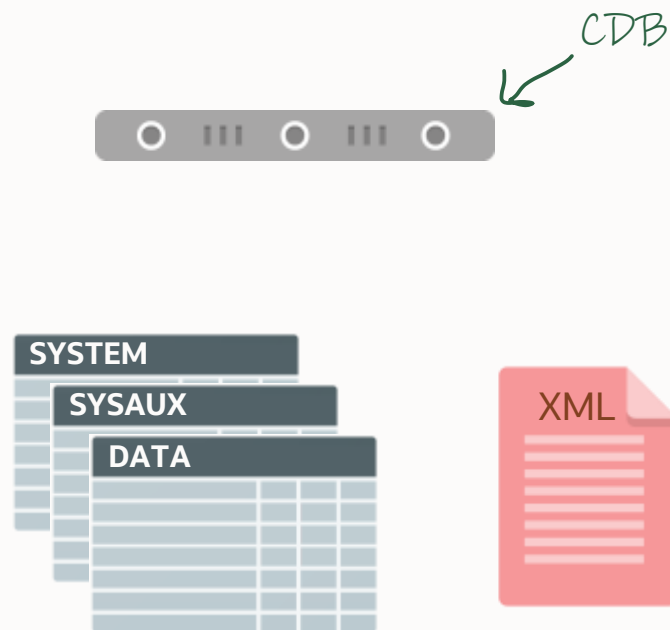
Plug-in NoCopy | Concept



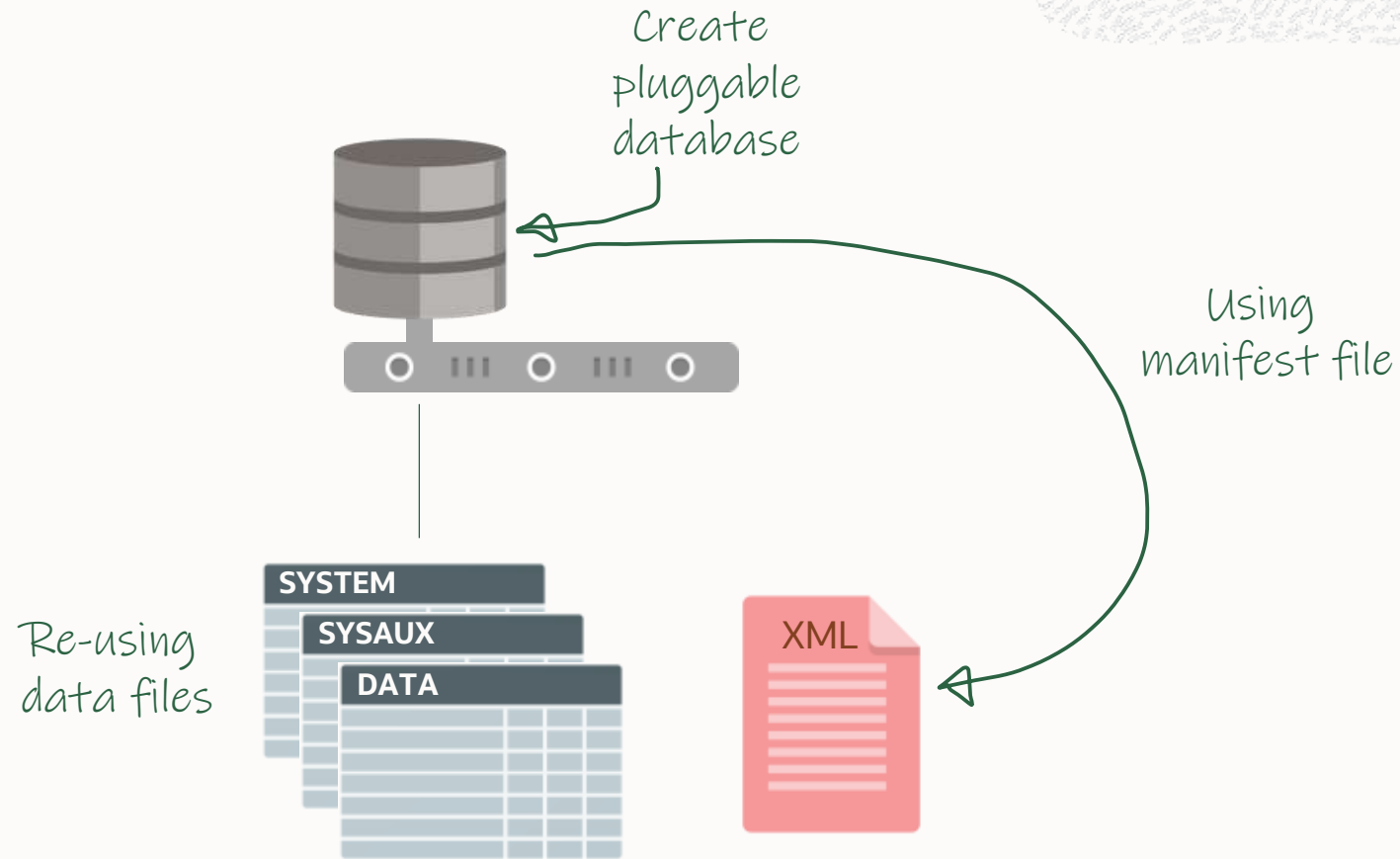
Plug-in NoCopy | Concept



Plug-in NoCopy | Concept



Plug-in NoCopy | Concept



Plug-in NoCopy | Create

Re-use existing data files

```
SQL> CREATE PLUGGABLE DATABASE DB19 ... NOCOPY ... ;
```

Move data files

```
SQL> CREATE PLUGGABLE DATABASE DB19 ... MOVE ... ;
```

Plug-in NoCopy | AutoUpgrade

Fully automated plug-in only

```
upg1.source_home=/u01/app/oracle/product/19  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=DB19  
upg1.target_cdb=CDB2
```

Command

```
java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

Pro tip: Always get latest version of AutoUpgrade from MOS [2485457.1](#)



Plug-in NoCopy | AutoUpgrade

Upgrade - and plug in

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=DB12  
upg1.target_cdb=CDB2
```

Command

```
java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

Blog post: [Oracle AutoUpgrade between two servers – and Plugin?](#)

Pro tip: You can also plug in manually and upgrade PDB with `dbupgrade -c DB19`



Plug-in NoCopy | Nice to know

No fallback

- Data files are re-used

Fast option

Cross-platform

- Potentially roll off patches before unplug
- But can't go across Endian format



other

MIGRATION

options

GoldenGate

Transportable

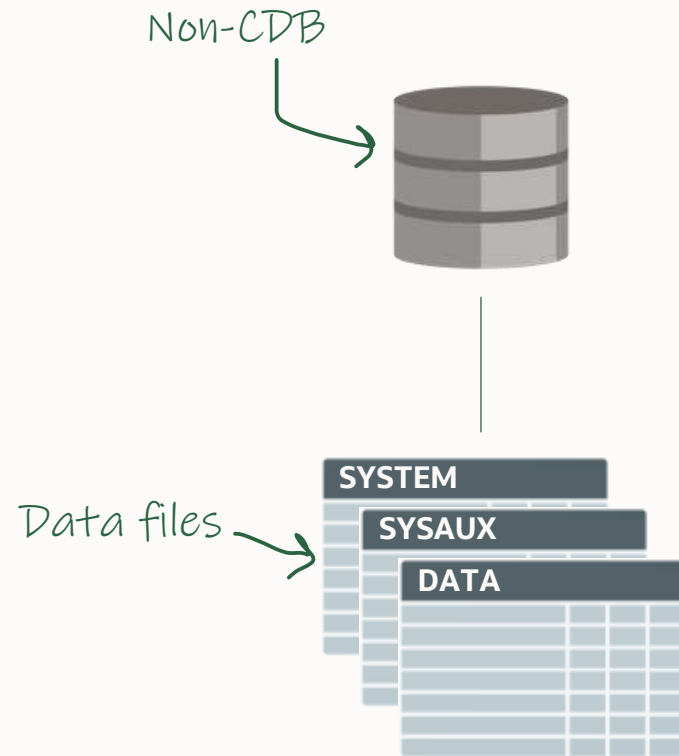
Data Pump

Plug-in Copy

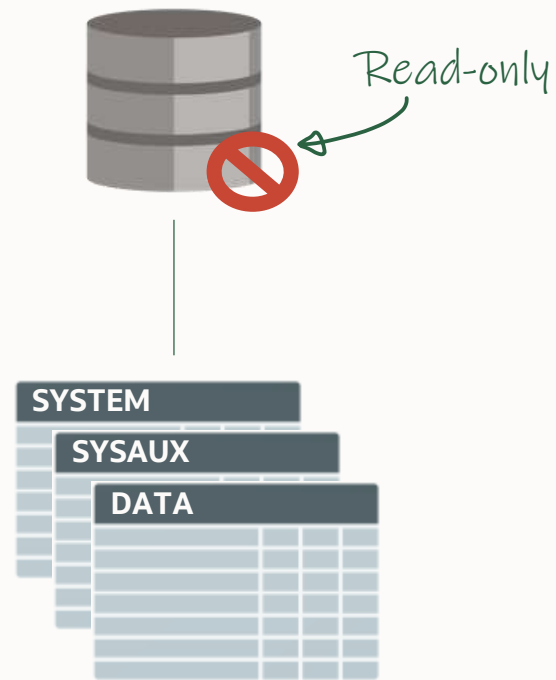
Plug-in NoCopy



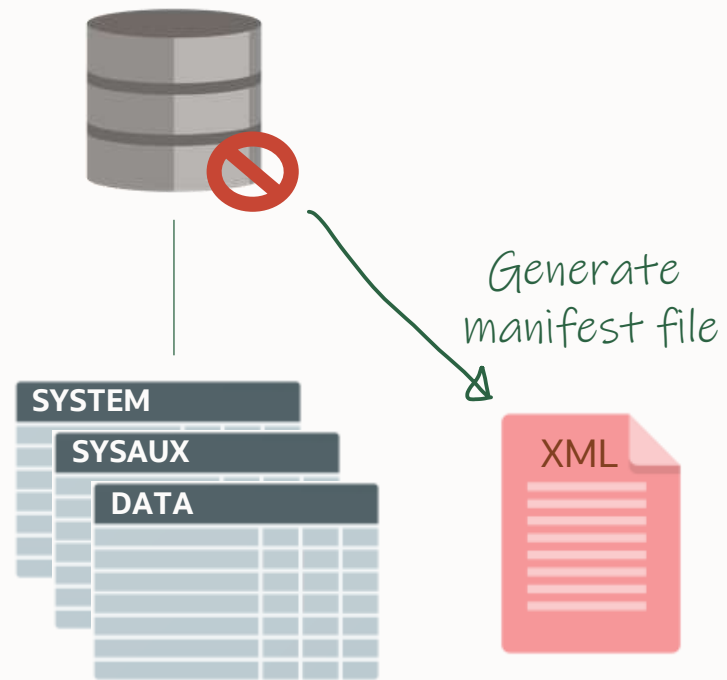
Plug-in Copy | Concept



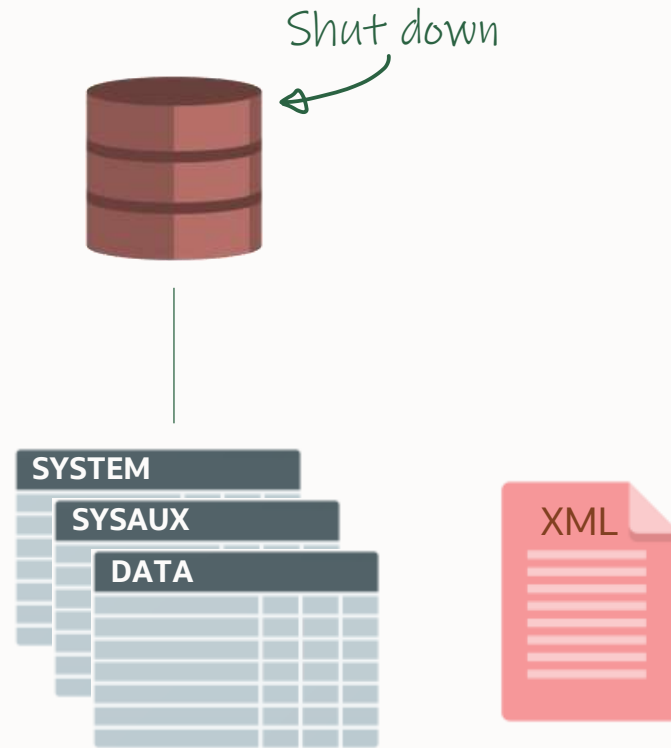
Plug-in Copy | **Concept**



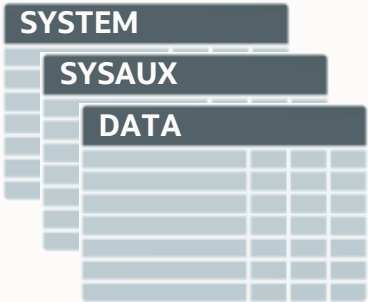
Plug-in Copy | Concept



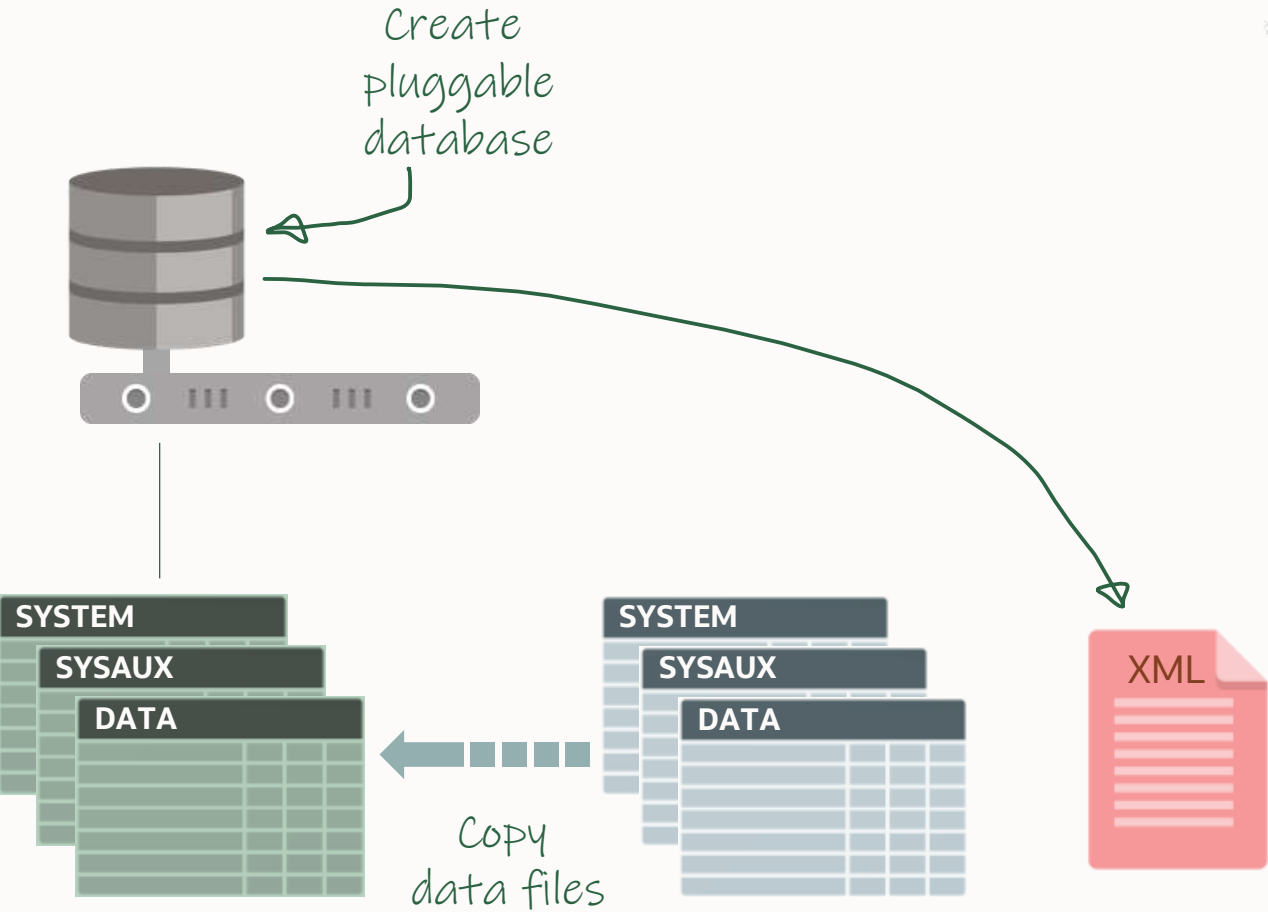
Plug-in Copy | Concept



Plug-in Copy | Concept



Plug-in Copy | Concept



Plug-in Copy | Create

Copy data files

