
# Secure your job

## Fallback is your insurance



Photo by Kamil Pietrzak on Unsplash


**ROY SWONGER**
Vice President
Database Upgrade, Utilities & Patching

royfswonger

@royfswonger






**MIKE DIETRICH**
Distinguished Product Manager
Database Upgrade and Migrations

mikedietrich

@mikedietrichde

https://mikedietrichde.com






**DANIEL OVERBY HANSEN**
Senior Principal Product Manager
Cloud Migrations

dohdatabase

@dohdatabase

https://dohdatabase.com





**RODRIGO JORGE**
Senior Principal Product Manager
Database Patching and Upgrade

rodrigoaraujorge

@rodrigojorgedba

https://dbarj.com.br/en





**BILL BEAUREGARD**
Senior Principal Product Manager
Data Pump and SQL Loader

william-beauregard-3053791

# Fallback | Get The Slides

https://dohdatabase.com/webinars

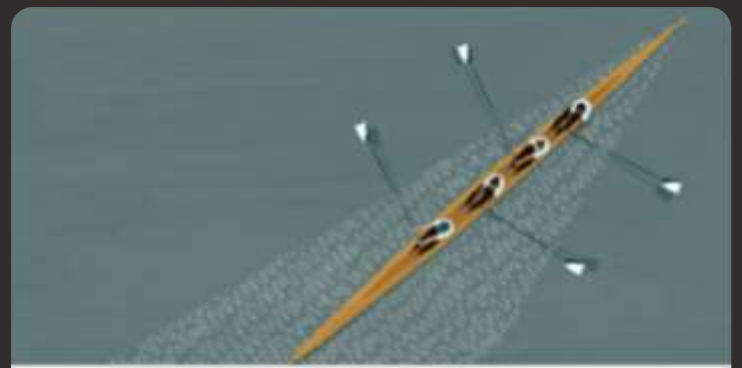
https://MikeDietrichDE.com/slides

# Fallback | Next Webinars

**Migrating Very Large Databases**

December 9, 2021 | 09:00 GMT / 10:00 CET / 11:00 EET / 13:00 GST

Duration: 120 mins



**Data Pump Extreme - Deep Dive with Development**

January 27, 2022 | 09:00 GMT / 10:00 CET / 11:00 EET / 13:00 GST

Duration: 120 mins

**REGISTER**

*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/

1. Intro

Fallback vs Rollback
COMPATIBLE
Time Zone File

2. Methods

3. Scenarios

4. Practice

5. Summary

# TWO CONCEPTS

**ROLLBACK**
Returns the database to the previous, consistent state

**FALLBACK**
Returns the database to a previous release without losing changes

The most limiting factor on your fallback options is `COMPATIBLE`

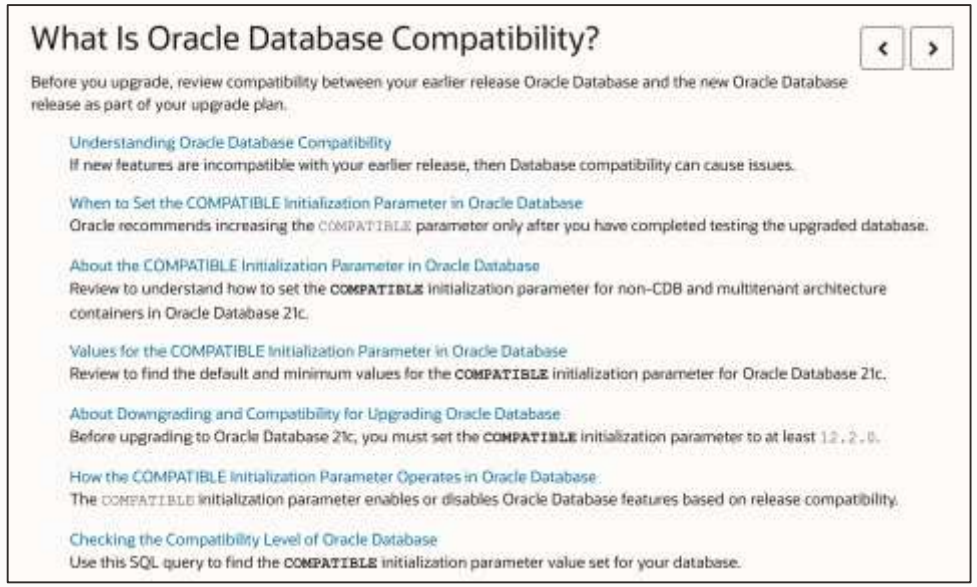
# Compatible | Overview

## What does COMPATIBLE do?

- Enables use of features, e.g.
  - Long identifiers
  - Blockchain tables
- Redo log file structure
- Data file format
- Tablespace header

## Where is it documented?

- Database Upgrade Guide



### What Is Oracle Database Compatibility?

Before you upgrade, review compatibility between your earlier release Oracle Database and the new Oracle Database release as part of your upgrade plan.

Understanding Oracle Database Compatibility
If new features are incompatible with your earlier release, then Database compatibility can cause issues.

When to Set the COMPATIBLE Initialization Parameter in Oracle Database
Oracle recommends increasing the COMPATIBLE parameter only after you have completed testing the upgraded database.

About the COMPATIBLE Initialization Parameter in Oracle Database
Review to understand how to set the COMPATIBLE initialization parameter for non-CDB and multitenant architecture containers in Oracle Database 21c.

Values for the COMPATIBLE Initialization Parameter in Oracle Database
Review to find the default and minimum values for the COMPATIBLE initialization parameter for Oracle Database 21c.

About Downgrading and Compatibility for Upgrading Oracle Database
Before upgrading to Oracle Database 21c, you must set the COMPATIBLE initialization parameter to at least 12.2.0.

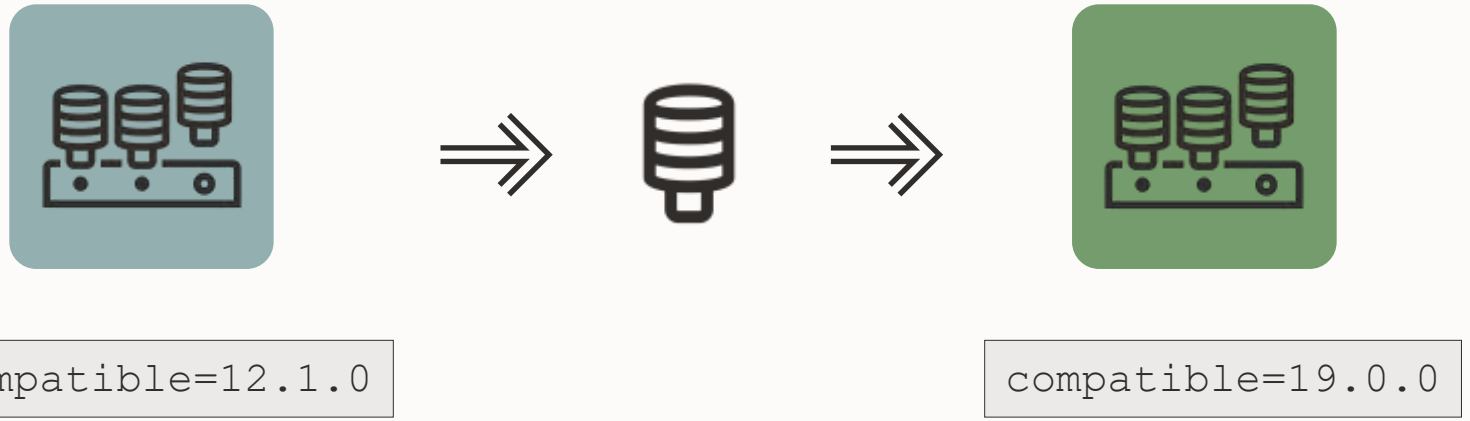
How the COMPATIBLE Initialization Parameter Operates in Oracle Database
The COMPATIBLE initialization parameter enables or disables Oracle Database features based on release compatibility.

Checking the Compatibility Level of Oracle Database
Use this SQL query to find the COMPATIBLE initialization parameter value set for your database.

`COMPATIBLE` can only be changed to a higher value

Changing `COMPATIBLE` prevents the use of Flashback Database and downgrade

# Compatible | Multitenant




On plug-in:

- PDB silently changes it's `COMPATIBLE` setting
- The change is irreversible

We recommend to change `COMPATIBLE` one or two weeks after the upgrade

Pro tip: Changing `COMPATIBLE` requires a database restart

# Compatible | Recommendation

Which value should you use for `COMPATIBLE`?

- The default of the database release
  - 11.2.0
  - 12.1.0
  - 12.2.0
  - 18.0.0
  - 19.0.0

Should you change `COMPATIBLE` when patching?

- NEVER!
  - Except for …




# Compatible | Comparison

Fully independent from each other

`COMPATIBLE`

- Enables features
- Changes on-disk structures

`OPTIMIZER_FEATURES_ENABLE`

- Just reverts to the parameters used in a previous release
- Avoid using it if possible
- This is not a Swiss Army knife!
- You will turn off a lot of great features

> *Modifying the OPTIMIZER_FEATURES_ENABLE parameter generally is strongly discouraged and should only be used as a short term measure at the suggestion of Oracle Global Support.*

Use Caution if Changing the OPTIMIZER_FEATURES_ENABLE Parameter After an Upgrade (Doc ID 1362332.1)

# Compatible | AutoUpgrade

AutoUpgrade does not change `COMPATIBLE`

Unless you want it

```
upg1.drop_grp_after_upgrade=yes
upg1.raise_compatible=yes
```

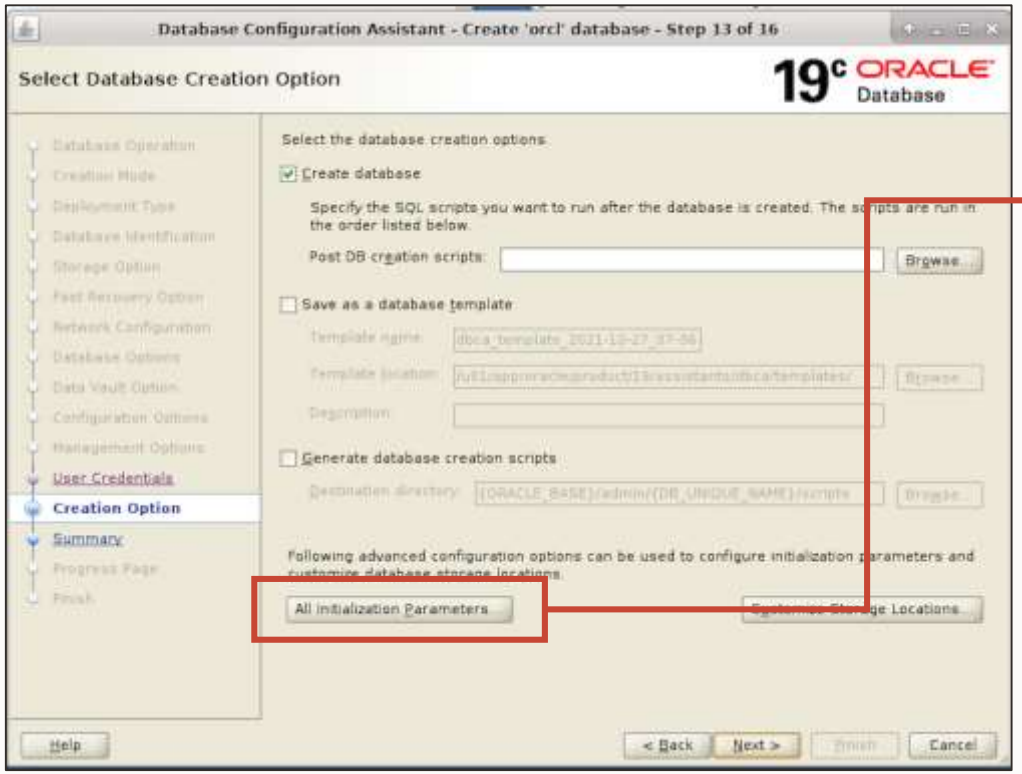


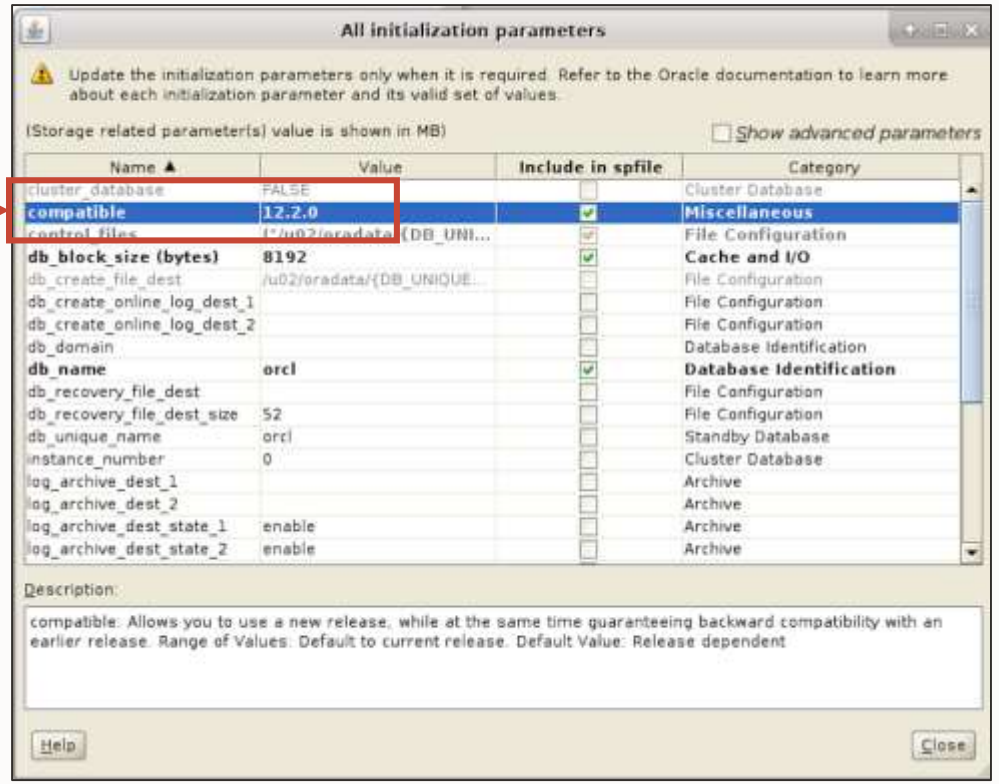

# How do I create a database with a non-default `COMPATIBLE` setting?

# Compatible | DBCA

`COMPATIBLE` choice is only available via CUSTOM database creation

- General Purpose, OLTP and DWH templates create databases with default `COMPATIBLE`

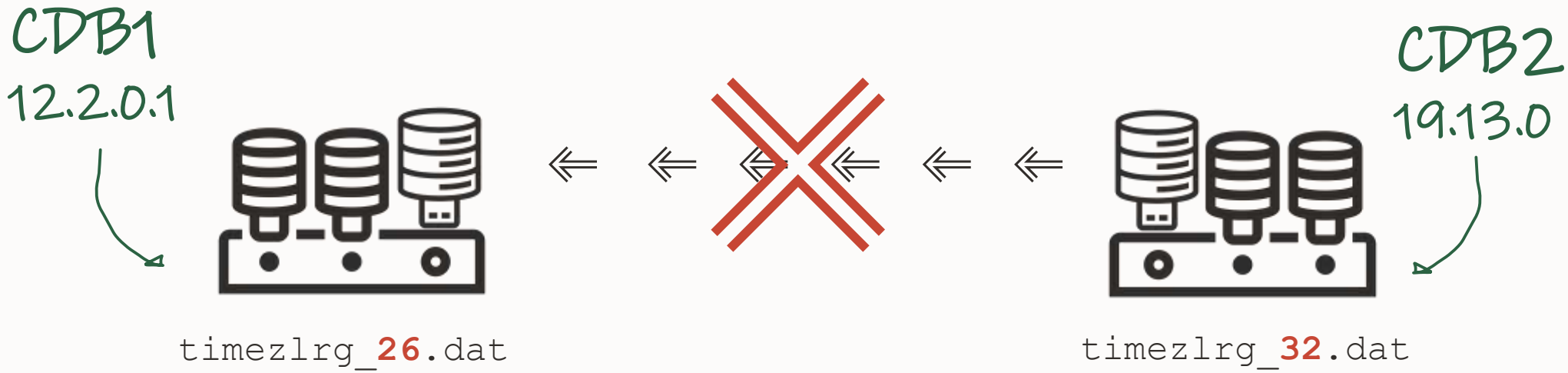


# Time Zone | Downgrade requirements

Identical zone files must exist in both homes and databases

- A lower time zone file's version will prevent downgrade

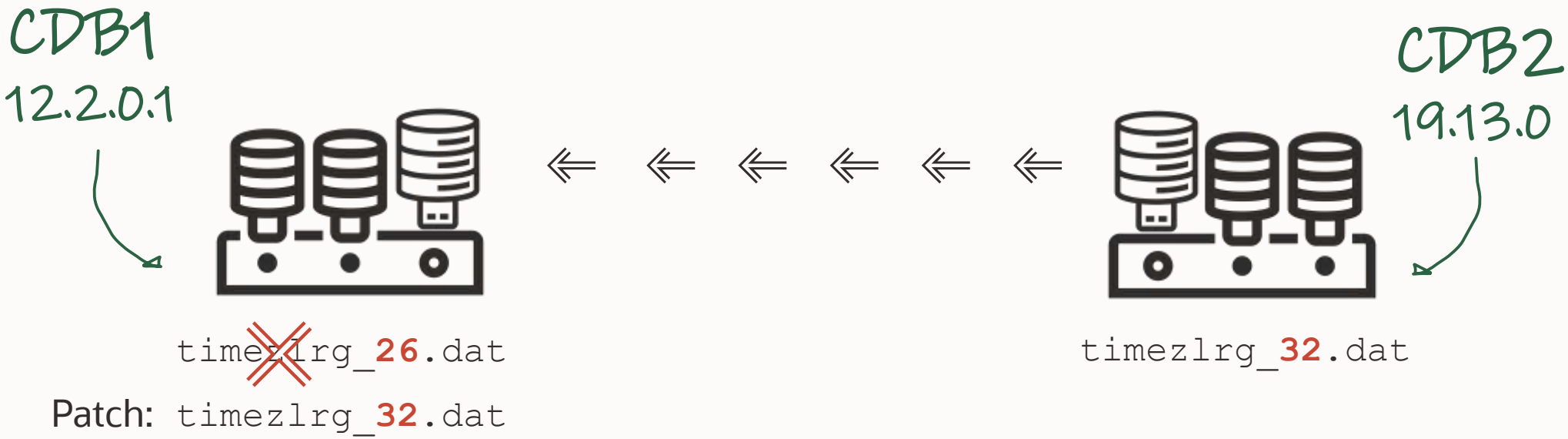


# Time Zone | Downgrade Solution

Apply matching time zone patch to source home and database

- MOS Note:412160.1

| | | | |
|---|---|---|---|
| Version 30 | tzdata2017b update | patch 25881255 * | patch 25881271 |
| Version 31 | tzdata2017c update | patch 27015449 * | patch 27015468 |
| Version 32 | tzdata2018e update | patch 28125601 * | patch 28127287 |
| Version 33 | tzdata2018g update | patch 28852325 * | patch 28852334 |
| Version 34 | tzdata2019b update | patch 29997937 * | patch 29997959 |
| Version 35 | tzdata2020a update | patch 31335037 * | patch 31335142 |

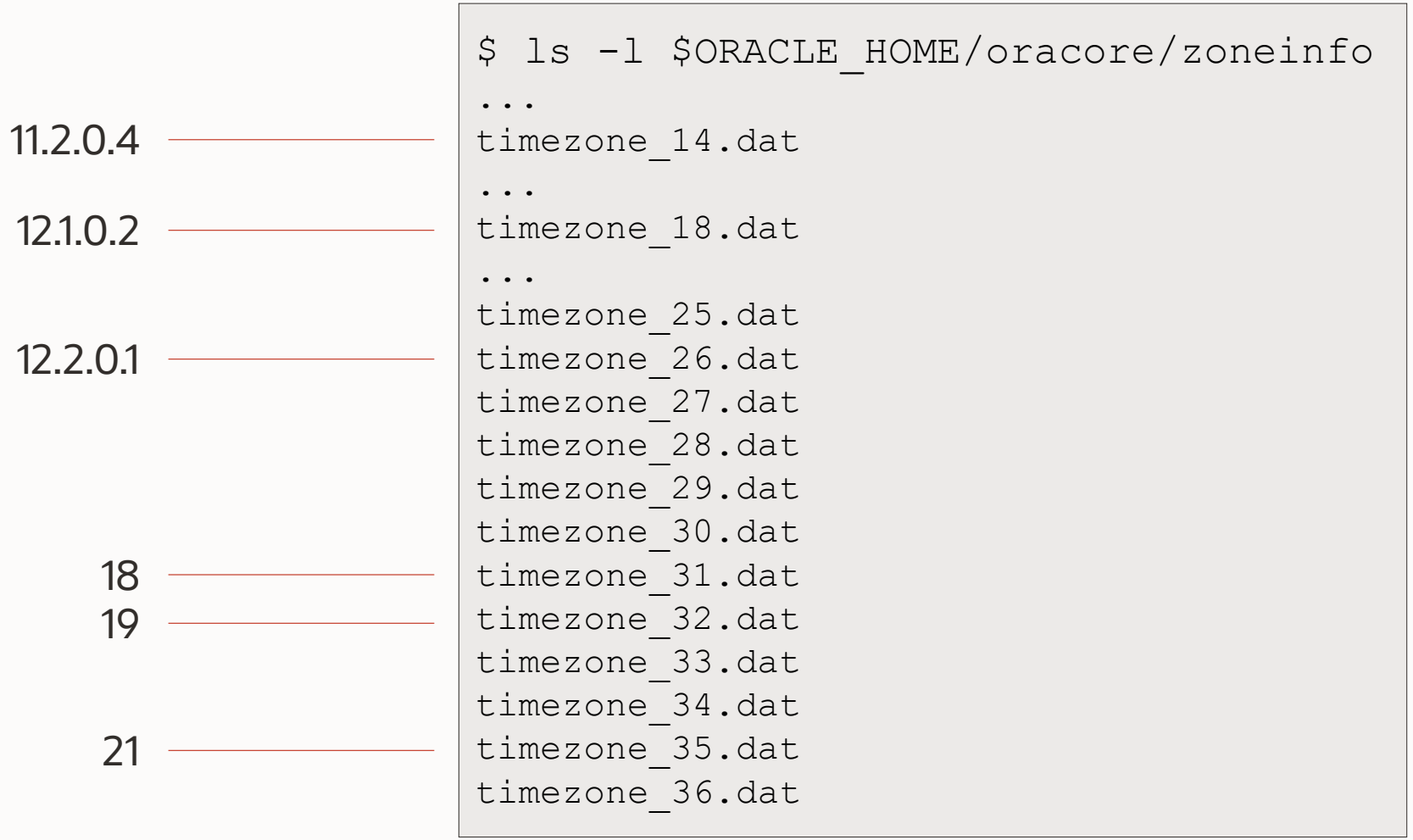


# Time Zone | Check

Check time zone file version upfront

```
SQL> select * from V$TIMEZONE_FILE;


FILENAME             VERSION    CON_ID
-------------------- ---------- ----------
timezlrg_26.dat              26          0
```

# Time Zone | Default Version



# Time Zone | Patch and Apply

Apply time zone patch to Oracle Home

- Files will be written to `$ORACLE_HOME/oracore/zoneinfo`
- Time zone patches are not part of any RU or patch bundle
  - Not RAC-rolling
- Time zone patches are not patch bundle agnostic but generic per release

Use scripts to adjust time zone

- `?/rdbms/admin/utltz_upg_check.sql`
- `?/rdbms/admin/utltz_upg_apply.sql`




# Time Zone | AutoUpgrade

AutoUpgrade upgrades time zone by default

To avoid time zone upgrade

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.pdbs=PDB1
upg1.target_cdb=CDB2
upg1.timezone_upg=no
```




Ease your life by having identical time zone files in all environments

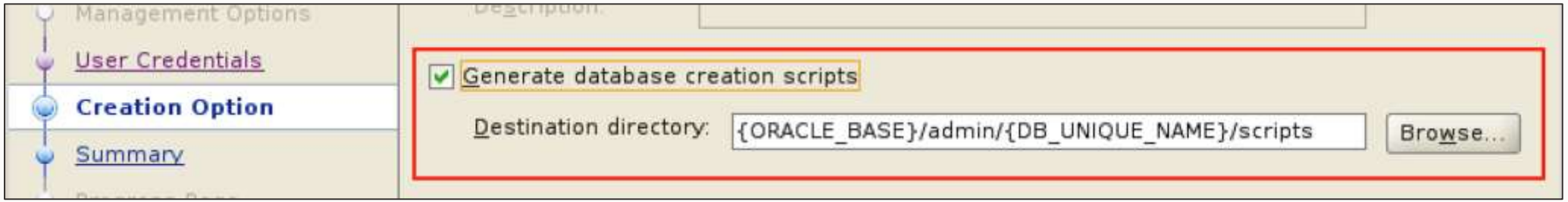
Updating the database time zone file is recommended, not but mandatory

# How to create a database with a non-default Time Zone setting?

# Time Zone | Custom Setting

DBCA does not offer an implicit solution

Workaround:
Create database creation scripts with DBCA



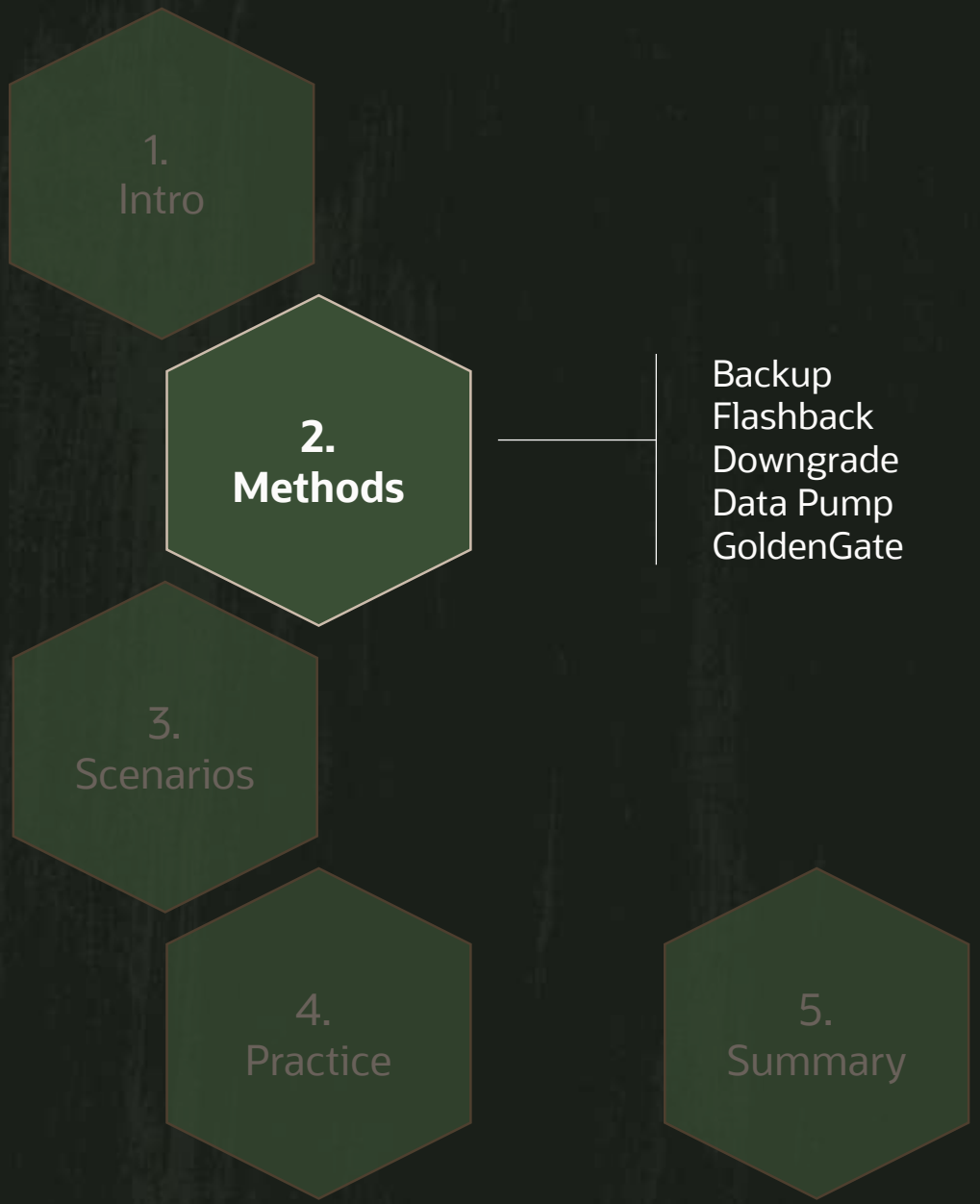

Set environment variable `ORA_TZFILE`