```
SQL> CREATE PLUGGABLE DATABASE DB19 ... COPY FILE_NAME_CONVERT= ... ;
```

Rename data files with FILE NAME CONVERT

- Regular search/replace
FILE_NAME_CONVERT= ('DB19', 'SALES')
- OMF
FILE_NAME_CONVERT=NONE

Pro tip: Use the same FILE_NAME_CONVERT clause for plug-in with MOVE keyword



Plug-in Copy | AutoUpgrade

Fully automated plug-in

```
upg1.source_home=/u01/app/oracle/product/19  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=DB19  
upg1.target_cdb=CDB2  
upg1.target_pdb_name=SALES  
upg1.target_pdb_copy_option=file_name_convert=('DB19','SALES')
```

Command

```
Java -jar autoupgrade.jar -config DB19.cfg -mode deploy
```

Plug-in Copy | Nice to know

Fallback option

- Original data files are preserved

Slow and requires additional disk space

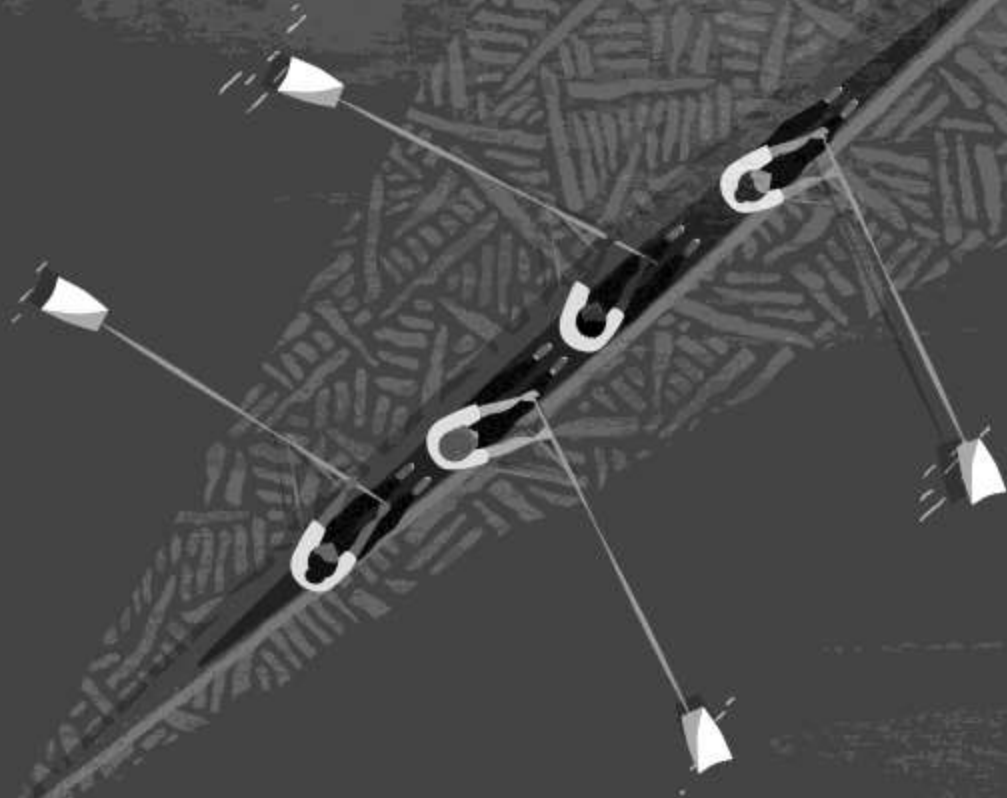
Cross-platform

- Potentially roll off patches before unplug
- But can't go across Endian format



FALLBACK

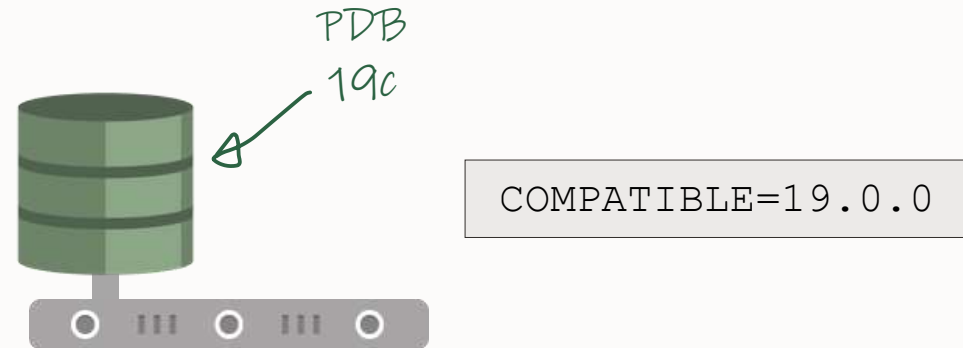
options



Fallback | Lower Release Non-CDB

Source database: 11.2.0.4 non-CDB

- Upgrade
- Plug-in
- Convert



Pro tip: Reduce downtime
with Oracle GoldenGate

Fallback | Lower Release Non-CDB

Source database: 11.2.0.4 non-CDB

- Upgrade
- Plug-in
- Convert



Pro tip: Reduce downtime
with Oracle GoldenGate

Fallback | Lower Release Non-CDB

Source database: 11.2.0.4 non-CDB

- Upgrade
- Plug-in
- Convert



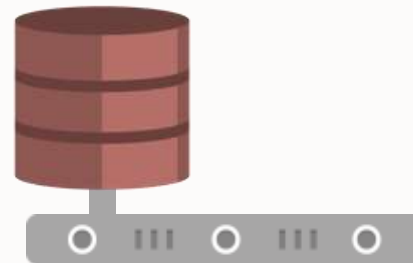
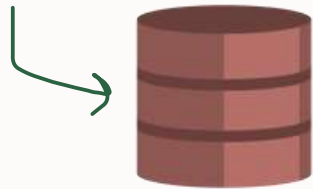
Pro tip: Reduce downtime
with Oracle GoldenGate

Fallback | Lower Release Non-CDB

Source database: 11.2.0.4 non-CDB

- Upgrade
- Plug-in
- Convert

New 11.2.0.4
non-CDB



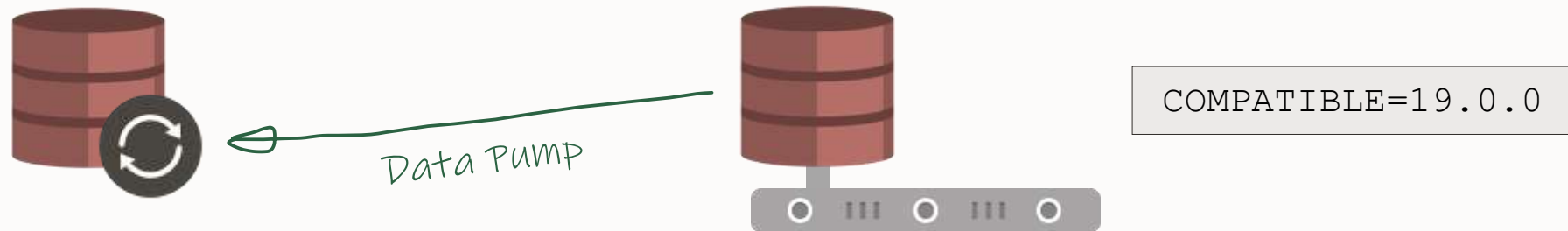
COMPATIBLE=19.0.0

Pro tip: Reduce downtime
with Oracle GoldenGate

Fallback | Lower Release Non-CDB

Source database: 11.2.0.4 non-CDB

- Upgrade
- Plug-in
- Convert



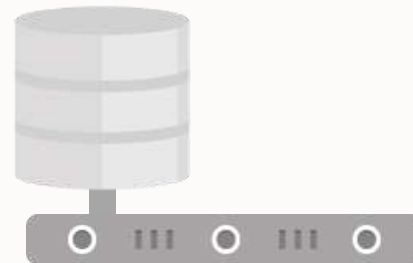
```
expdp ... version=11.2.0.4
```

Pro tip: Reduce downtime
with Oracle GoldenGate

Fallback | Lower Release Non-CDB

Source database: 11.2.0.4 non-CDB

- Upgrade
- Plug-in
- Convert



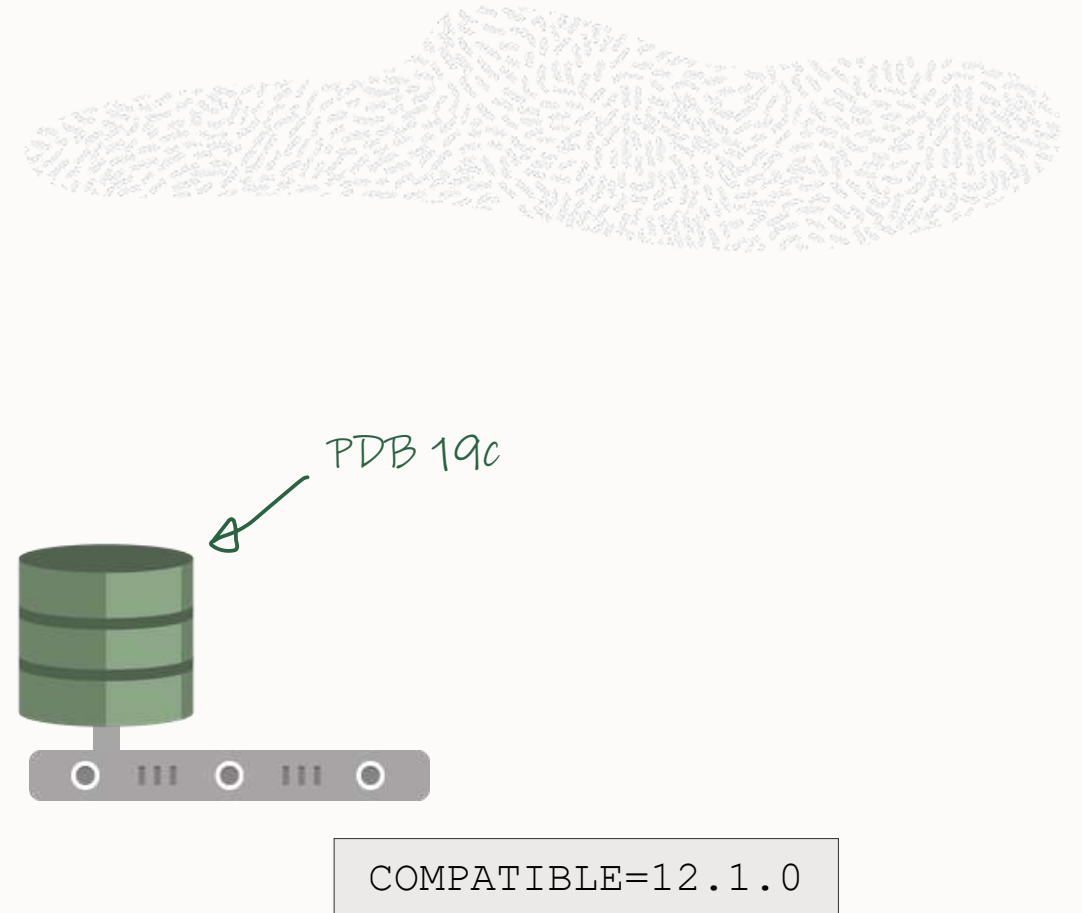
COMPATIBLE=19.0.0

Pro tip: Reduce downtime
with Oracle GoldenGate

Fallback | Lower Release CDB

Source database: 12.1 non-CDB

- Upgrade
- Plug-in
- Convert



Fallback | Lower Release CDB

Source database: 12.1 non-CDB

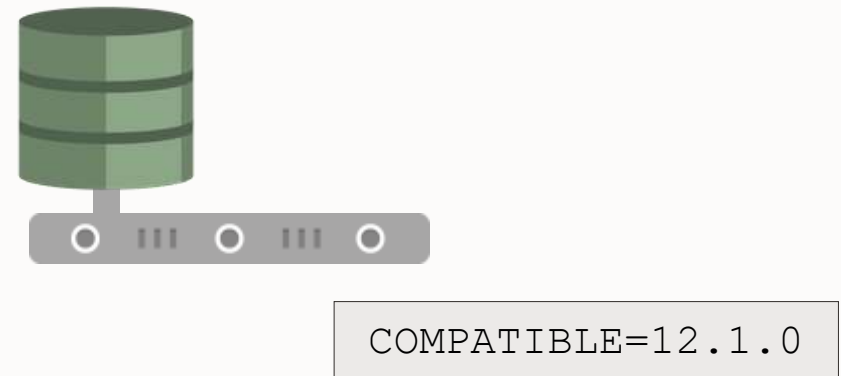
- Upgrade
- Plug-in
- Convert



Fallback | Lower Release CDB

Source database: 12.1 non-CDB

- Upgrade
- Plug-in
- Convert



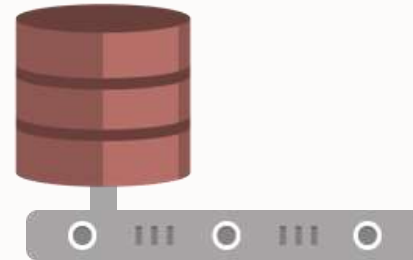
Fallback | Lower Release CDB

Source database: 12.1 non-CDB

- Upgrade
- Plug-in
- Convert



Downtime

A green curved arrow pointing from the word 'Downtime' to the target database instance.

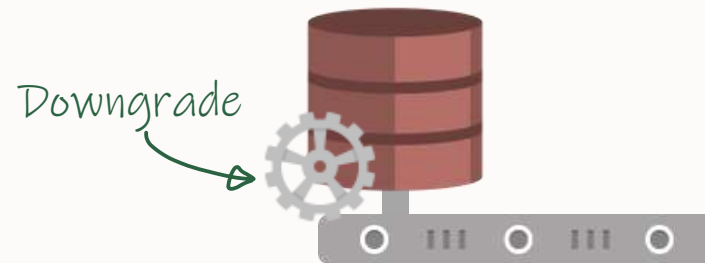