```
$ export ORA_TZFILE=/u01/app/oracle/product/19/oracore/zoneinfo/timezone_14.dat
```

Create database with script

```
$ ./ORCL.sh
```

Within the same run, you can define time zone and `COMPATIBLE` settings



various methods to

# fallback and rollback

# BACKUP

# Backup

- You should always have backups as one of your fallback methods
- But it should never be the primary fallback method
- Because it takes too long to restore a backup




You should perform a backup before and after an event

If time allows, you should use level 0 backups,
If not, level 1 offers the same level of protection

Pro tip: Cumulative incremental might offer a good balance between time to backup and time to recover

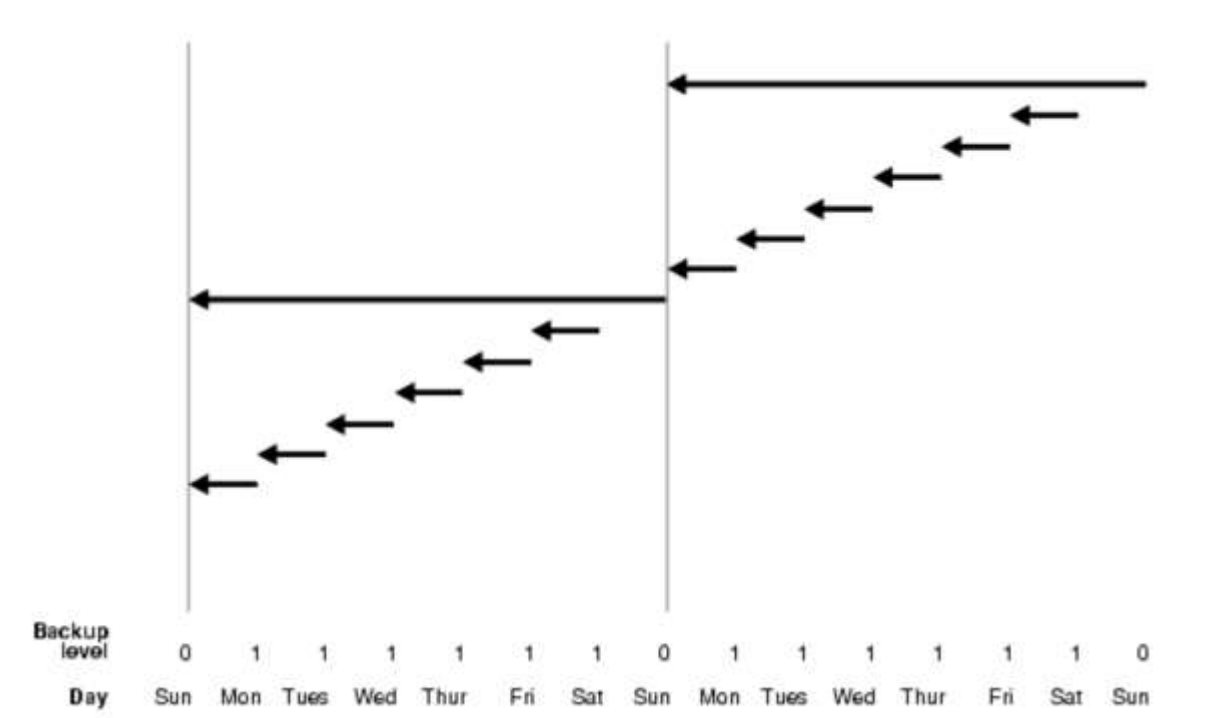
# Block Change Tracking can significantly speed up incremental backups

Pro tip: BCT is an Enterprise Edition feature, but requires Active Data Guard if enabled on standby database

# Backup | Level 0 / 1 Incremental

Differential:

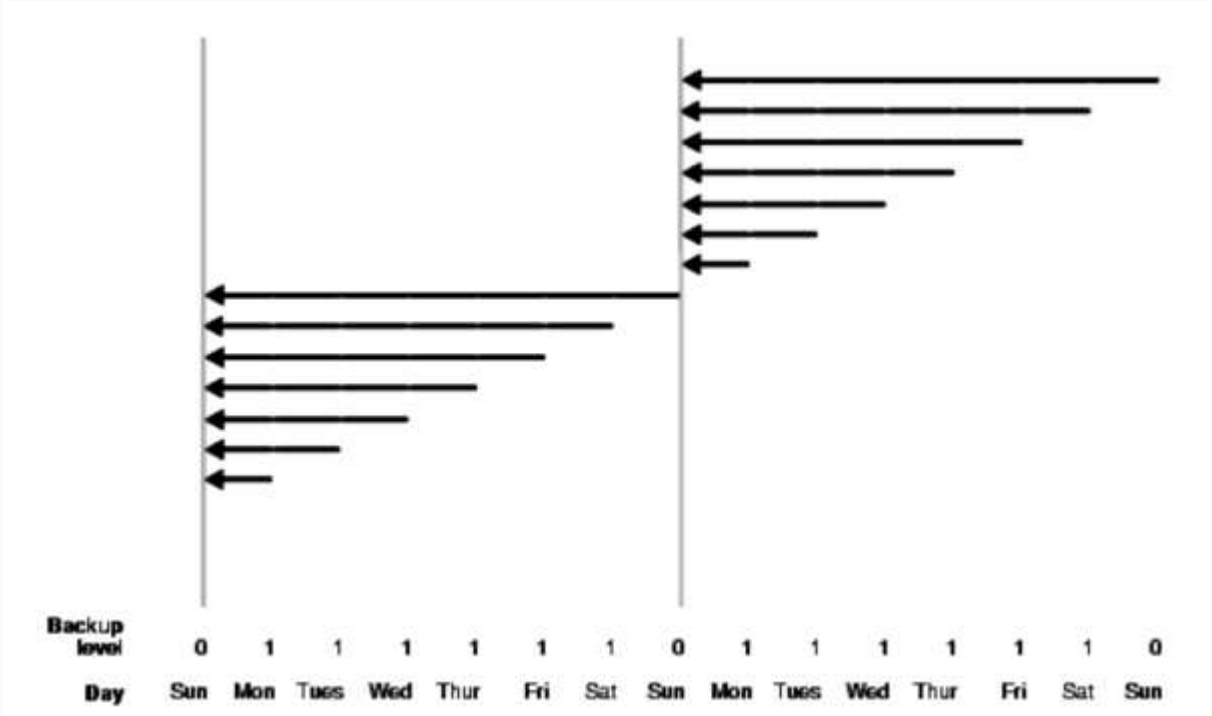
```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;
```



# Backup | Level 0 / 1 Incremental

Cumulative:

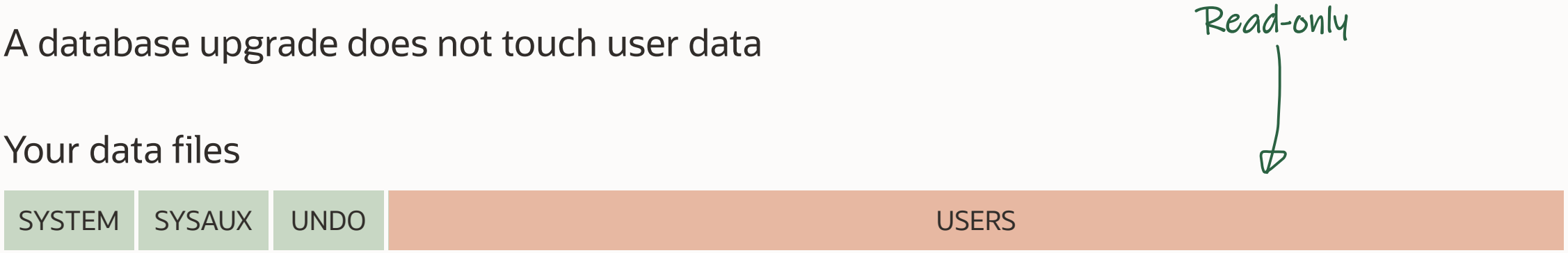
```
RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
```



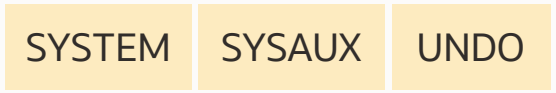
# Backup | Partial Offline

A database upgrade does not touch user data

Your data files




Partial offline backup (plus redo log and control files)




Start upgrade

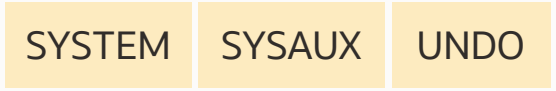
Pro tip: Works for SE2 and databases in NOARCHIVELOG mode

# Backup | Partial Offline

To restore

Your data files


Your backup (plus redo log and control files)




Pro tip: Partial offline backup is not applicable for unplug-plug upgrade

various methods to

# fallback and rollback

# Flashback | Overview

- Reverts the database back in time,
  all changes to the database is undone
- Often preferred because it is easy,
  and very fast
- Requires:
  - Enterprise Edition
  - ARCHIVELOG mode
  - 10-20 GB for Flashback Logs
  - `COMPATIBLE` must not be changed

# Flashback | Concept

## PRE-UPGRADE ENVIRONMENT

```
SQL> create restore point BEFORE_UPG
     guarantee flashback database;
```

## POST-UPGRADE ENVIRONMENT

**FLASHBACK**

```
SQL> shutdown immediate
SQL> startup mount
SQL> flashback database
     to restore point BEFORE_UPG;
SQL> shutdown immediate
```

```
SQL> startup mount
SQL> alter database open resetlogs;
```

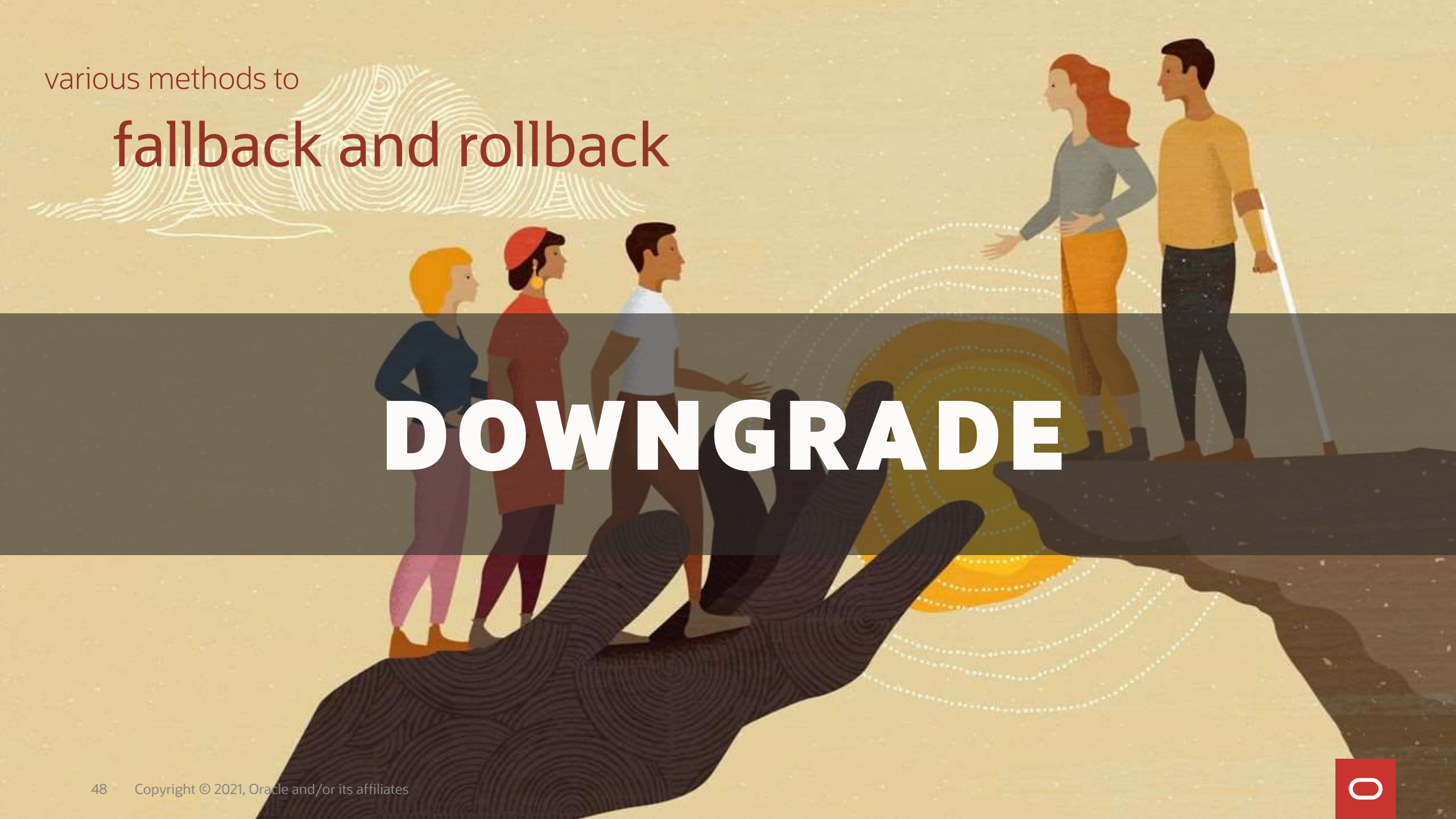



# Always use guaranteed restore points, and remember to drop them again

Pro tip: Forgetting to drop a GRP will eventually bring the database to a complete halt

various methods to

# fallback and rollback



# Downgrade | Overview

- Brings the database back to a previous release
- Works days, weeks or months after upgrade
- No data loss
- Requires:
  - `COMPATIBLE` must not be changed
  - Time zone file version must match




## Downgrade | Data Dictionary

A downgraded database is not identical to the pre-upgraded database

**The data dictionary will be different - but compatible**

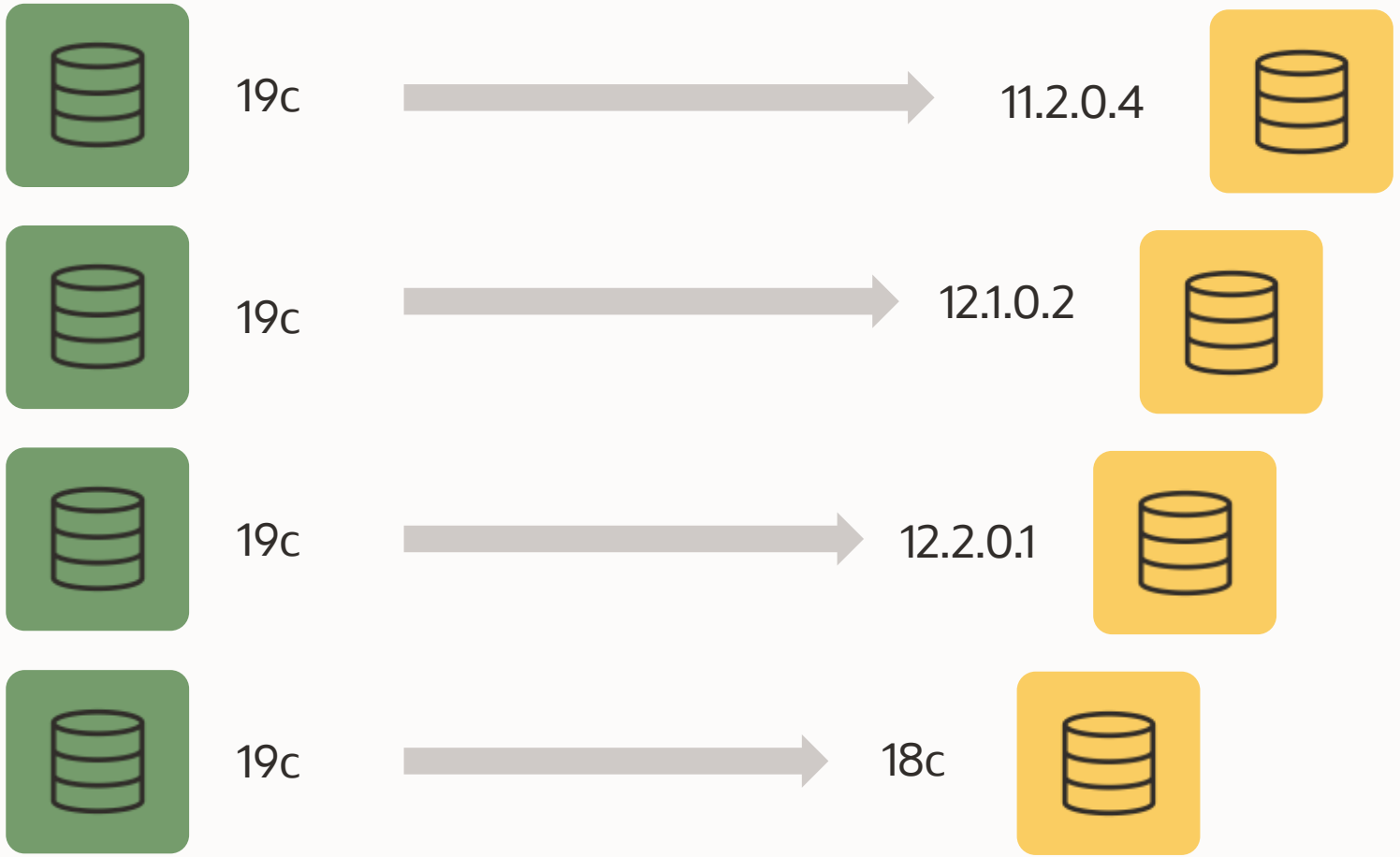
Examples:

- New dictionary tables are not dropped, but truncated
- New indexes are not dropped
- Generally, dropping is avoided

Downgrade reverts only the data dictionary to a state compatible with a previous release

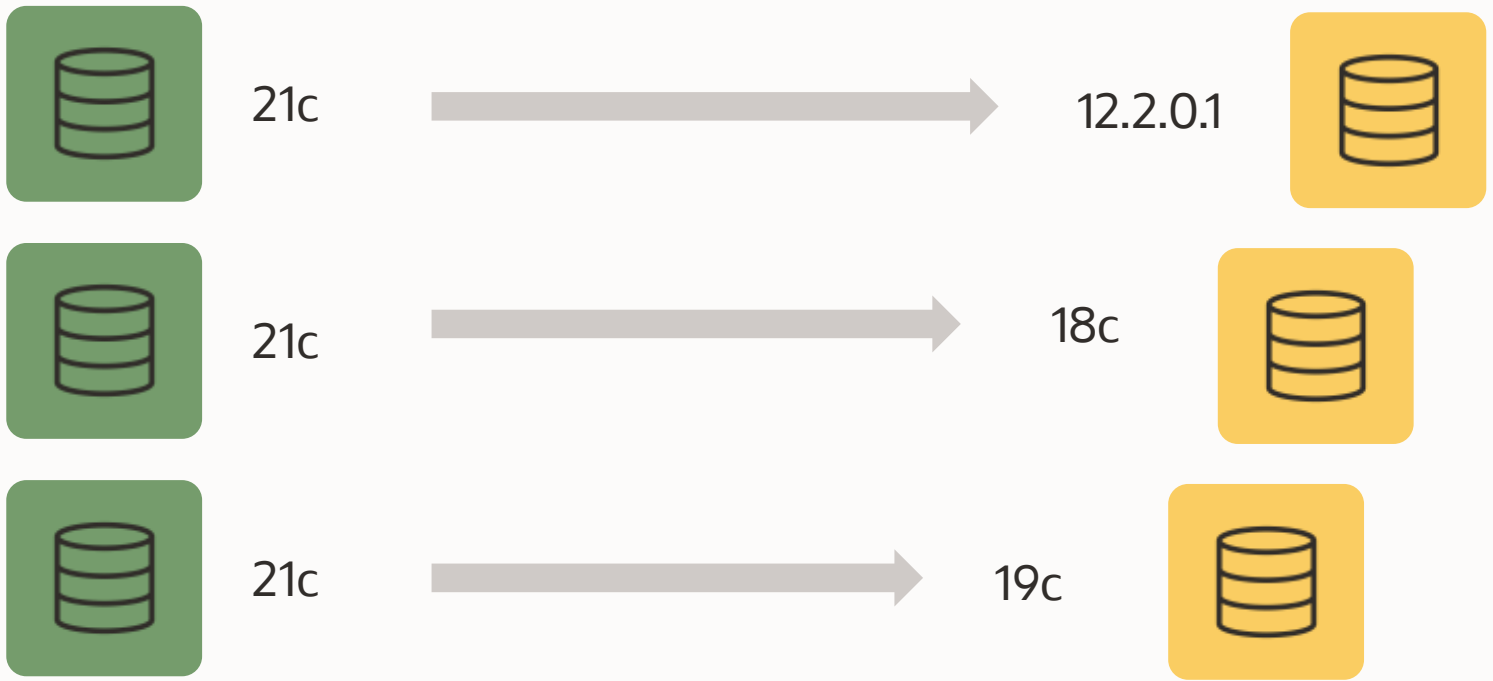
Pro tip: Keep your old Oracle Home available after the upgrade, *just in case*

# Downgrade | Oracle Database 19c




For CDB architecture, you can only downgrade to these release
- not 11.2.0.4

# Downgrade | Oracle Database 21c




CDB-only architecture

# Downgrade | Fallback Concept

## PRE-UPGRADE ENVIRONMENT

## POST-UPGRADE ENVIRONMENT

```
SQL> startup downgrade
```

**DOWNGRADE**

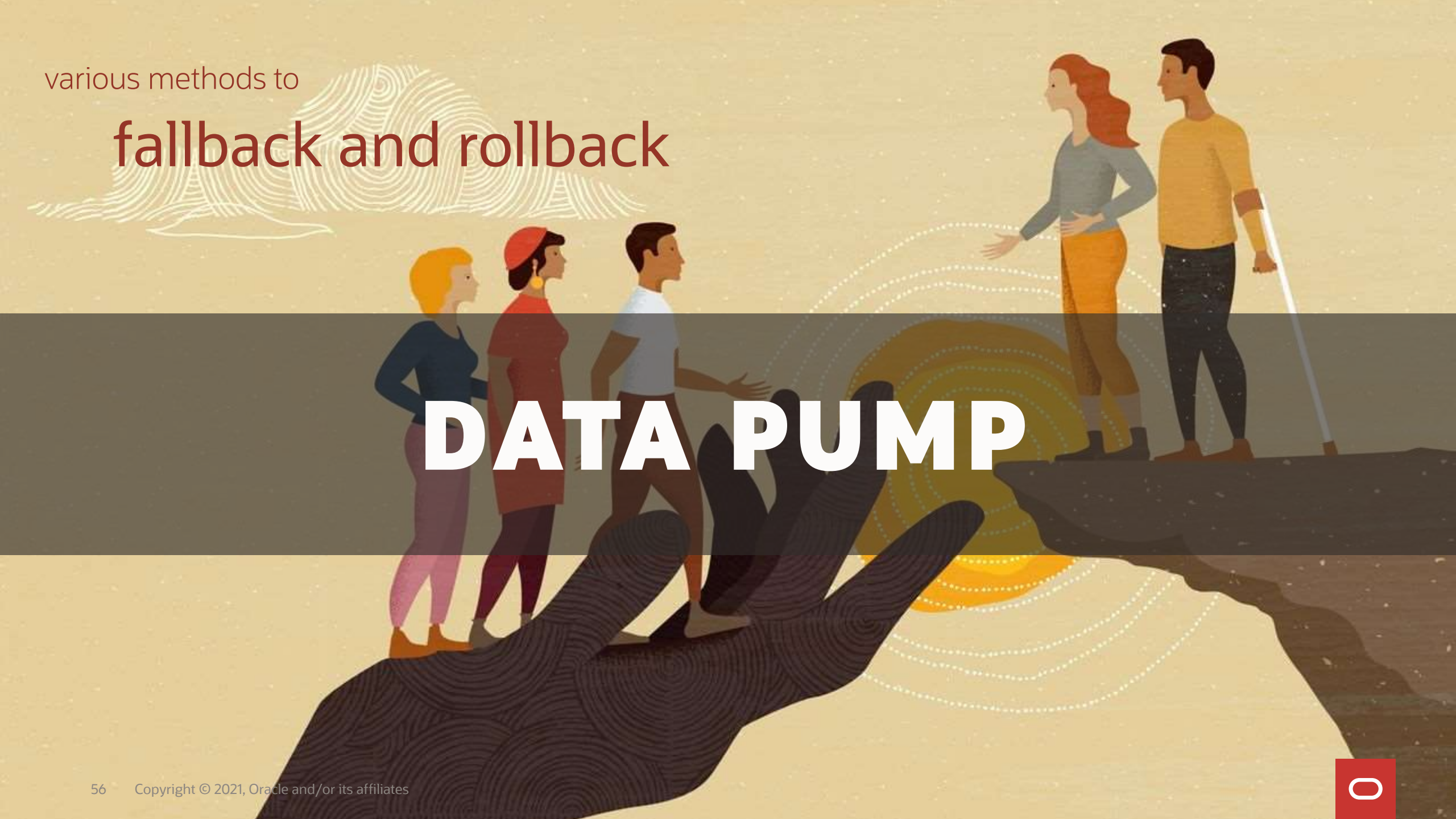
```
SQL> @?/rdbms/admin/catdwgrd.sql
SQL> shutdown immediate
```

```
SQL> startup upgrade
SQL> @?/rdbms/admin/catrelod.sql
SQL> @?/rdbms/admin/utlrp.sql
```

Before downgrading, there is no need to rollback any patches, neither RUs or RURs nor one-off patches

various methods to

# fallback and rollback



## DATA PUMP

# Data Pump | Overview

- Universal fallback solution
- When all other fails,
  Data Pump works
- Often not used because considerable downtime is needed
- Requires:
  - Time zone file version must match




When downgrading, tables using new features are not imported

# Data Pump | Overview

## Create a dump file compatible with a lower release

```
version=11.2.0.4
```

## Other options are

- `COMPATIBLE` (default)
- `LATEST`

Pro tip: See MOS Doc ID 553337.1 for further information




# Data Pump over database link does not work for downgrades

Pro tip: Data Pump over database link is using the `NETWORK_LINK` parameter




# Data Pump | Fallback Concept

## PRE-UPGRADE ENVIRONMENT

## POST-UPGRADE ENVIRONMENT

**DATA PUMP**

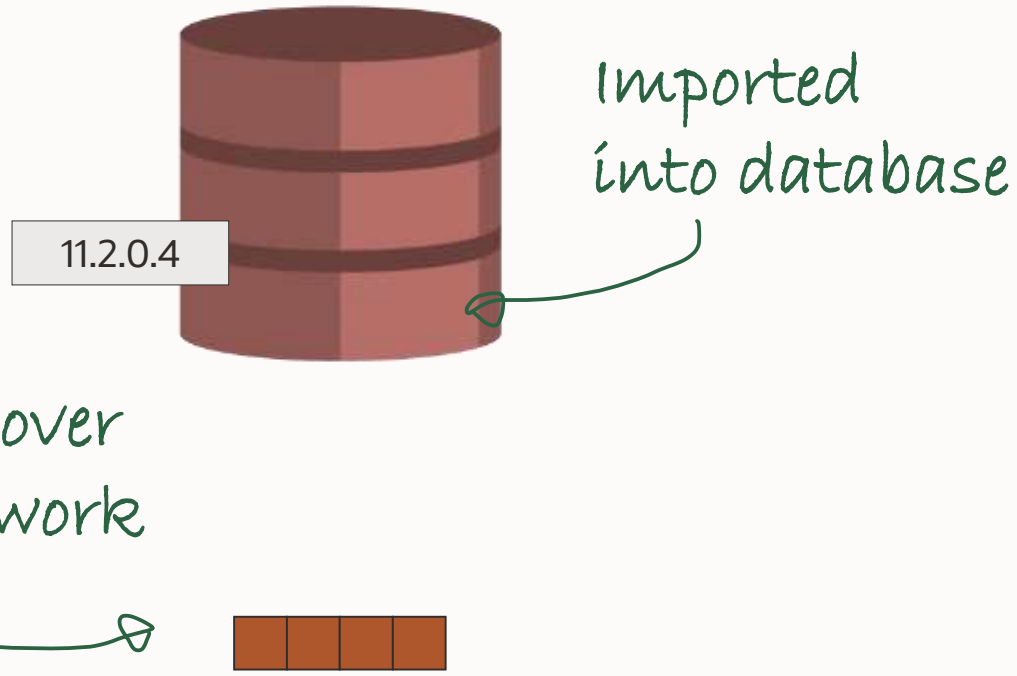
```
expdp ... VERSION=<source release>
```