COMPATIBLE=12.1.0



Fallback | Lower Release CDB

Source database: 12.1 non-CDB

- Upgrade
- Plug-in
- Convert

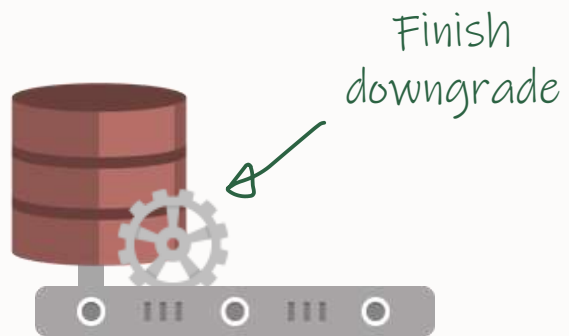


COMPATIBLE=12.1.0

Fallback | Lower Release CDB

Source database: 12.1 non-CDB

- Upgrade
- Plug-in
- Convert

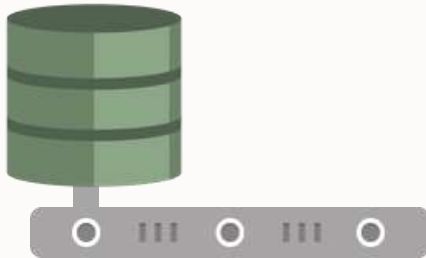


COMPATIBLE=12.1.0

Fallback | Lower Release CDB

Source database: 12.1 non-CDB

- Upgrade
- Plug-in
- Convert



COMPATIBLE=12.1.0

Fallback| PDB Downgrade

Downgrade works for CDB/PDB entirely as well as for single/multiple PDBs

- Manual tasks
 - `catdwgrd.sql` in current (*after* upgrade) environment
 - `catrelod.sql` in previous (*before* upgrade) environment
 - Don't change `COMPATIBLE`
- `datapatch` must roll back SPUs/PSUs/BPs manually
- [MOS Note: 2172185.1](#)
[How to Downgrade a Single Pluggable Oracle Database \(PDB \) to previous release](#)



Fallback | Summary

Move to Multitenant architecture = Migration

1. Usual fallback techniques doesn't work

- No downgrade
- No flashback to Restore Point

2. Only possible fallback options

- Data Pump
- Transportable Tablespaces (only same version)
- GoldenGate
- Plug into a source-version CDB
- Be very **cautious** about COMPATIBLE



Migration | Last Words

Every migration

- Is an architectural change
- Requires downtime
- Requires a fallback
- Ends with a backup



Chapter 3

Focus on the Details



Multitenant and Data Guard

Data Guard | Migration Options

It is possible to **preserve** the standby database when you migrate from non-CDB to PDB

Special **attention** is needed

You don't have to rebuild your standby database, but you might find it is the **easiest** solution

Data Guard | Migration Options

If you CDB has a standby database,
you must follow these **guidelines**

If you don't, you will crash the MRP process
and **redo apply stops**

preserve

STANDBY

database

GoldenGate

Transportable

Data Pump

Plug-in Copy

Plug-in NoCopy



Data Guard | **Plug-in**

Three options

1. Re-use data files

- **PDB is immediately protected**

2. Defer creation of PDB on standby

- PDB is protected as soon as data files are restored on standby

3. Create or recreate standby database

- CDB is protected when standby database is ready



Data Guard Example | Initial Setup



Oracle 12.1.0.2
PROD
10 TB size

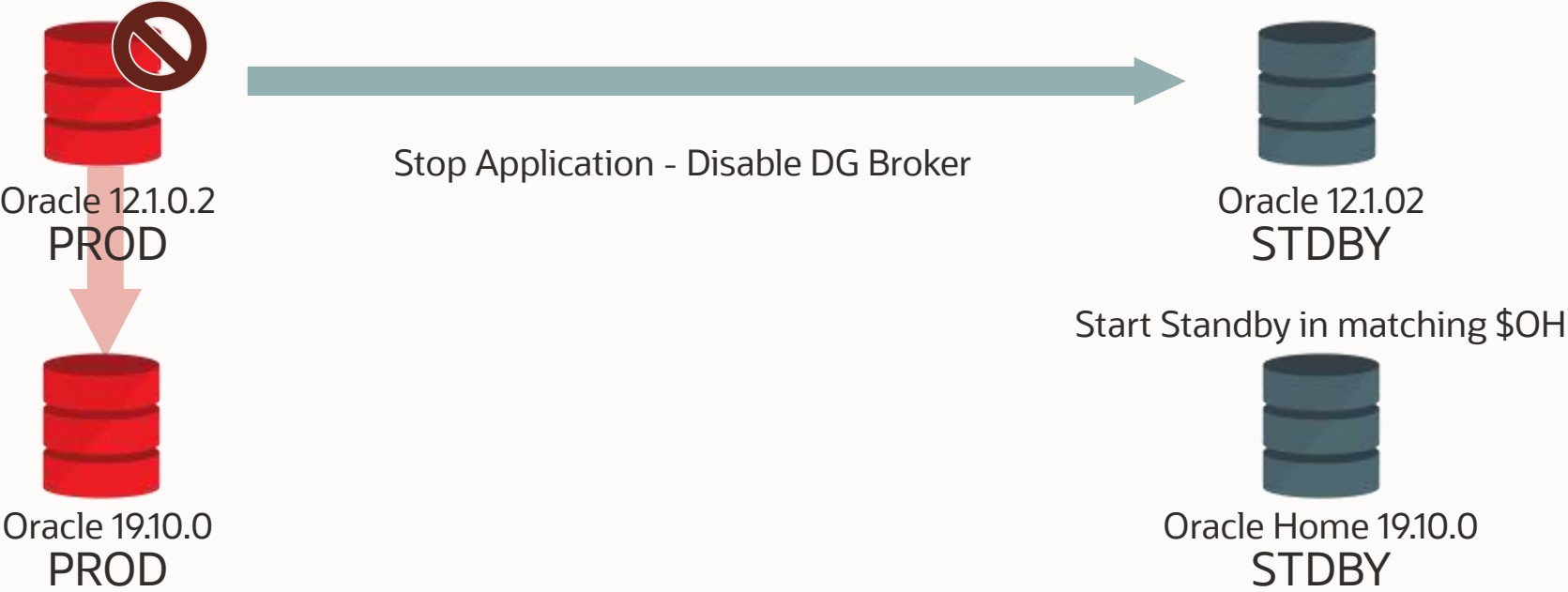
100 km distance Maximum Protection



Oracle 12.1.0.2
STDBY
10 TB size

Data Guard Example | AutoUpgrade

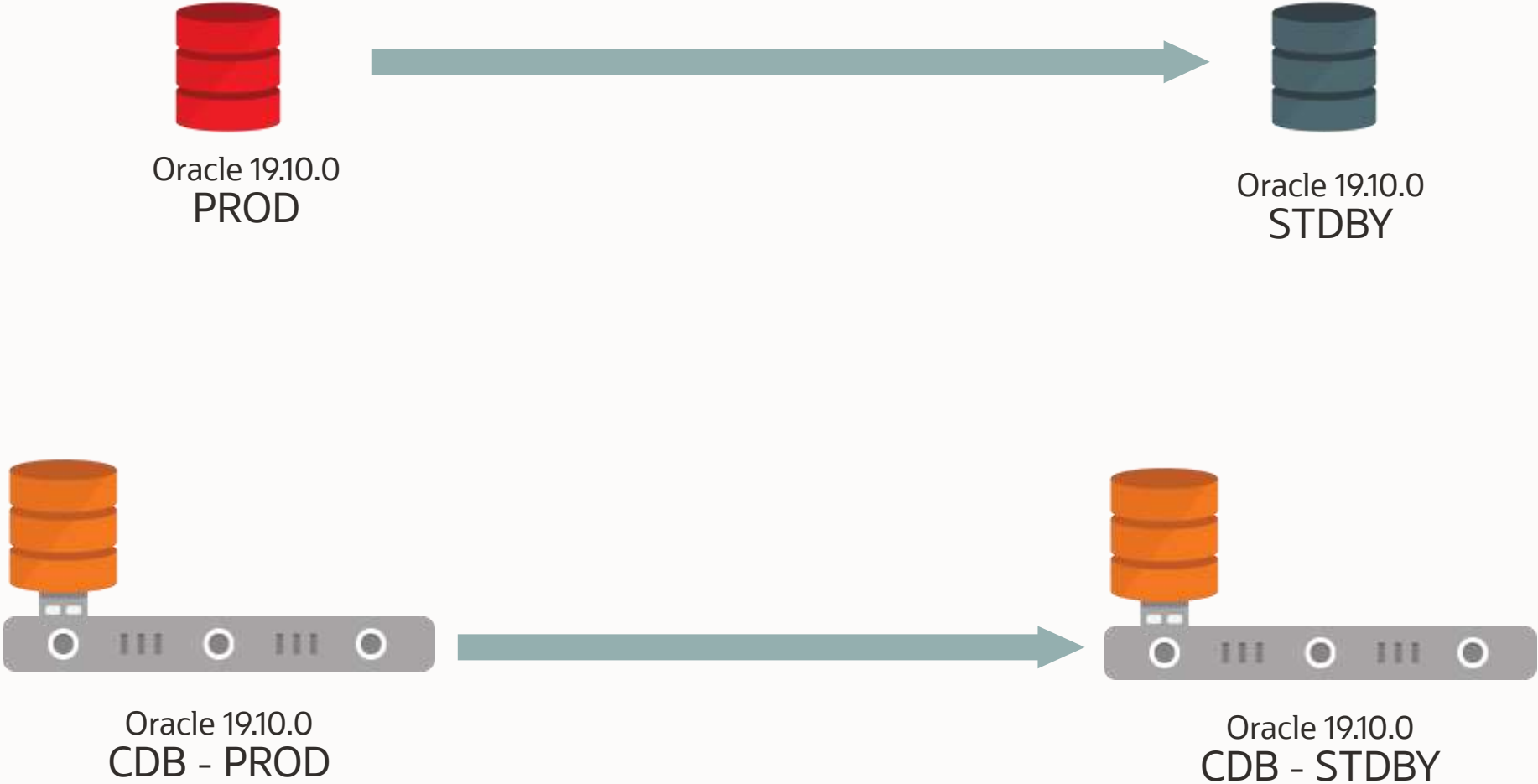
>> DOWNTIME <<



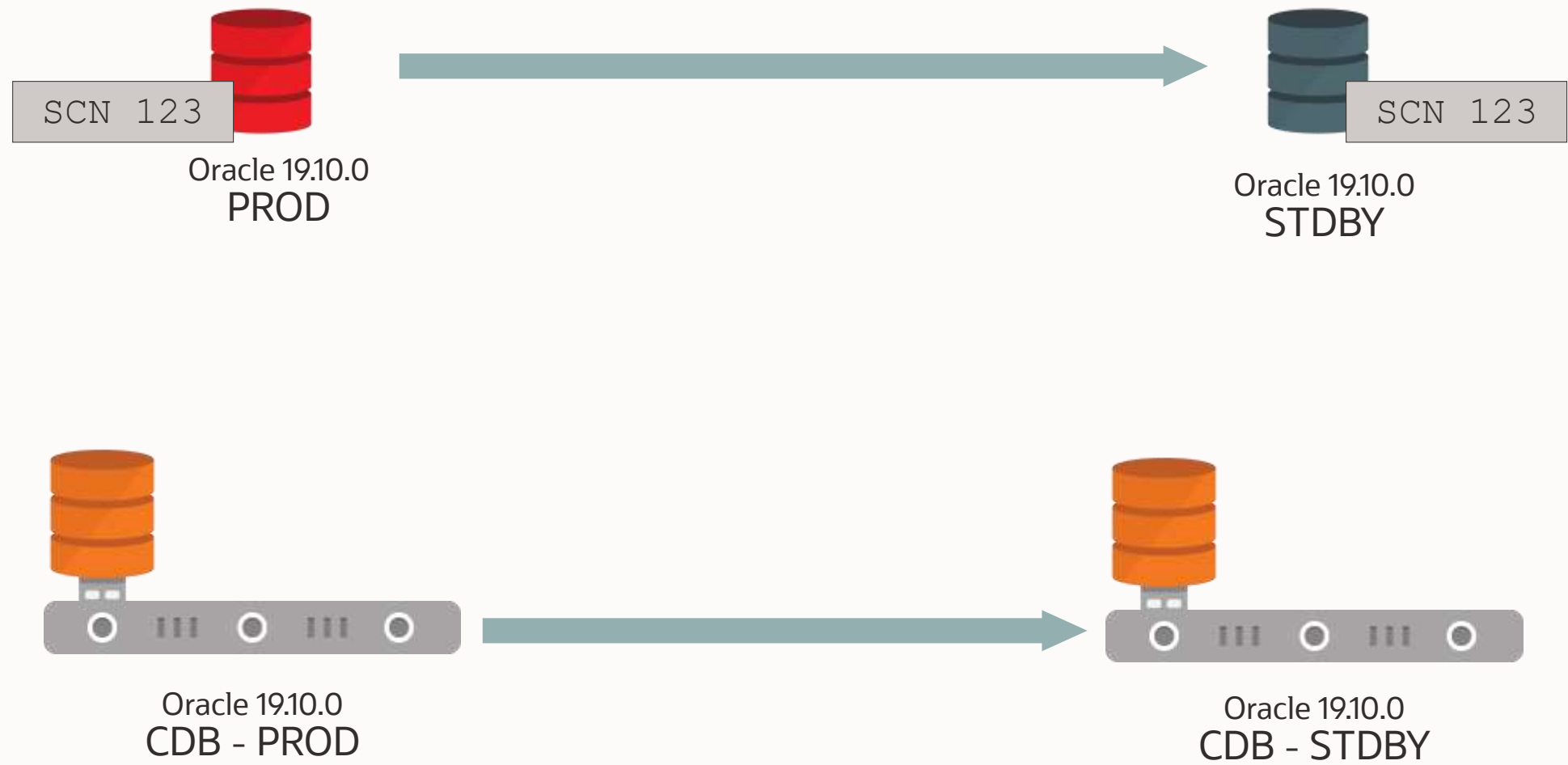
Data Guard Example | **Implicit Standby Upgrade**



Data Guard Example | Create CDBs



Data Guard Example | Synchronize



Data Guard Example | Synchronize

[MOS Doc ID 2273304.1](#)

Primary

Standby