```
impdp ...
```

Pro tip: Keep your old Oracle Home available and create an empty database




# Data Pump | Fallback Explained



various methods to

# fallback and rollback

# GoldenGate | Overview

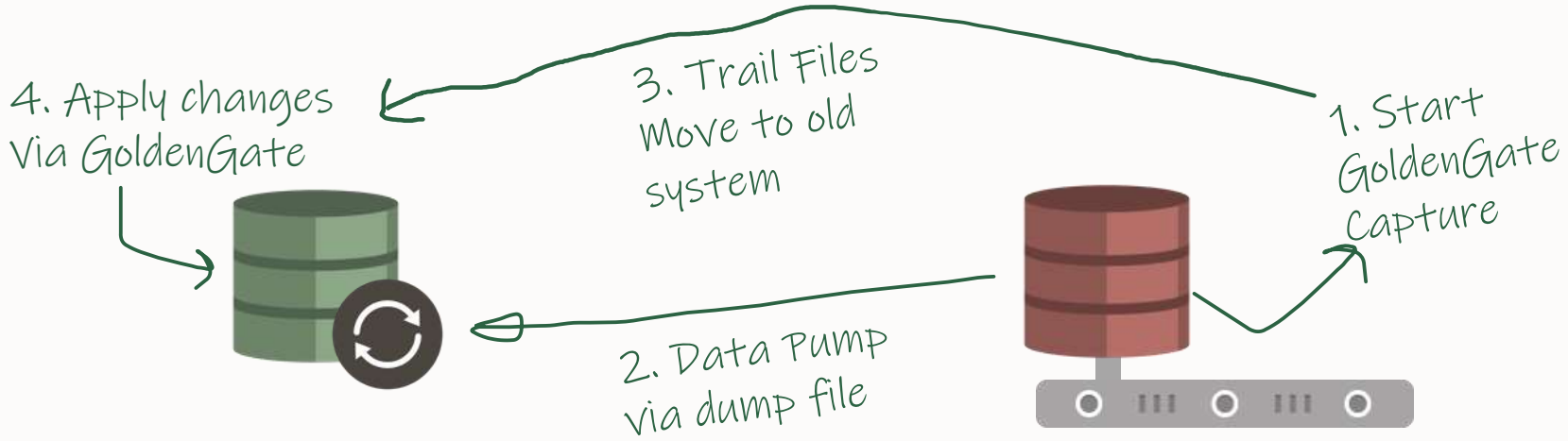
- Zero downtime fallback option
- Universal and flexible,
  but complex
- Use in combination with Data Pump




# GoldenGate | Fallback

Original database: 11.2.0.4 non-CDB
- Upgrade
- Plug-in
- Convert

Upgraded database: 19c PDB
- Capture
- Export
- Apply

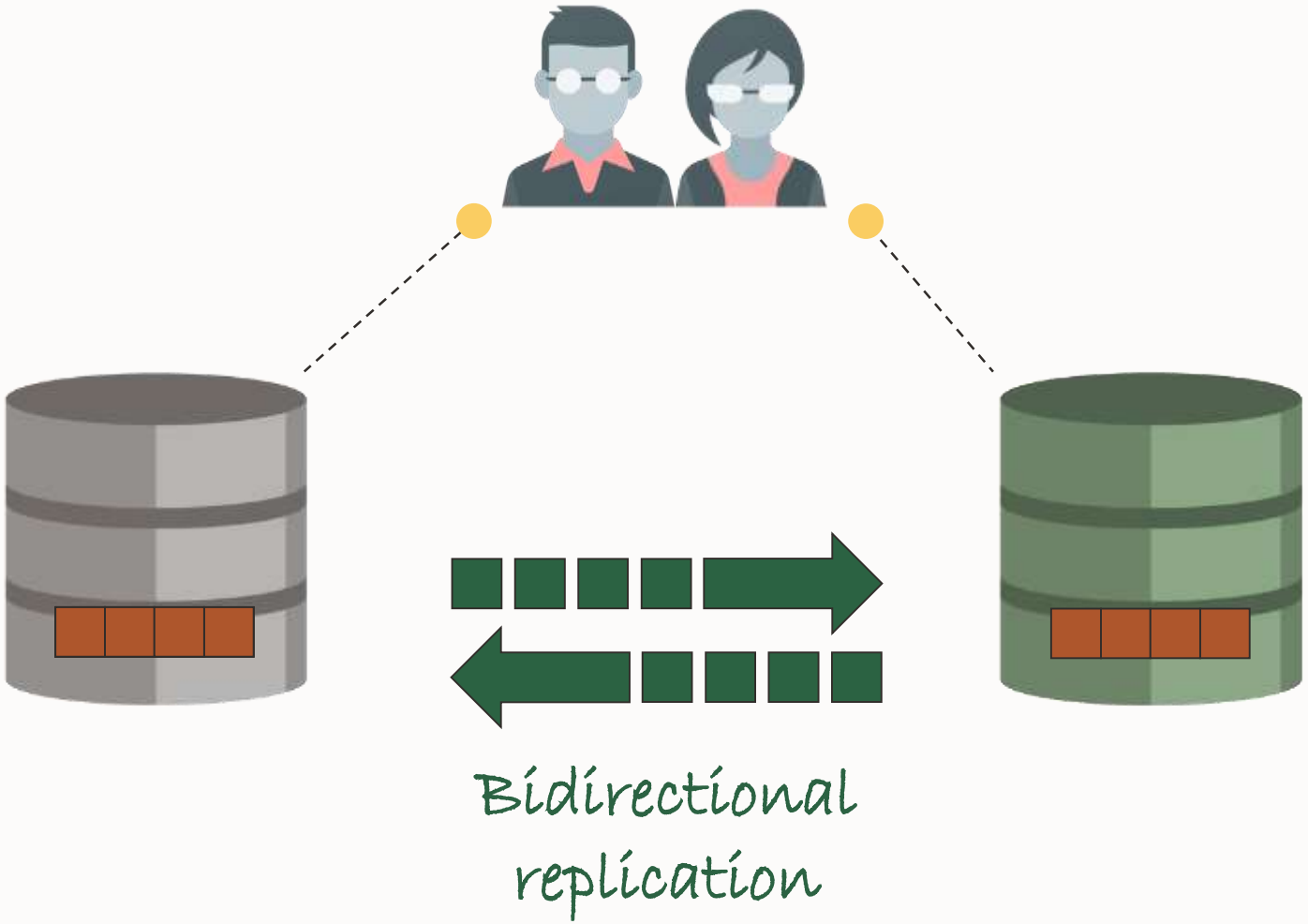



5. Switch clients back when synchronization has caught up

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

Pro tip: If you keep your old database in place, you can skip the data pump step

# GoldenGate | Phased Migration



various methods to

# fallback and rollback

# SUMMARY

# comparing FALLBACK methods

| | Backup | Flashback | Downgrade | Data Pump | GoldenGate |
|---|---|---|---|---|---|
| Data Loss | x | x | | | |
| Use after go-live | | | x | x | x |
| Downtime | Considerable | Almost none | Some | Considerable | None |
| Phased migration | | | | | x |
| Revert PDB conversion | | | | x | x |

Whichever method you choose,
be sure to practice, practice and practice

# Fallback | Grid Infrastructure Downgrade

## Options for Oracle Grid Infrastructure Downgrades

You can downgrade Oracle Grid Infrastructure 19c to earlier releases.

Downgrade options include the following earlier releases:

- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 18c.
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 12c Release 2 (12.2).
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 12c Release 1 (12.1).
- Oracle Grid Infrastructure downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2). Because all cluster configurations in Oracle Grid Infrastructure 19c are Oracle Flex Clusters, when you downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2), you downgrade from an Oracle Flex cluster configuration to a Standard cluster configuration.

**Note:** When you downgrade Oracle Grid Infrastructure to an earlier release, for example from Oracle Grid Infrastructure 19c to Oracle Grid Infrastructure 18c, the later release RAC databases already registered with Oracle Grid Infrastructure will not start after the downgrade.

**Related Topics**

- My Oracle Support Note 2180188.1

**Parent topic:** Downgrading Oracle Clusterware to an Earlier Release

Documentation

1. Intro

2. Methods

**3. Scenarios**

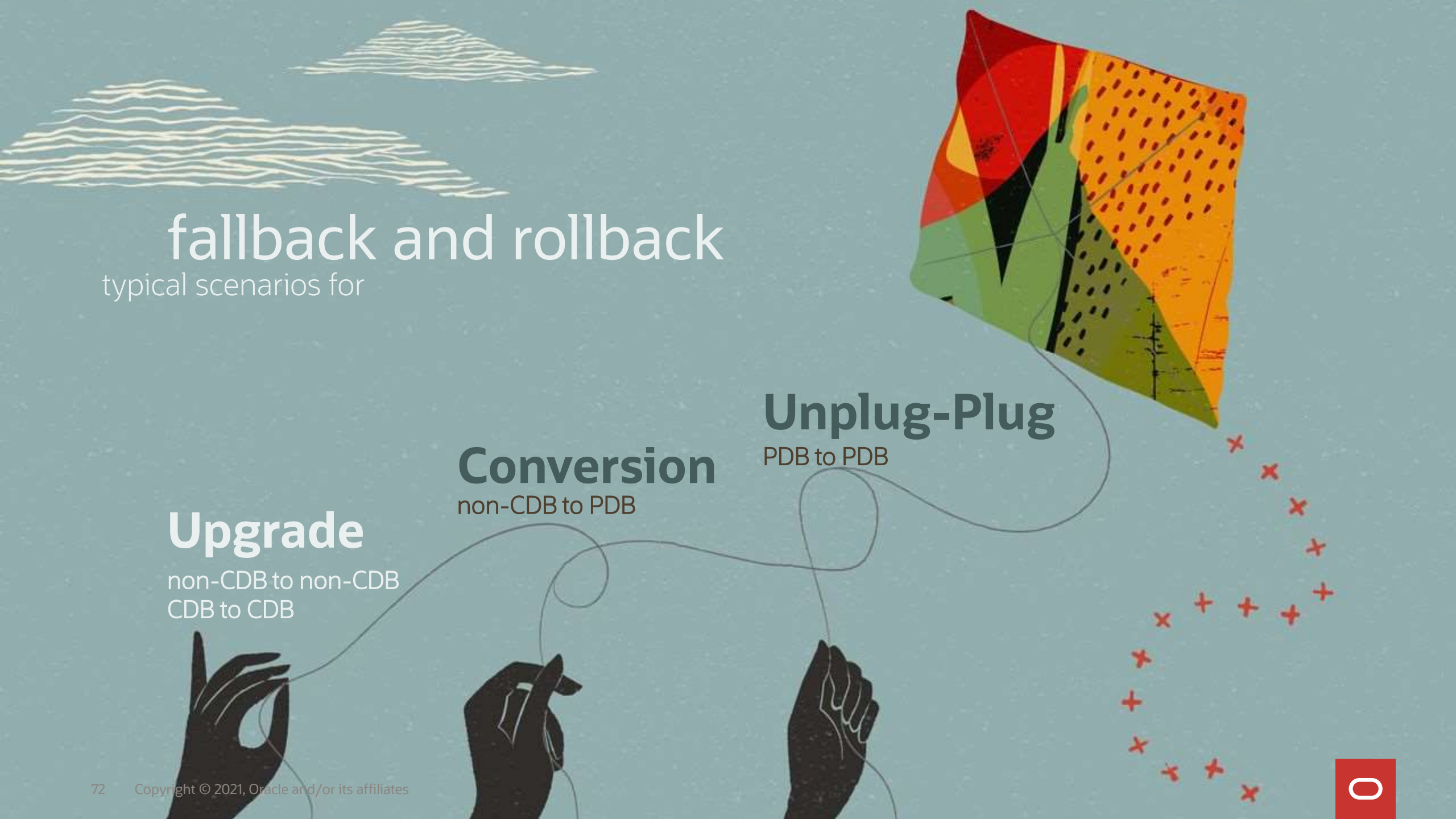
- Upgrade
- Upgrade with PDB Conversion
- Unplug / Plug / Upgrade

4. Practice

5. Summary

# fallback and rollback

typical scenarios for



## Upgrade

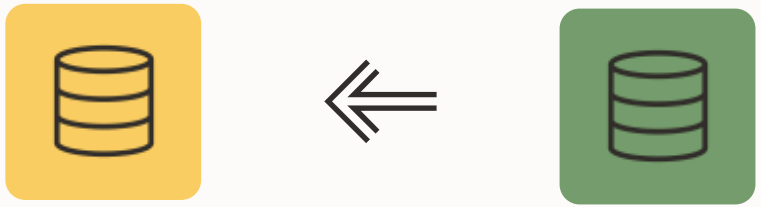
non-CDB to non-CDB
CDB to CDB

## Conversion

non-CDB to PDB

## Unplug-Plug

PDB to PDB



# FLASHBACK

- Preferred
- Data Loss

# Flashback | AutoUpgrade

## Guaranteed Restore Points

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.restoration=yes
upg1.drop_grp_after_upgrade=no
```

- Default behavior:
  - AutoUpgrade creates GRP except for
    - Standard Edition 2
    - `restoration=no`
  - GRP will be kept
  - GRP needs to be removed manually except for
    - `drop_grp_after_upgrade=yes` will only remove it when upgrade completed successfully