```
DGMGRL> edit database stdby set state='APPLY-OFF';  
SQL> shutdown immediate  
SQL> startup  
SQL> alter database  
       recover managed standby database cancel;
```

Data Guard Example | Synchronize

[MOS Doc ID 2273304.1](#)

Primary

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter system
      flush redo to stdby no confirm apply;
SQL> alter database open read only;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

Standby

```
DGMGRL> edit database stdby set state='APPLY-OFF';
SQL> shutdown immediate
SQL> startup
SQL> alter database
      recover managed standby database cancel;
```


Data Guard Example | Synchronize

[MOS Doc ID 2273304.1](#)

Primary

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter system
      flush redo to stdby no confirm apply;
SQL> alter database open read only;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

Standby

```
DGMGRL> edit database stdby set state='APPLY-OFF';
SQL> shutdown immediate
SQL> startup
SQL> alter database
      recover managed standby database cancel;
```

```
SQL> alter database recover managed standby database
      until change 2319950;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

Data Guard Example | Synchronize

[MOS Doc ID 2273304.1](#)

Primary

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter system
      flush redo to stdby no confirm apply;
SQL> alter database open read only;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

Standby

```
DGMGRL> edit database stdby set state='APPLY-OFF';
SQL> shutdown immediate
SQL> startup
SQL> alter database
      recover managed standby database cancel;
```

Replace with checkpoint_change#

```
SQL> alter database recover managed standby database
      until change 2319950;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

MUST MATCH!

Data Guard Example | Synchronize

[MOS Doc ID 2273304.1](#)

Primary

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter system
      flush redo to stdby no confirm apply;
SQL> alter database open read only;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

```
SQL> exec dbms_pdb.describe('/home/oracle/prmy.xml');
SQL> shutdown immediate
```

Standby

```
DGMGRL> edit database stdby set state='APPLY-OFF';
SQL> shutdown immediate
SQL> startup
SQL> alter database
      recover managed standby database cancel;
```

```
SQL> alter database recover managed standby database
      until change 2319950;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

Data Guard Example | Synchronize

[MOS Doc ID 2273304.1](#)

Primary

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter system
      flush redo to stdby no confirm apply;
SQL> alter database open read only;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

```
SQL> exec dbms_pdb.describe('/home/oracle/prmy.xml');
SQL> shutdown immediate
```

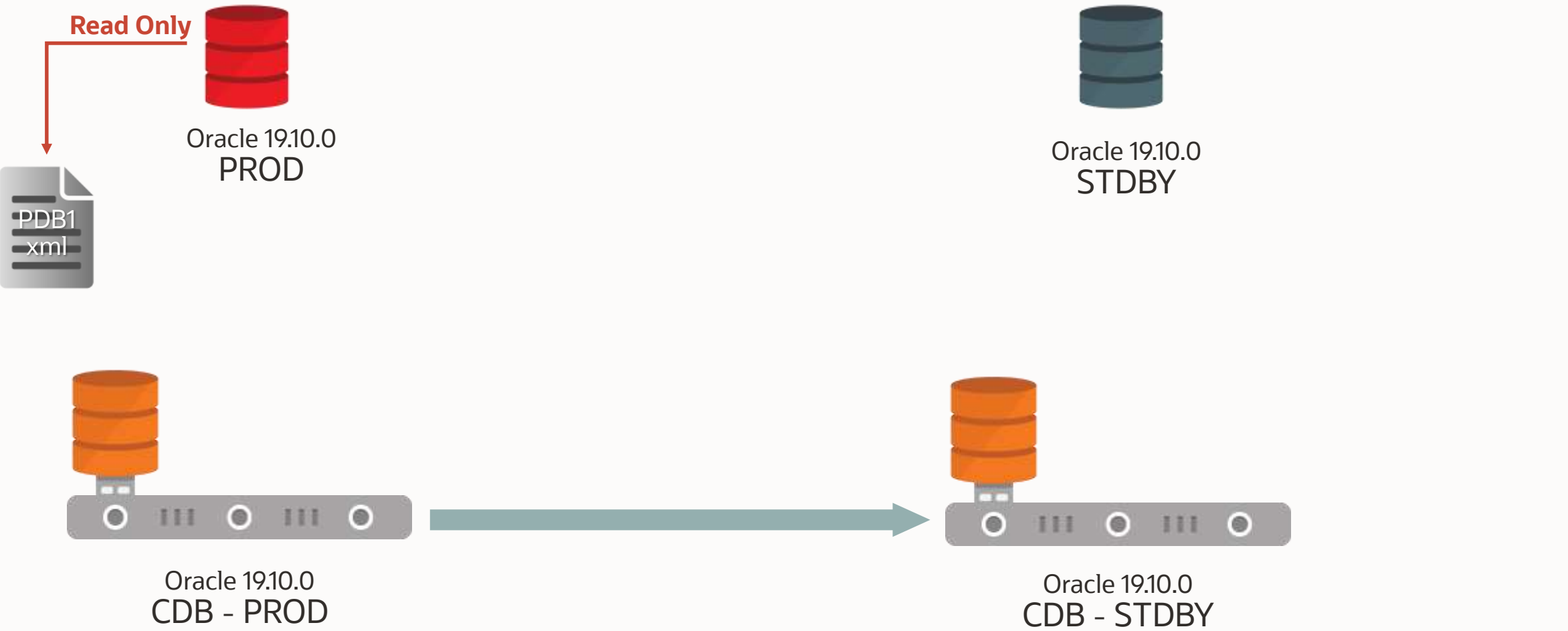
Standby

```
DGMGRL> edit database stdby set state='APPLY-OFF';
SQL> shutdown immediate
SQL> startup
SQL> alter database
      recover managed standby database cancel;
```

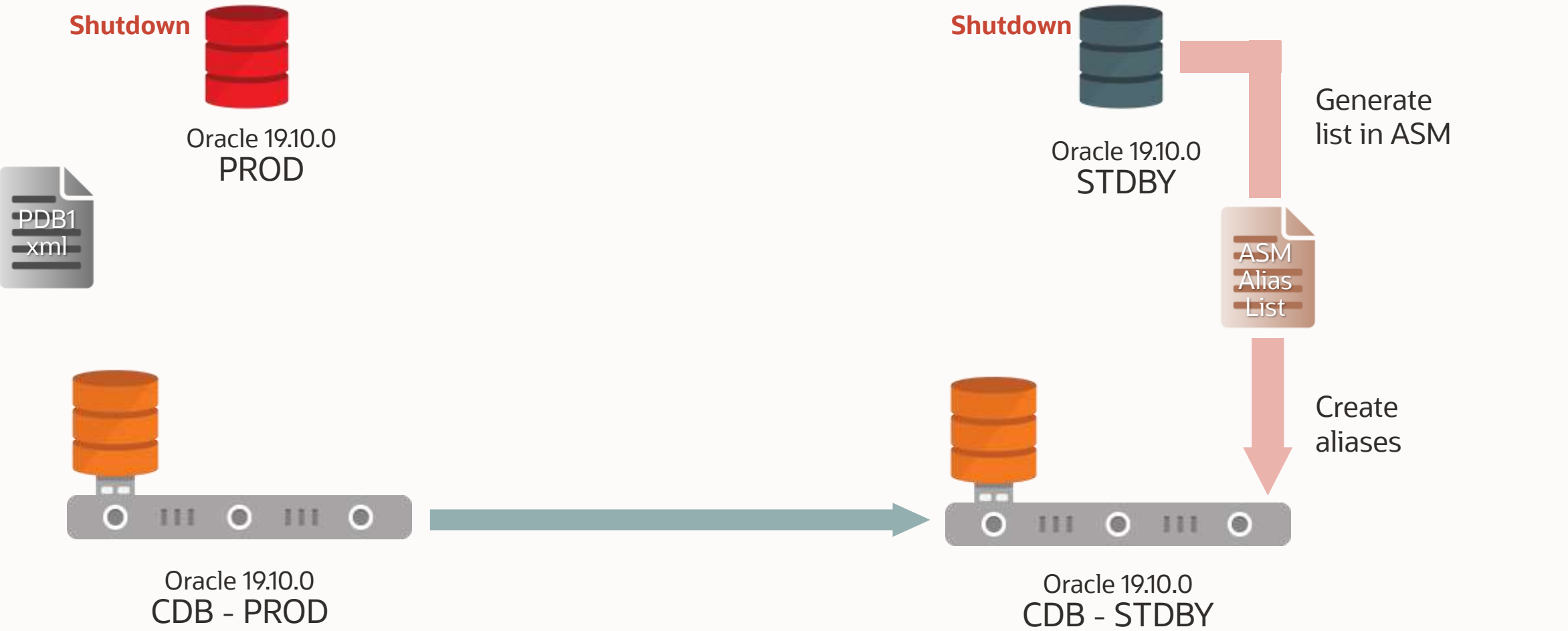
```
SQL> alter database recover managed standby database
      until change 2319950;
SQL> select checkpoint_change#
      from v$datafile_header where file#=1;
```

```
SQL> shutdown immediate
```