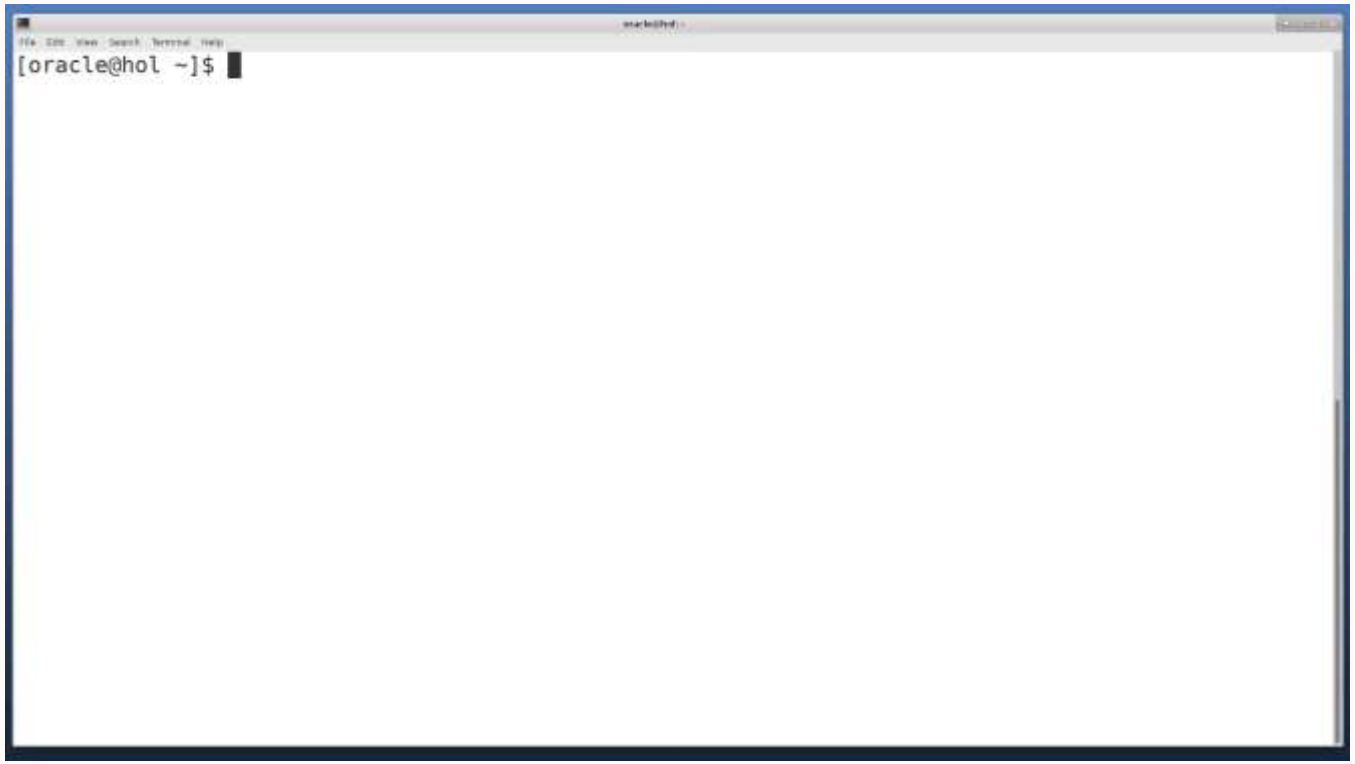
# Flashback | AutoUpgrade

AutoUpgrades handles everything, including

- `/etc/oratab`
- Clusterware registration
- Moving files
  - PFile
  - SPFile
  - Password file
  - Etc.




# Flashback | AutoUpgrade




Watch on YouTube

# What about Data Guard?

You can use Flashback Database
without compromising your standby databases

# Flashback | Data Guard

- Restore Points are needed on primary and standby databases
- Important: First standby database, then primary
- Always use guaranteed restore points - also, on standby databases
- Don't rely on restore point propagation

# Flashback | Data Guard

## PRIMARY

```
SQL> create restore point ...
     guarantee flashback database;
```

## STANDBY

```
SQL> create restore point ...
     Guarantee flashback database;
```

## FLASHBACK

```
SQL> shutdown immediate
SQL> startup mount
SQL> flashback database ...;
SQL> alter database open resetlogs;
```

```
SQL> shutdown immediate
```

```
SQL> startup mount
SQL> flashback database ...
SQL> alter database recover managed
     standby database ...;
```

# Data Guard broker does not support flashing back to a previous release

Pro tip: Check the Data Guard Broker documentation for details




Data Guard broker must be shut down during a flashback to a previous release

# Flashback | Data Guard

You should either:

- Backup the broker configuration files before the upgrade

```
SQL> select value from v$parameter where name like 'dg\_broker\_config\_%' escape '\';
```

- Recreate the Data Guard broker configuration

```
DGMGRL> create configuration ...
DGMGRL> add database ...
DGMGRL> enable configuration
```

In Oracle Database 19c you can `EXPORT CONFIGURATION` and `IMPORT CONFIGURATION` in Data Guard CLI (DGMGRL)




# What about RAC?

You can use Flashback Database on a RAC database

Only one instance should be running during a flashback operation

# Flashback | RAC

## Stop database (all instances) and mount one instance

```
$ srvctl stop database -d $ORACLE_UNQNAME
$ srvctl start instance -d $ORACLE_UNQNAME -i $ORACLE_SID -o MOUNT
```

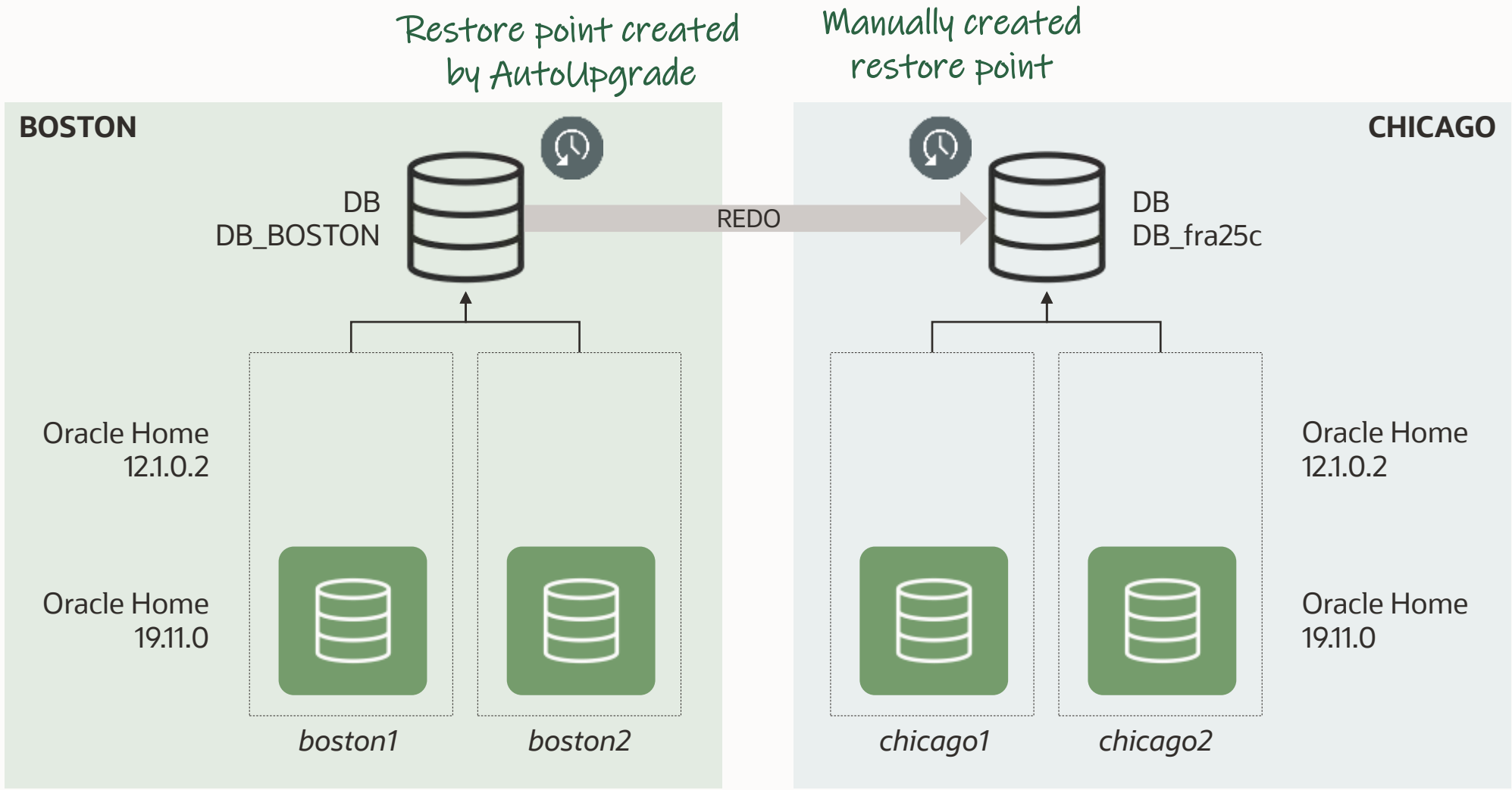
## Flash back the database

```
SQL> flashback database ...              --Higher release Oracle Home
SQL> alter database open resetlogs;      --Lower release Oracle Home
```

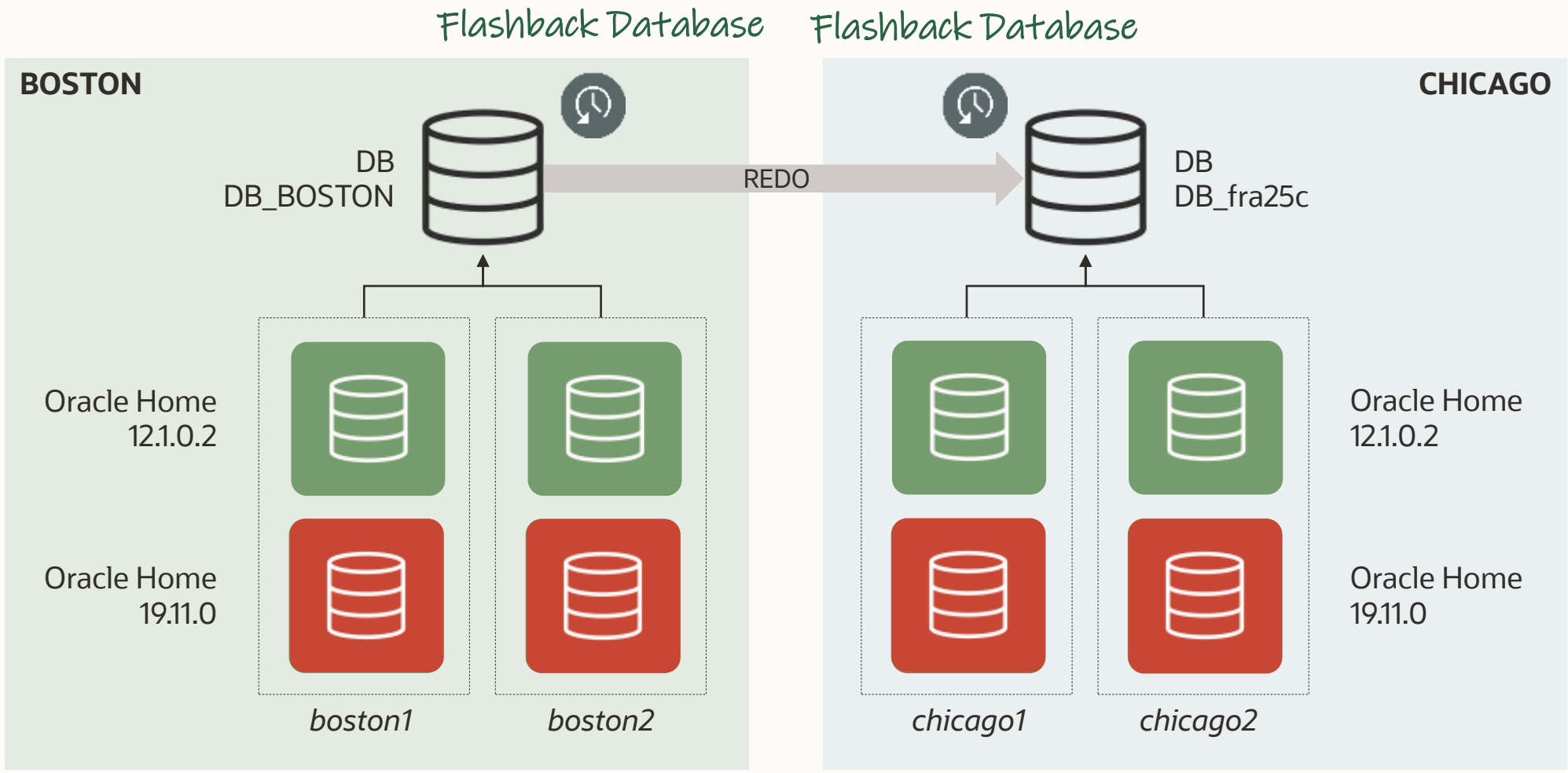
## Start database (all instances)

```
$ srvctl stop instance -d $ORACLE_UNQNAME -i $ORACLE_SID
$ srvctl downgrade database -d $ORACLE_UNQNAME -o <lower_release_home> -t 12.2.0.1
$ srvctl start database -d $ORACLE_UNQNAME
```

# Flashback | Demo

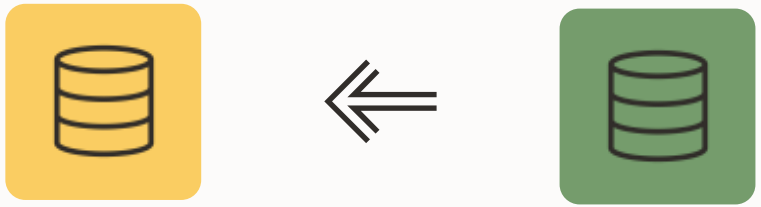


# Flashback | Demo



# Flashback | Demo


Watch on YouTube



# DOWNGRADE

- No Data Loss

# Downgrade | Releases

You can downgrade from Oracle Database 19 to:

- 18
- 12.2
- 12.1.0.2
- 11.2.0.4 (non-CDB only)

Pro tip: Check the Upgrade Guide for details




Perform level 0 backup before and after downgrading, or if time does not allow, at least a level 1 backup

It is recommended to install the latest Release Update in higher and lower release Oracle Homes before you start the downgrade

Check MOS notes 2539751.1 and 2548962.1
for important patches to apply before downgrading

# Downgrade | Statistics

What about statistics?