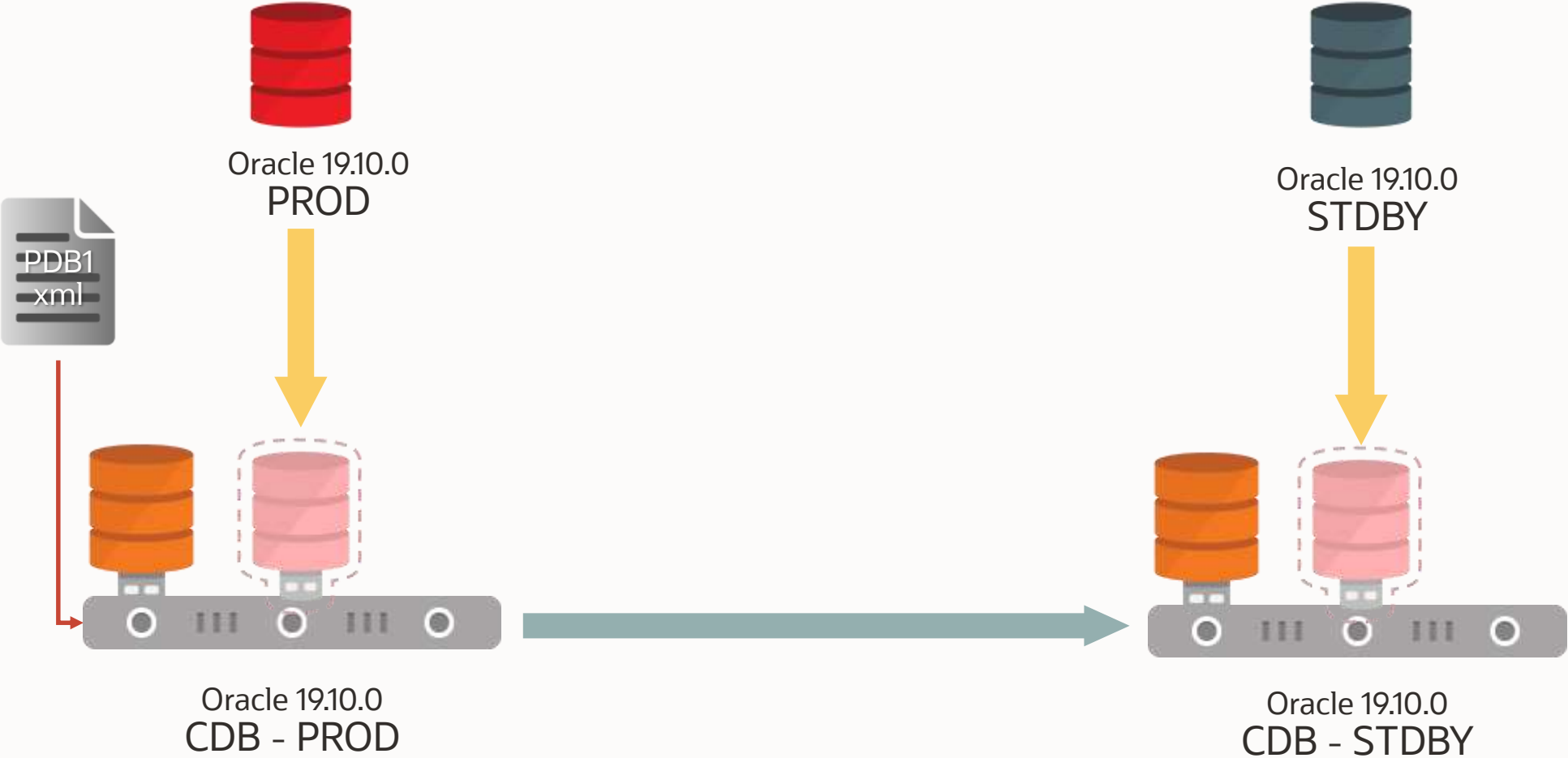
Data Guard Example | Read Only Phase



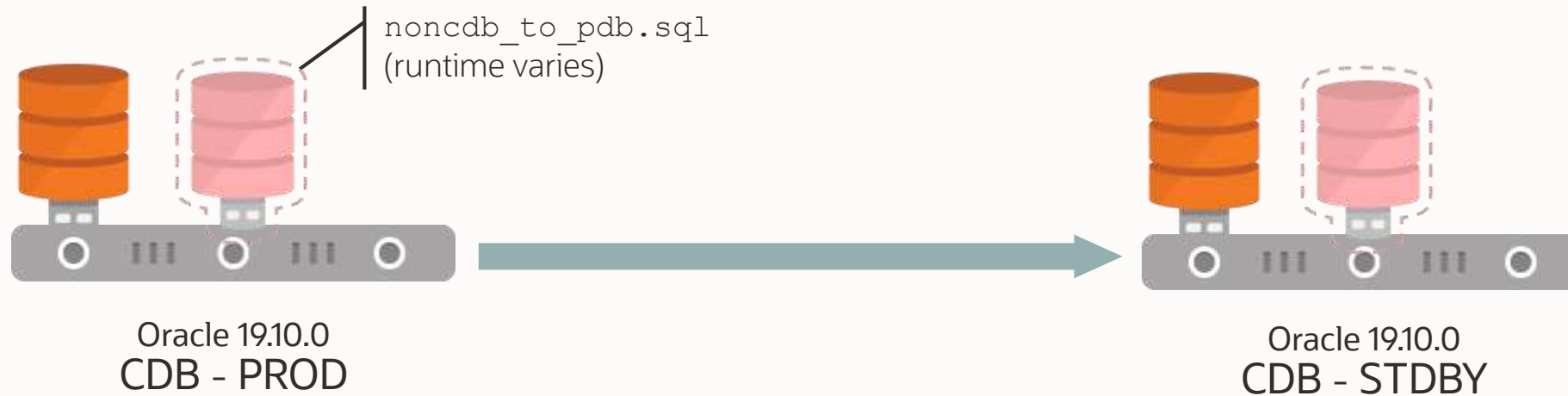
Data Guard Example | **ASM Alias File**



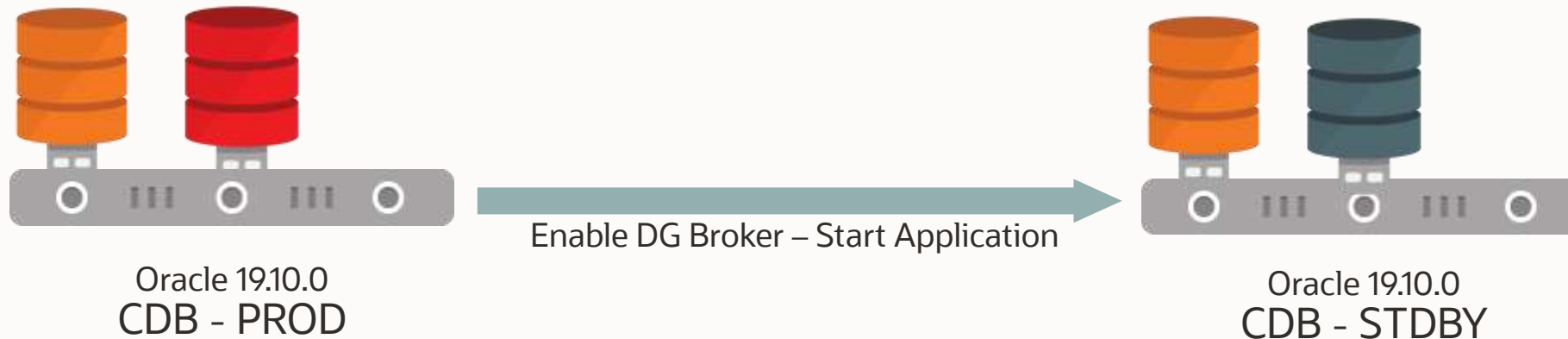
Data Guard Example | Plugin



Data Guard Example | **Convert non-CDB to PDB**



Data Guard Example | **Plugin Completed!**



Data Guard | Preserve Data Files

- Data Guard PM - Pieter van Puymbroeck:
<https://vanpupi.stepi.net/2019/06/to-cdb-or-not-to-cdb-thats-the-question/>
- MAA Team:
[MOS Note: 2273304.1](#)
[Reusing the Source Standby Database Files When Plugging a non-CDB as a PDB into the Primary Database of a Data Guard Configuration](#)
- Explanation and troubleshooting
<https://dohdatabase.com/2020/12/03/upgrade-plug-in-with-asm-data-guard-tde-and-no-keystore-password/>

Data Guard | **Worth to Mention**

- Create TEMP files on standby PDB
 - Not added automatically
 - Open standby in read-only and add temp files
 - [Data Guard Physical Standby - Managing temporary tablespace tempfiles \(Doc ID 1514588.1\)](#)



Data Guard | **Worth to Mention**

- When you plug in a non-CDB the GUID doesn't change
 - Get GUID from manifest file or `V$CONTAINERS`
 - GUID only changes when you use `AS CLONE` keyword
- Impossible to preserve standby data files when
 - Using OMF and `CREATE PLUGGABLE DATABASE ... AS CLONE`
 - Regardless of whether you use regular file system or ASM
 - Only works with regular file system and non-OMF



Data Guard | **Worth to Mention**

- Regular file system and OMF
 - Put standby data files in OMF location
`<DB_CREATE_FILE_DEST>/<STDBY_DB_UNIQUE_NAME>/<GUID>/datafile`
 - Also set `DB_FILE_NAME_CONVERT` to convert primary file path to standby file path, e.g.
`DB_FILE_NAME_CONVERT='PRMY','STDBY'`
- Regular file system and non-OMF
 - Put data files at the same location as primary data files
 - Take `FILE_NAME_CONVERT` into account (`CREATE PLUGGABLE DATABASE`)
 - Adjust according to `DB_FILE_NAME_CONVERT`

Data Guard | **Plug-in**

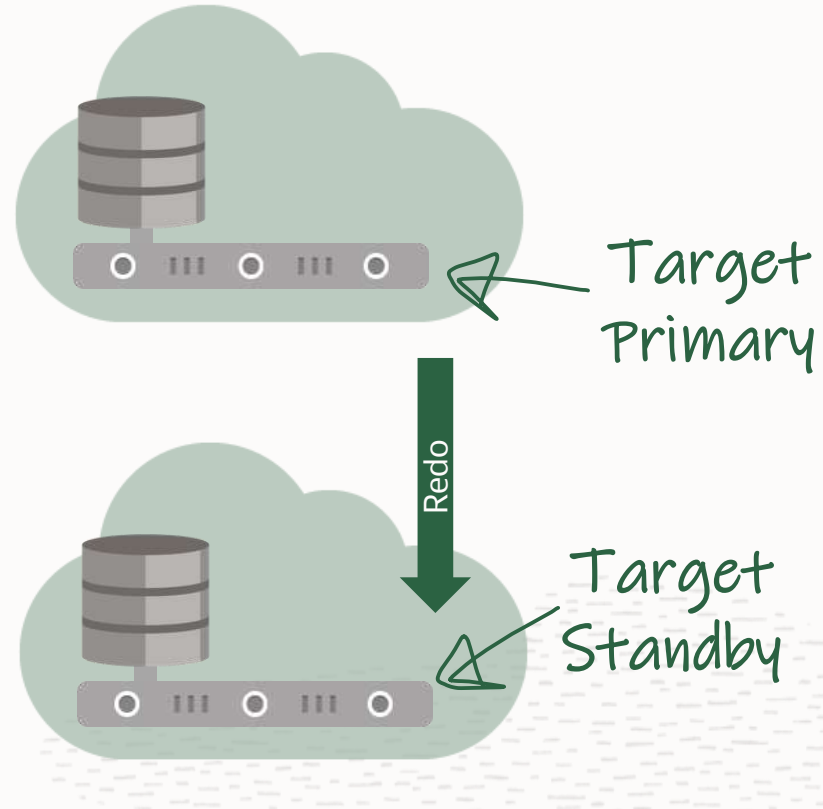
Three options

1. Re-use data files
 - PDB is immediately protected
- 2. Defer creation of PDB on standby**
 - PDB is protected as soon as data files are restored on standby**
3. Create or recreate standby database
 - CDB is protected when standby database is ready

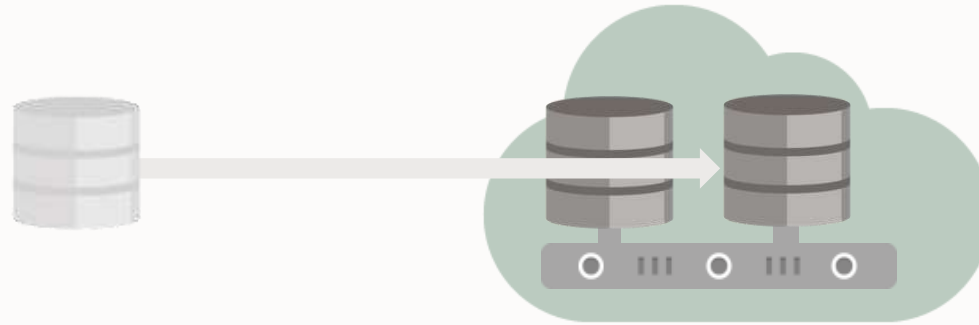


Data Guard | **Plug-in**

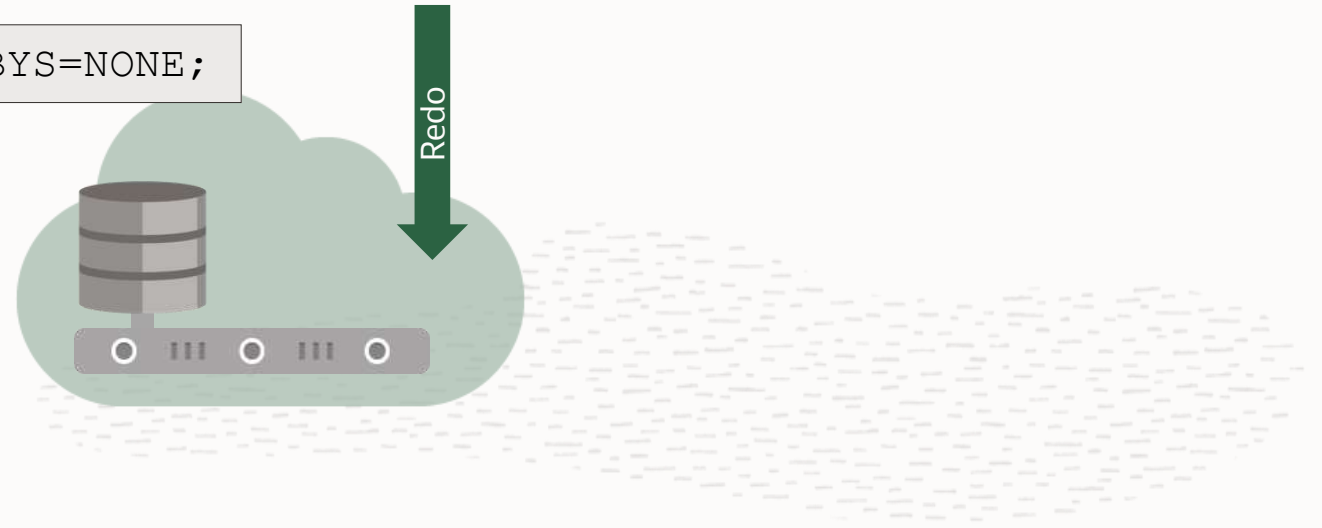
Source



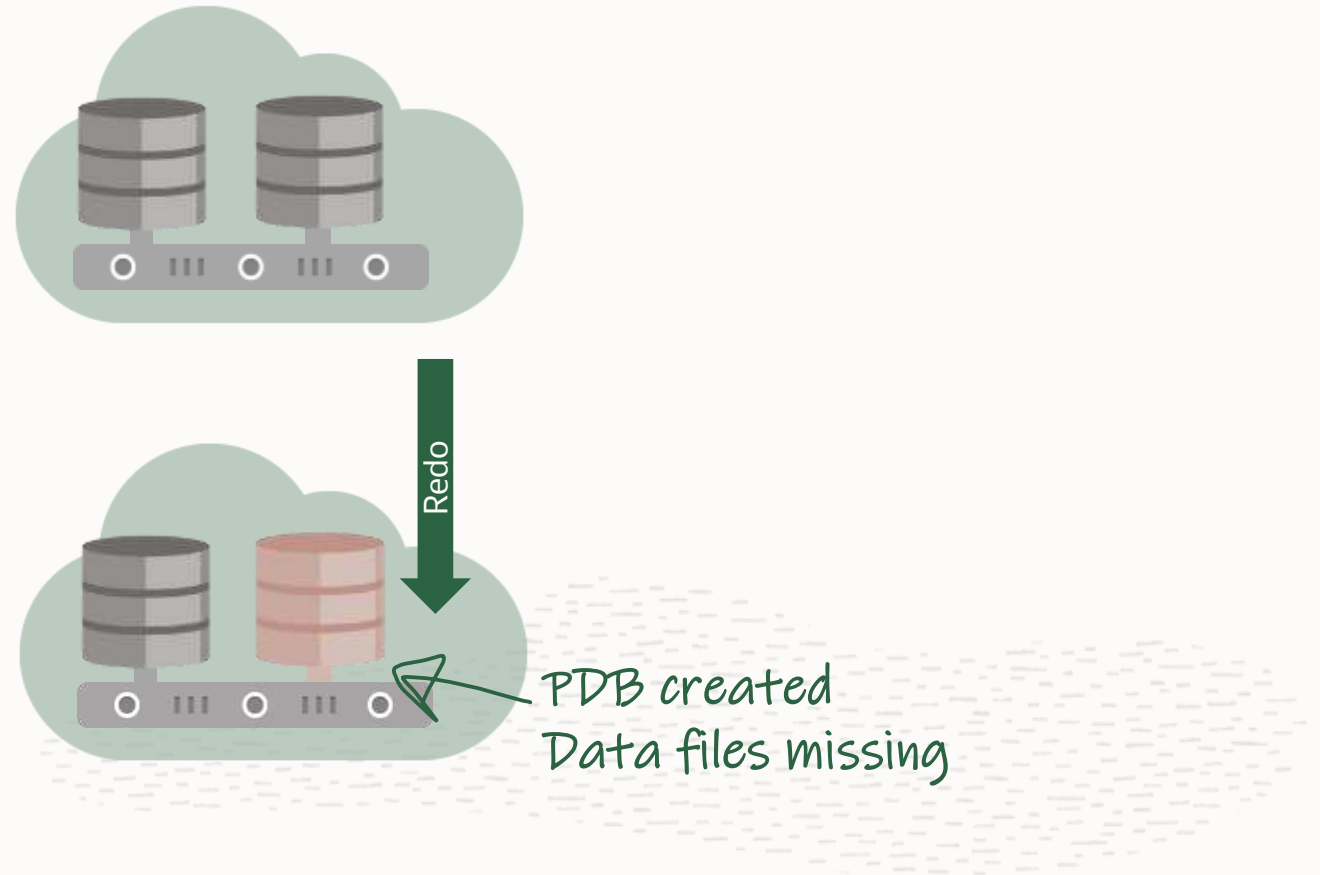
Data Guard | **Plug-in**



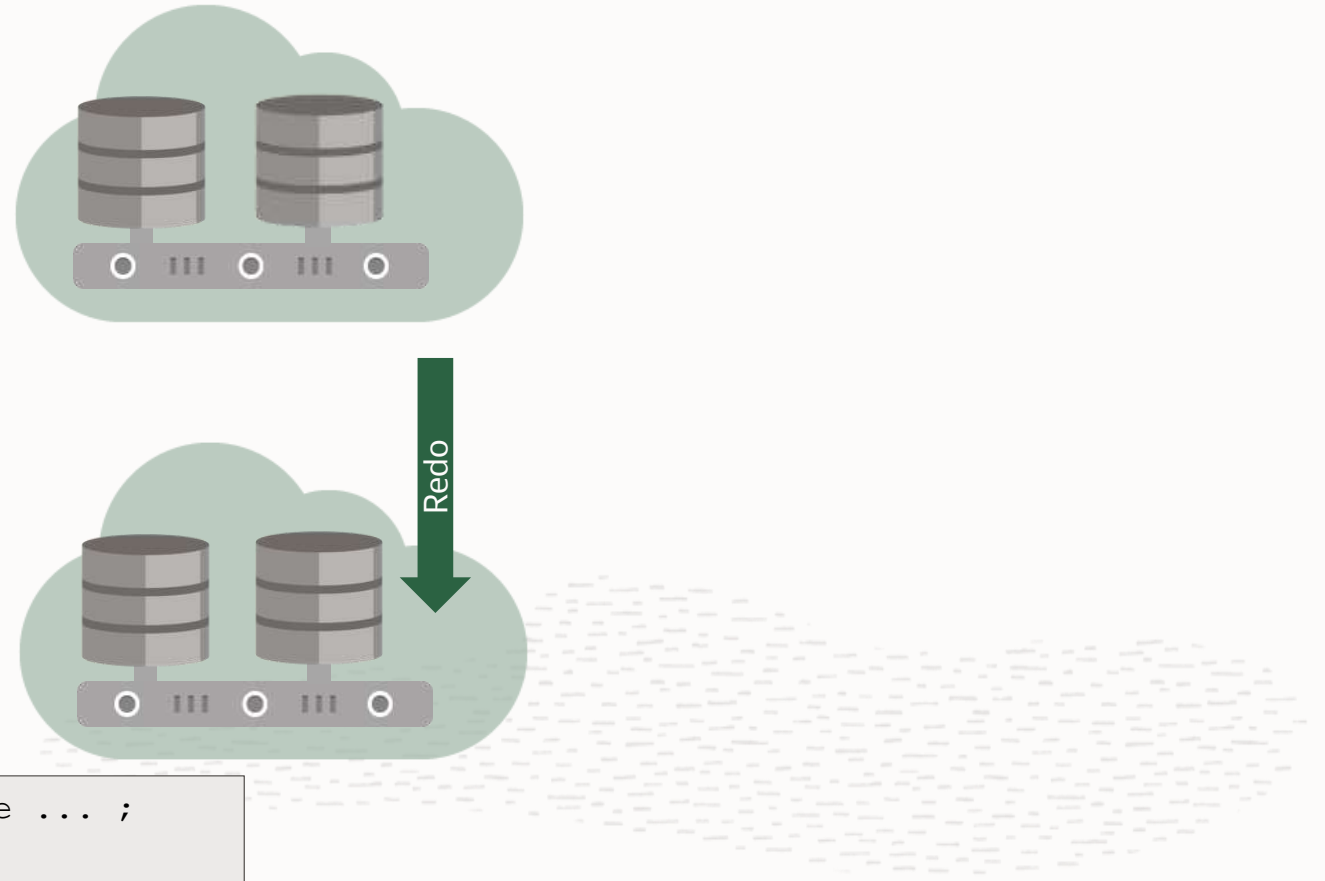
```
SQL> CREATE PLUGGABLE DATASE ... STANDBYS=NONE;
```



Data Guard | Plug-in



Data Guard | Plug-in



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

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 \(Doc ID 1916648.1\)](#)

Data Guard | **Plug-in**

Three options

1. Re-use data files
 - PDB is immediately protected
2. Defer creation of PDB on standby
 - PDB is protected as soon as data files are restored on standby
- 3. Create or recreate standby database**
 - CDB is protected when standby database is ready**

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.