- Dictionary statistics
  - Gather immediately after downgrade
- Fixed objects statistics
  - Gather when database is warmed-up
- Optimizer statistics
  - Regather stale statistics




# What about Data Guard?

You can downgrade a database
without compromising your standby database

# Downgrade | Data Guard

## PRIMARY

```
SQL> startup downgrade

$ ./dbdowngrade
```

Restart database in lower release Oracle Home

```
SQL> @catrelod
SQL> @utlrp

$ datapatch -verbose
```

## STANDBY

Wait for all redo to be applied

Restart database in lower release Oracle Home

Wait for all redo to be applied




# Data Guard broker does not support downgrading

Pro tip: Check the Data Guard Broker documentation for details

Data Guard broker must be shut down during a downgrade

# Downgrade | Data Guard

You should either:

- Backup the broker configuration files before the upgrade

```
SQL> select value from v$parameter where name like 'dg\_broker\_config\_%';
```

- Recreate the Data Guard broker configuration

```
DGMGRL> create configuration ...
DGMGRL> add database ...
DGMGRL> enable configuration
```

In Oracle Database 19c you can `EXPORT CONFIGURATION` and `IMPORT CONFIGURATION` in Data Guard CLI (DGMGRL)




# What about RAC?

It is possible to downgrade a RAC database

During a downgrade, the parameter `CLUSTER_DATABASE` must be set to `FALSE`

# Downgrade | RAC

## Stop database (all instances) and start one instance in higher release Oracle Home

```
SQL> alter system set cluster_database=false scope=spfile sid='*';

$ srvctl stop database -d $ORACLE_UNQNAME

SQL> startup downgrade
```

## Downgrade

```
$ ./dbdowngrade
```




# Downgrade | RAC

## Restart one instance in lower release Oracle Home

```
SQL> startup upgrade
```

## Reload, recompile and datapatch

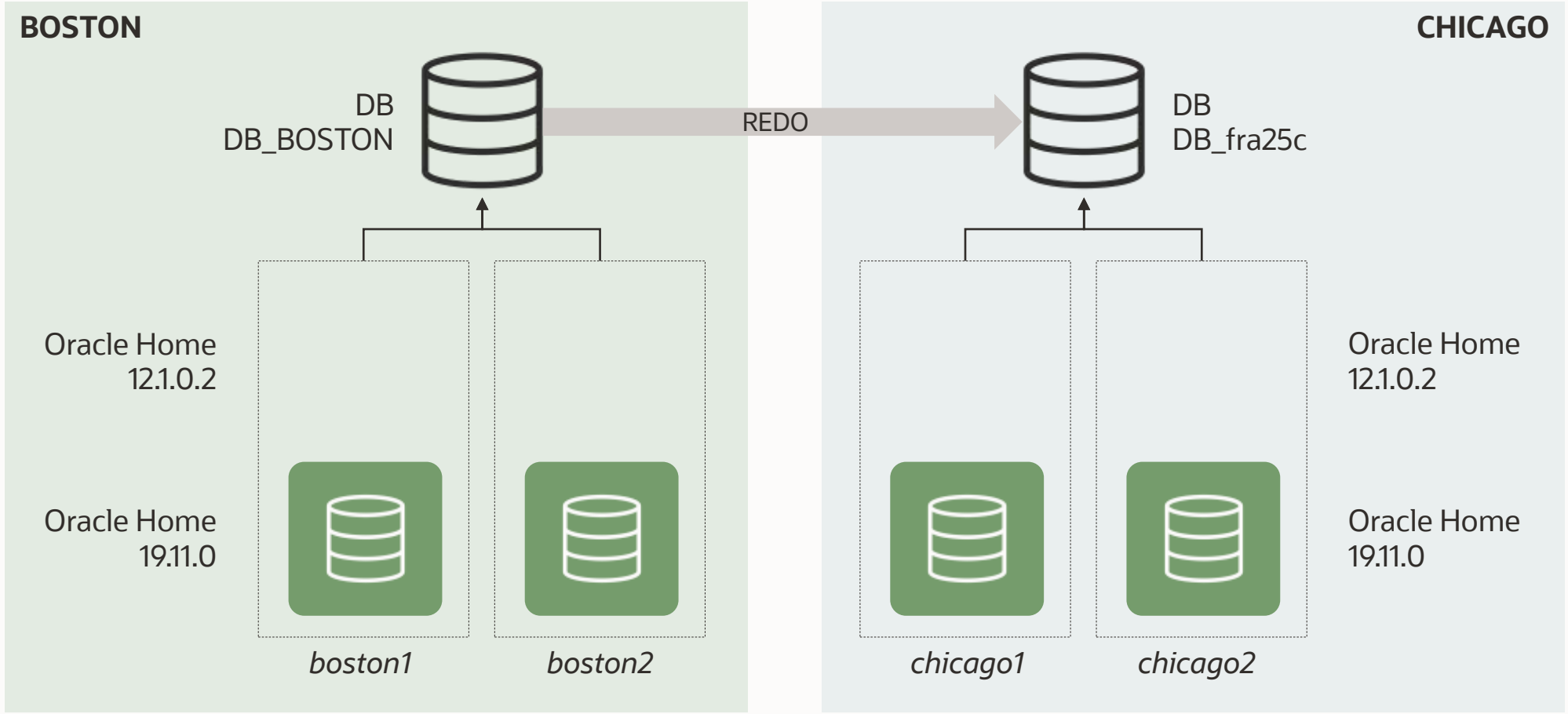
```
SQL> @catrelod
SQL> @utlrp

$ datapatch -verbose
```

## Stop instance, clusterware downgrade and restart database (all instances)

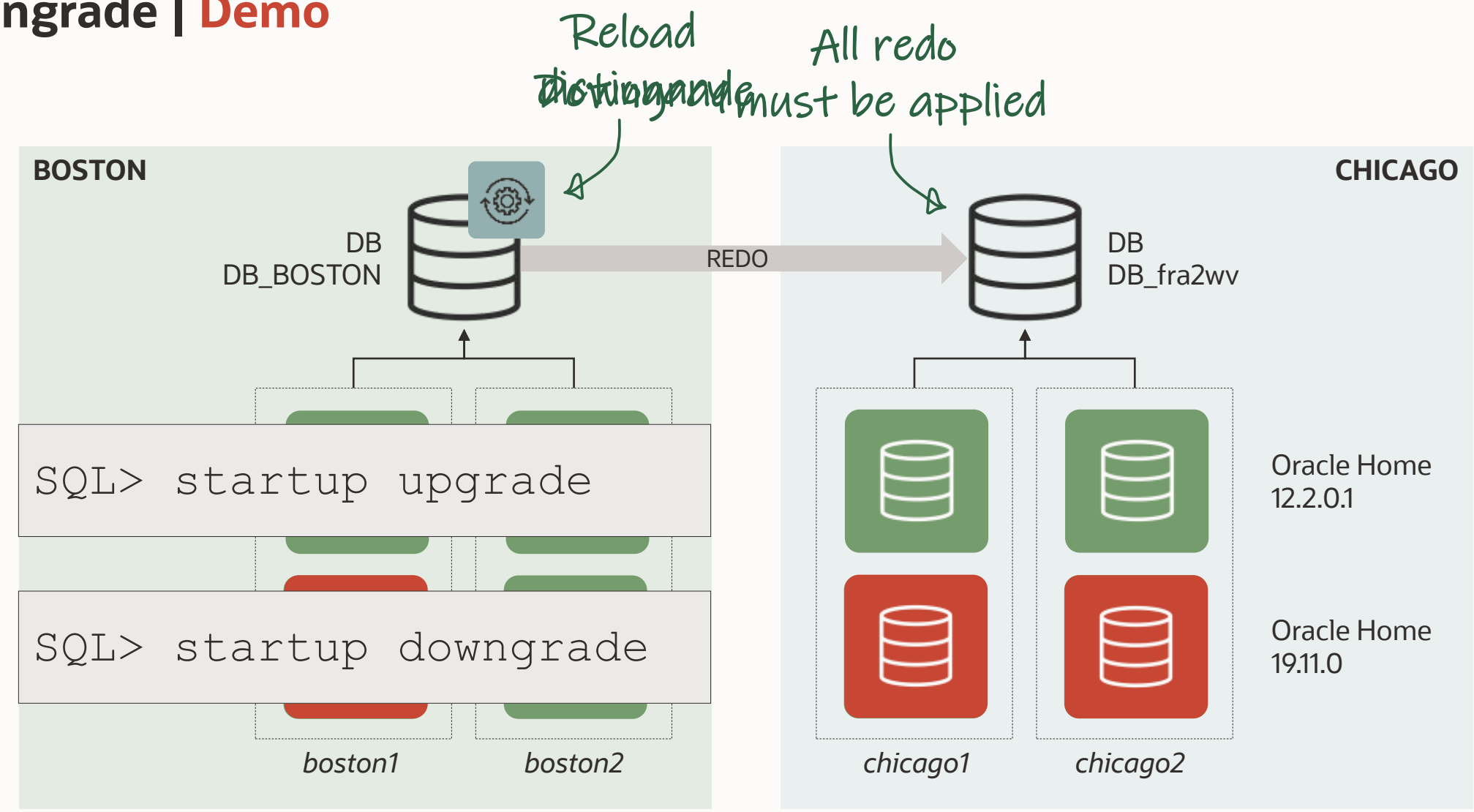
```
SQL> alter system set cluster_database=true scope=spfile sid='*';
SQL> shutdown immediate

$ srvctl downgrade database -d $ORACLE_UNQNAME -o <lower_release_home> -t 12.2.0.1
$ srvctl start database -d $ORACLE_UNQNAME
```

MOS note 2548962.1 contains detailed step-by-step instructions on downgrade

# Downgrade | Demo



# Downgrade | Demo



# Downgrade | Demo


Watch on YouTube

# Bug | Flashback and downgrade

## Restarting database in previous Oracle Home

```
$ $ORACLE_HOME/bin/srvctl start database -d $ORACLE_UNQNAME
PRCR-1079 : Failed to start resource ora.db_fra25c.db
CRS-5017: The resource action "ora.db_fra25c.db start" encountered the following error:
ORA-01078: failure in processing system parameters
LRM-00101: unknown parameter name '__unified_pga_pool_size'
. For details refer to "(:CLSN00107:)" in
"/u01/app/grid/diag/crs/chicago2/crs/trace/crsd_oraagent_oracle.trc".
```

## Create PFile in idle instance, then start up on corrected PFile

```
SQL> create pfile='/tmp/init.ora' from spfile

...

SQL> startup pfile='/tmp/init.ora';
```



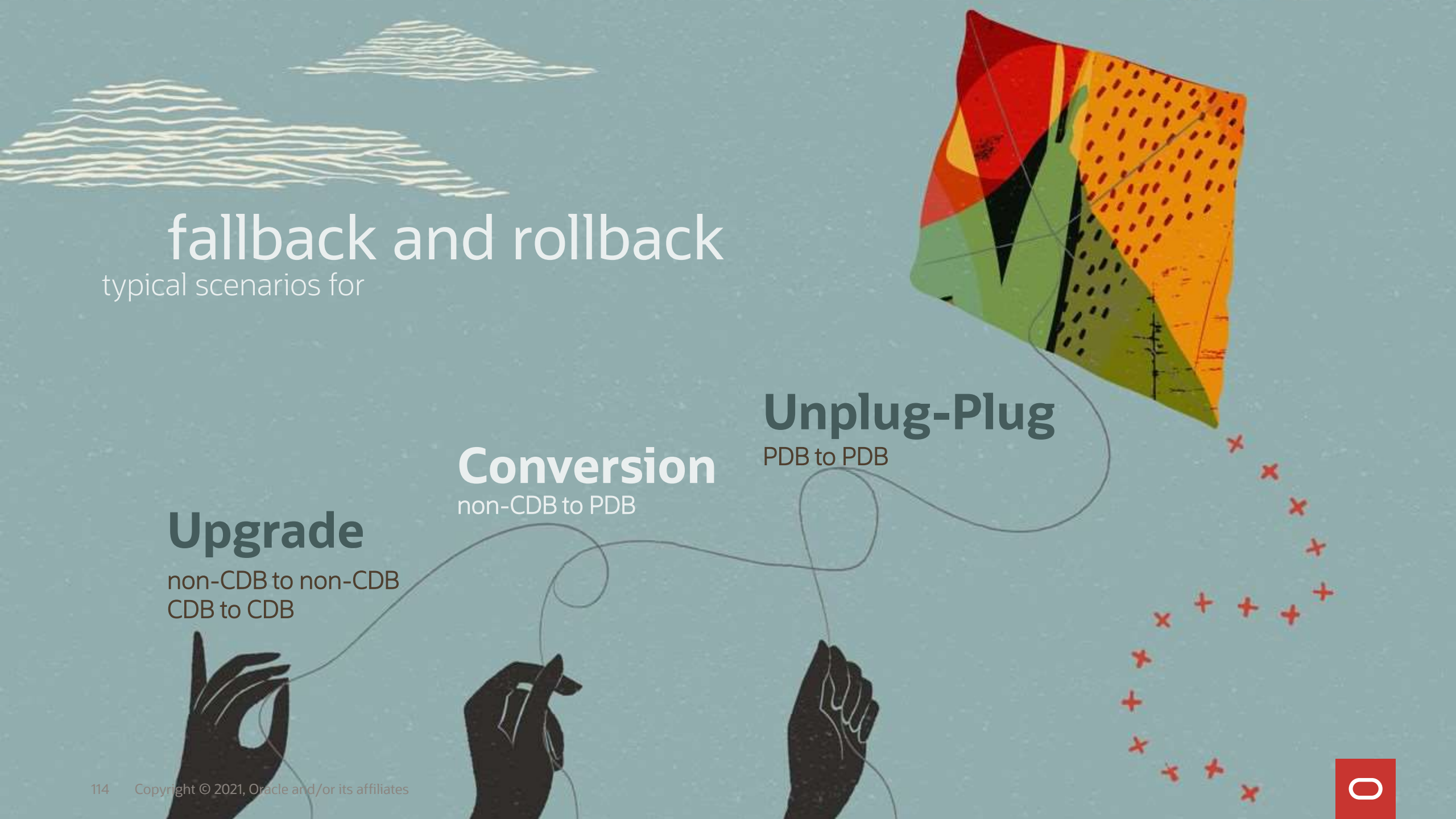

# Bug | Flashback and downgrade

- Parameter unknown to previous release
- Parameter gets written to SPFile when database is running new release
- New parameters should not be written to SPFile unless `COMPATIBLE` is raised
- Bug 30072483