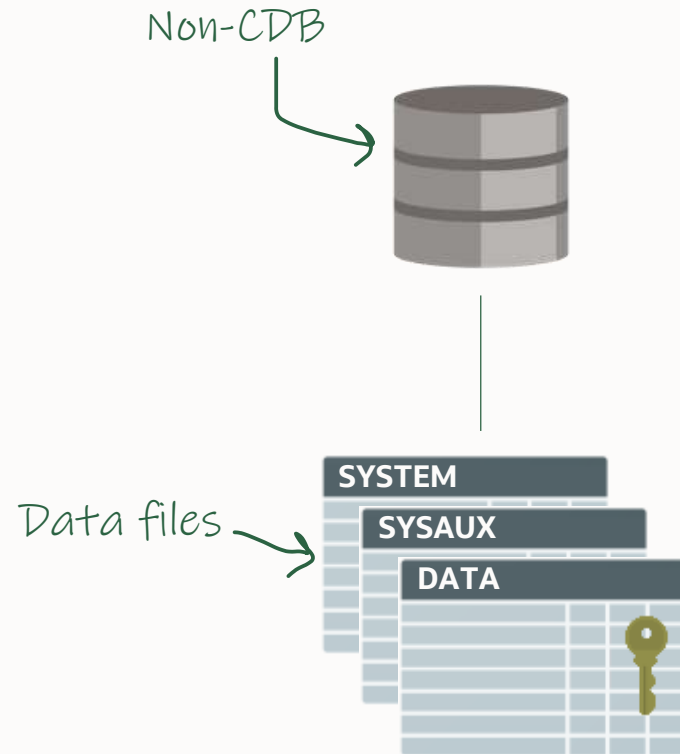


Photo by [John Salvino](#) on [Unsplash](#)

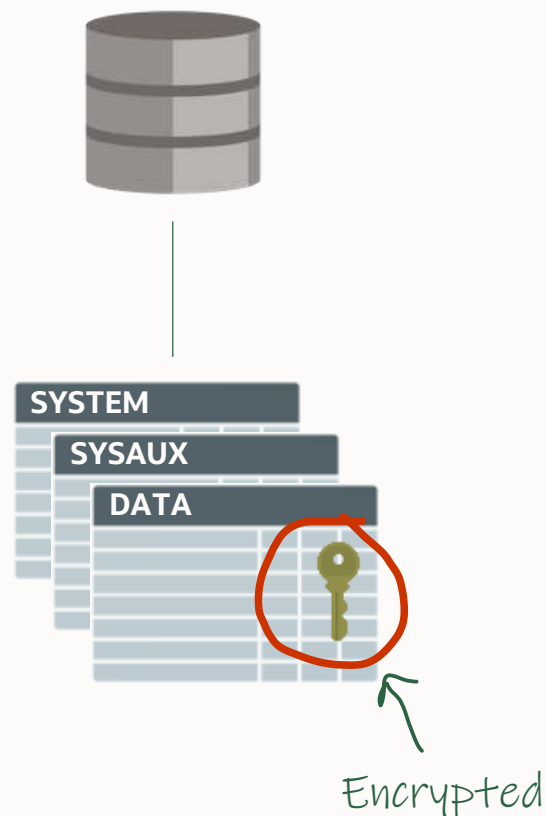
Multitenant

TDE Tablespace Encryption

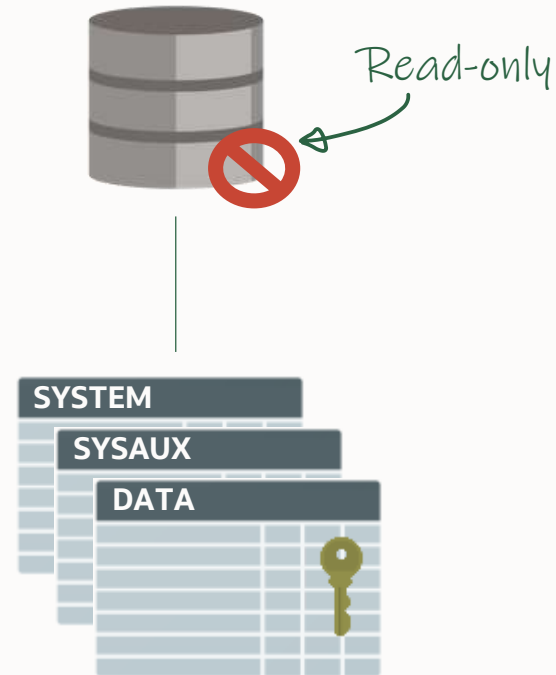
TDE | Migrating Encrypted Database



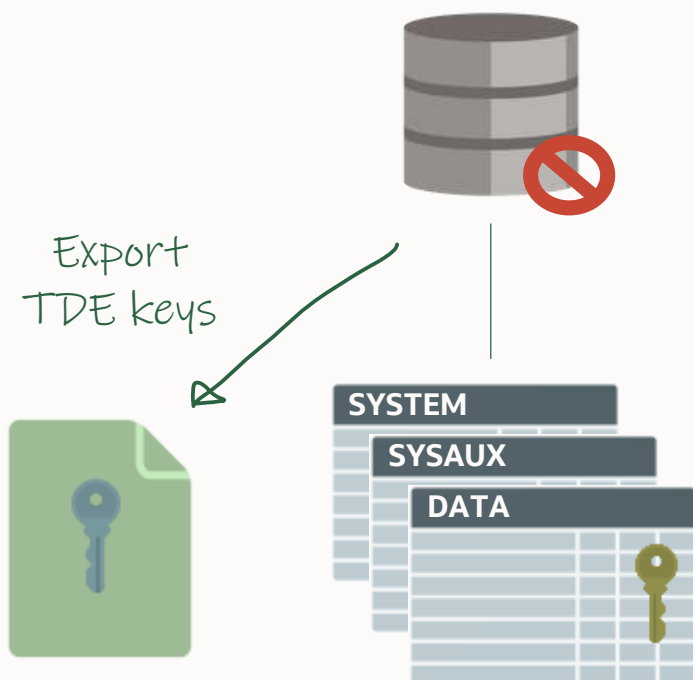
TDE | Migrating Encrypted Database



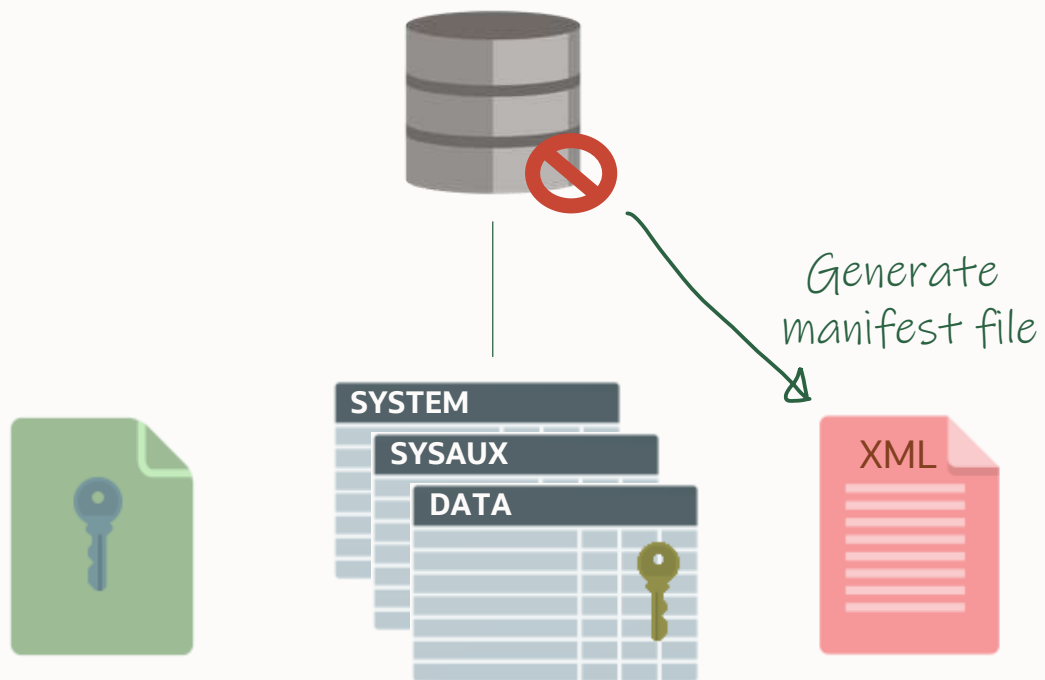
TDE | Migrating Encrypted Database



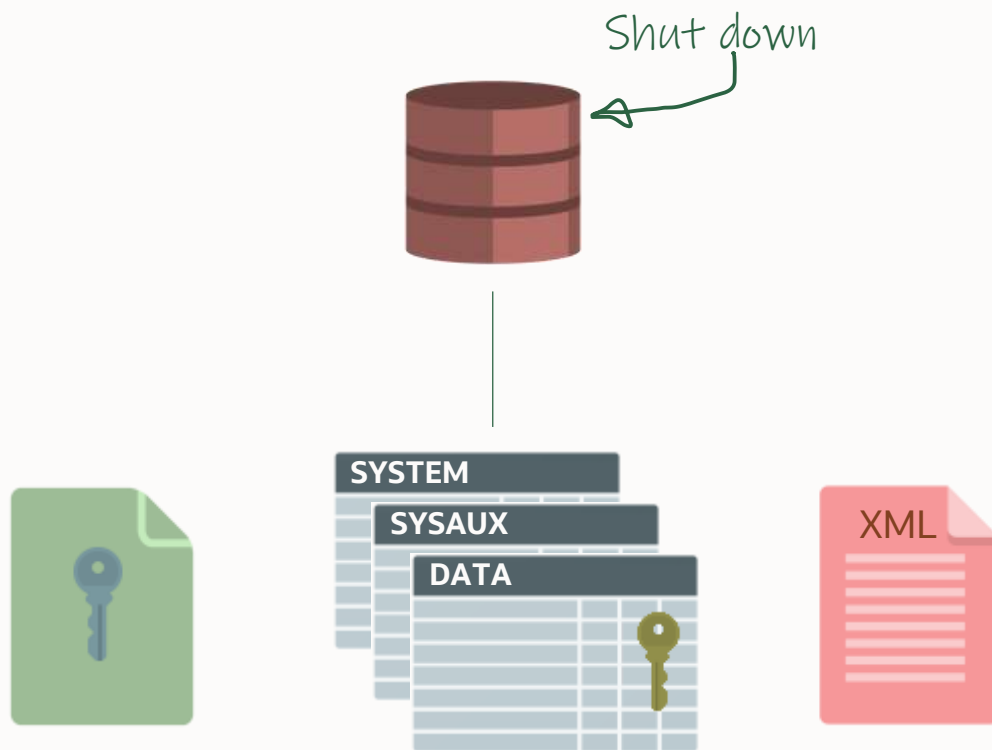
TDE | Migrating Encrypted Database



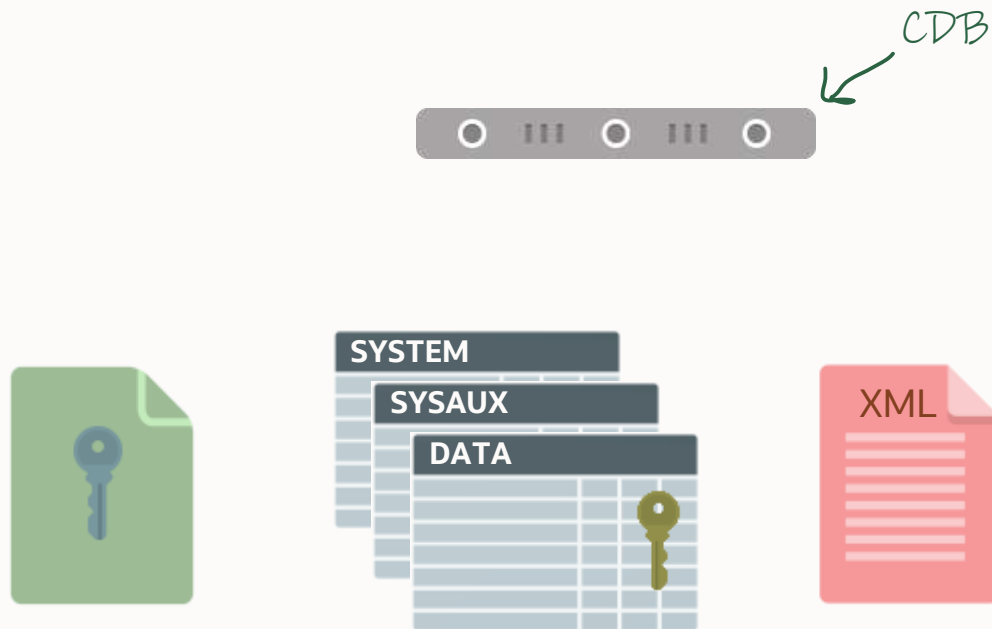
TDE | Migrating Encrypted Database



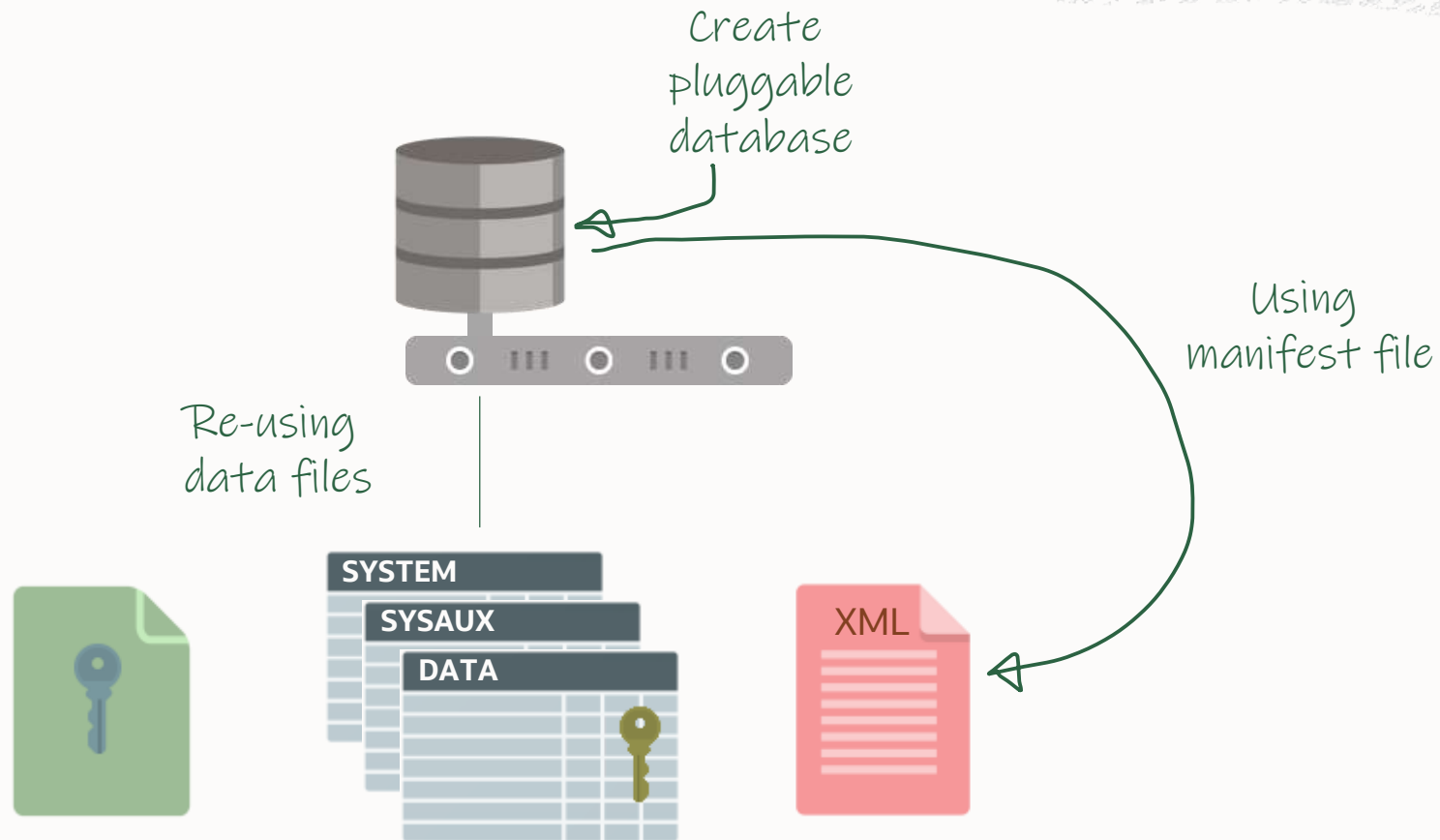
TDE | Migrating Encrypted Database



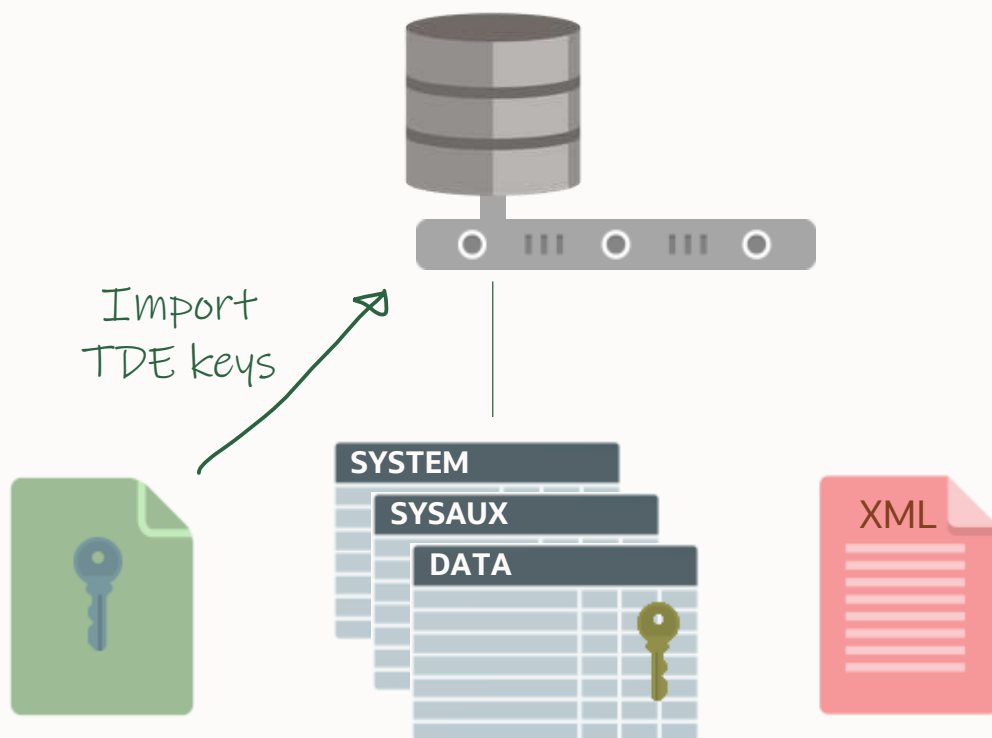
TDE | Migrating Encrypted Database



TDE | Migrating Encrypted Database



TDE | Migrating Encrypted Database



TDE | Migrating Encrypted Database



Export keys before generating manifest file

```
SQL> administer key management export keys
      with secret "secret-passphrase"
      to '/tmp/db19-exported-keys'
      force keystore identified by "S3cr3t";
SQL> exec dbms_pdb.describe('/tmp/DB19.xml');
```

After creating the PDB import encryption keys:

```
SQL> create pluggable database DB19 using '/tmp/DB19.xml' copy tempfile reuse;
SQL> alter pluggable database DB19 open;
SQL> alter session set container=DB19;
SQL> administer key management import keys
      with secret "secret-passphrase"
      from '/tmp/exported-keys'
      force keystore identified by "S3cr3t" with backup;
SQL> @?/rdbms/admin/noncdb_to_pdb.sql
```

TDE | Nice to know



Always use the latest Release Update

- For 19c, use 19.10.0
- Be aware of bug 31494095: NONCDB_TO_PDB.SQL FAILS WITH ORA-01219: DATABASE OR PLUGGABLE DATABASE NOT OPEN: QUERIES ALLOWED ON FIXED TABLES OR VIEWS ONLY

[How to migrate a non pluggable database that uses TDE to pluggable database ? \(Doc ID 1678525.1\)](#)

TDE | Migration and Fallback Options



Data Pump

- Full support for TDE
- Export from and import into encrypted databases
- Dump files can be encrypted as well
- Use ENCRYPTION PWD PROMPT to avoid typing password in cleartext on command line

Transportable Tablespaces

- Same endian migration: Full support for TDE
- Use `ENCRYPTION_PWD_PROMPT`
- Cross endian migration: No support for TDE

GoldenGate

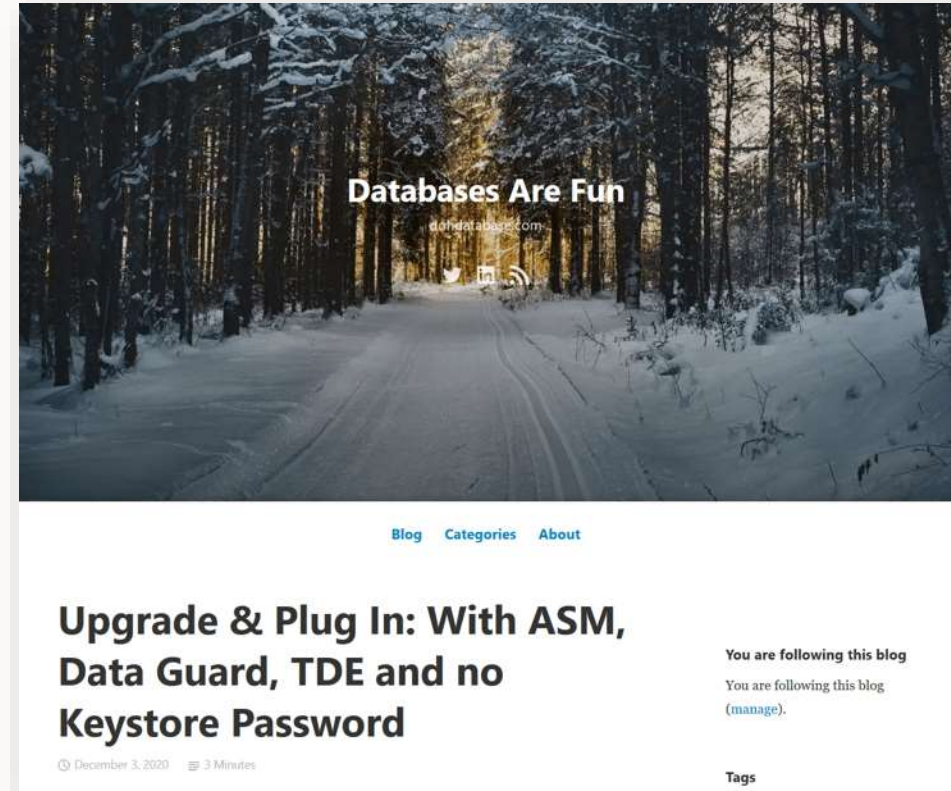
- Full support for TDE

TDE | Full House

Interesting case - **upgrade and plug in:**

- ASM
- Data Guard
- TDE Tablespace Encryption
- No Keystore Password (separation of duties)

Details in [blog post](#)



TDE | Cloning PDBs

Parameter ONE STEP PLUGIN FOR PDB WITH TDE

Allow you to clone a PDB via database link
without specify (or knowing) the keystore password

Sounds like a good idea. **Superseded** functionality!



TDE | Keystore External Password Store

Recommended by Security Product Managers

Why? With encrypted PDBs:

```
SQL> CREATE PLUGGABLE DATABASE DB19 ... IDENTIFIED BY "S3cr3t" ... ;
```

Better option






```
SQL> CREATE PLUGGABLE DATABASE DB19 ... IDENTIFIED BY EXTERNAL STORE ...;
```

YouTube [video](#) and [blog post](#)




https://MikeDietrichDE.com

https://DOHdatabase.com


Upgrade your Database - NOW!     

[Blog](#) [Slides](#) [Hands-On Lab](#) [Events](#) [Papers / Docs](#) [Videos](#) [Scripts](#) [Links](#) [Oracle Documentation](#) [Privacy Policy](#) [About](#)


Upgrade to Oracle 19...




Test Transportable Ta...



User Group Seminars...



[<](#) [>](#)



Which database version can be used with Grid Infrastructure 19c?




Posted on October 6, 2020 by Mike.Dietrich [ASM / RAC / GI](#)

This will be a very short blog post today. But a colleague asked me this question just a few minutes ago: Which database version can be used with Grid Infrastructure 19c? And I realized that I neither put it on the blog yet nor did I bookmark it. But I answered this question several times already.

Which database version

Databases Are Fun

dohdatabase.com

[Blog](#) [Categories](#) [About](#)


Blog

Debut on the Big Stage

In mid-October you can join our two webinars on database migration. One of them is exclusively about cloud migration — for techies only.

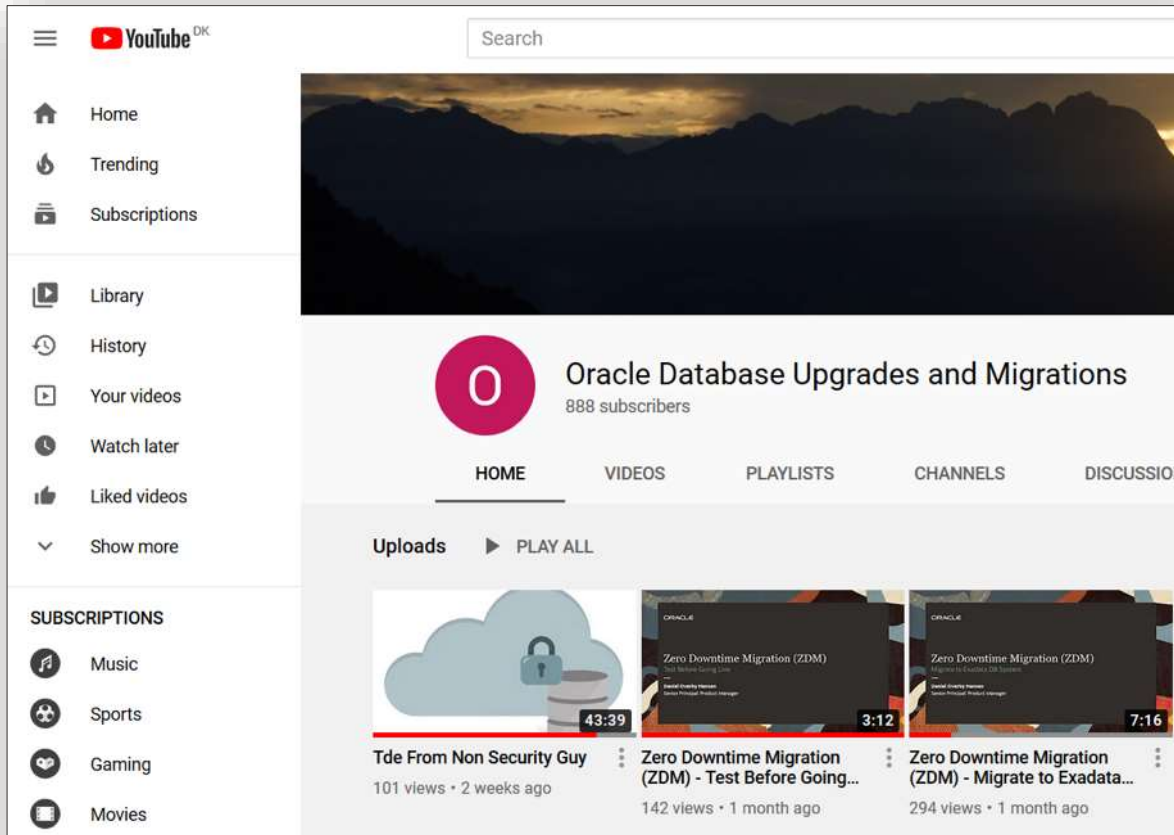
Follow Blog via Email

Enter your email address to follow this blog and receive notifications of new posts by email.

 [Follow](#) [***](#)



YouTube | Oracle Database Upgrades and Migrations



[YouTube Channel](#)

Thank you!