# fallback and rollback

typical scenarios for



## Upgrade

non-CDB to non-CDB
CDB to CDB

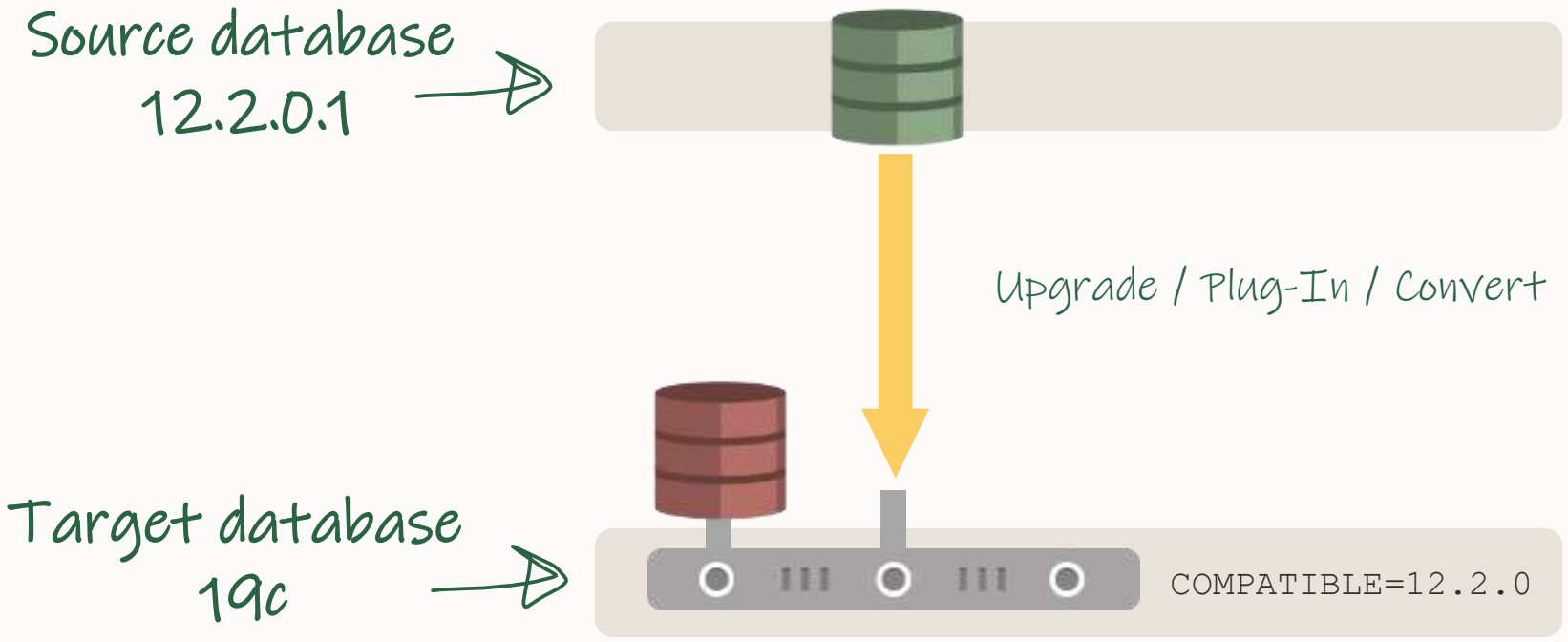
## Conversion

non-CDB to PDB

## Unplug-Plug

PDB to PDB

# 12.2 to 19c | Concept



# 12.2 to 19c | 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=cdb19c
```

## Command

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

Blog post: Oracle AutoUpgrade between two servers – and Plugin?

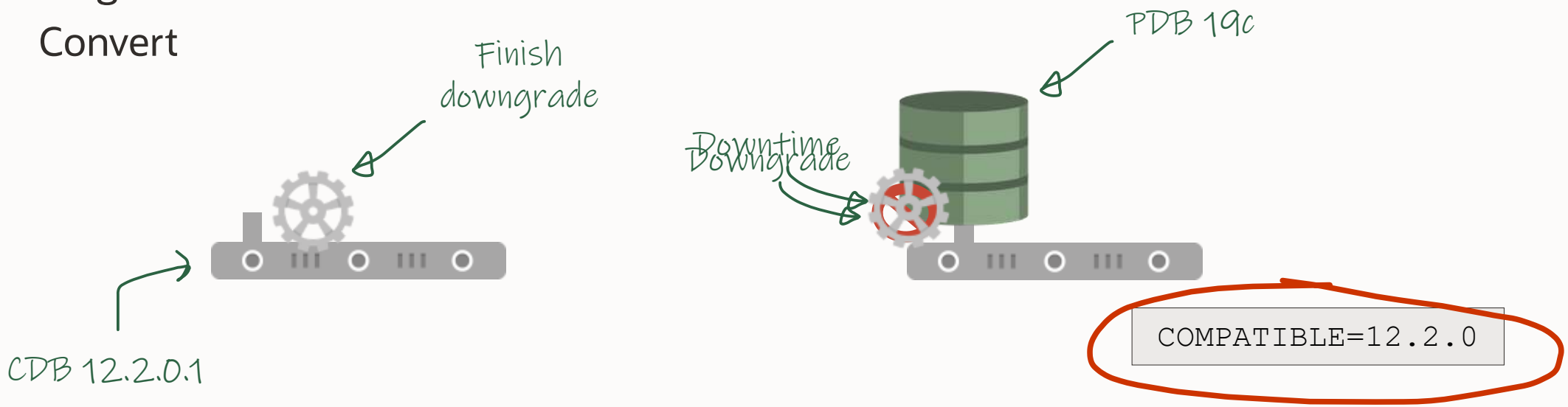
Pro tip: You can also plug in manually and upgrade PDB with `dbupgrade -c DB19`




# Fallback | Lower Release CDB

Source database: 12.2 CDB

- Upgrade
- Plug-in
- Convert



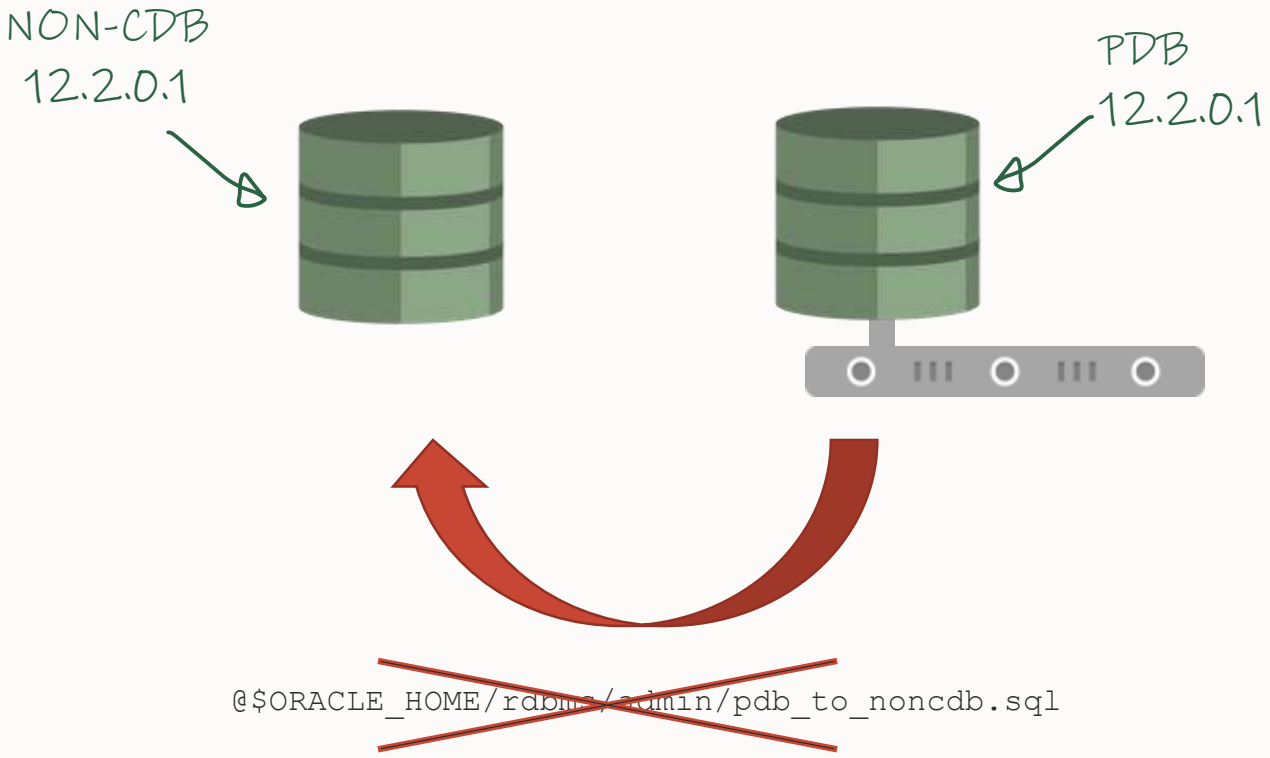
# Fallback | PDB Downgrade

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

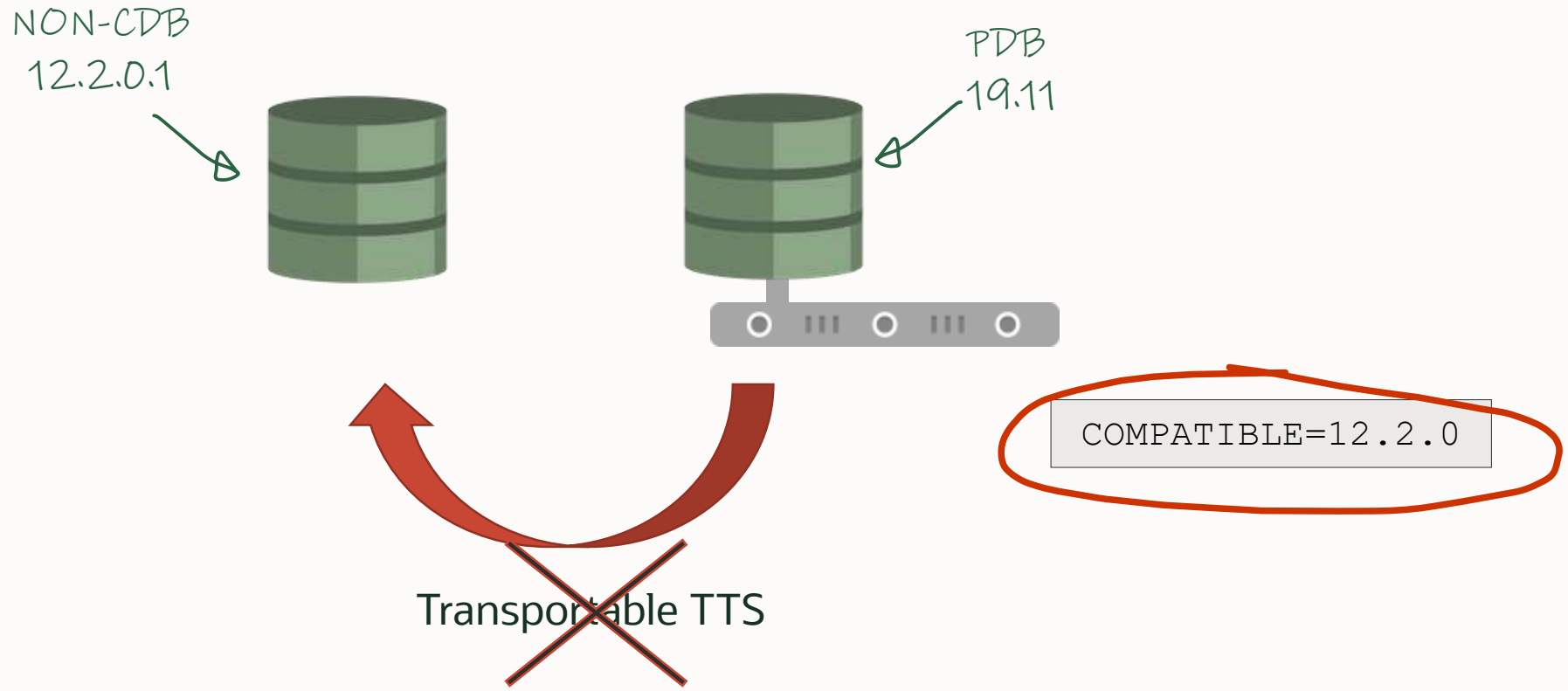
- Manual tasks
  - `catdwgrd.sql` in current (*after* upgrade) environment
  - `catrelod.sql` in previous (*before* upgrade) environment
  - Don't change `COMPATIBLE`

- MOS Note: 2172185.1
  How to Downgrade a Single Pluggable Oracle Database ( PDB ) to previous release

# Fallback | Lower Release Non-CDB



# Fallback | Lower Release Non-CDB



# Fallback | Transportable TTS

Not possible to lower releases

- Blog post

```
Import: Release 11.2.0.4.0 - Production on Mon Jul 12 16:42:50 2021

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
ORA-39001: invalid argument value
ORA-39000: bad dump file specification
ORA-39142: incompatible version number 4.1 in dump file "/u01/app/oracle/admin/orcl/dpdump/expdat.dmp"
```

**"While a transport tablespace to a lower release may work in some cases, it is not an action that Oracle supports."**

**Compatibility and New Features when Transporting Tablespaces with Export and Import (Doc ID 291024.1)**

# How to ...

Non-CDB to PDB conversion is irreversible

# Fallback | Concept

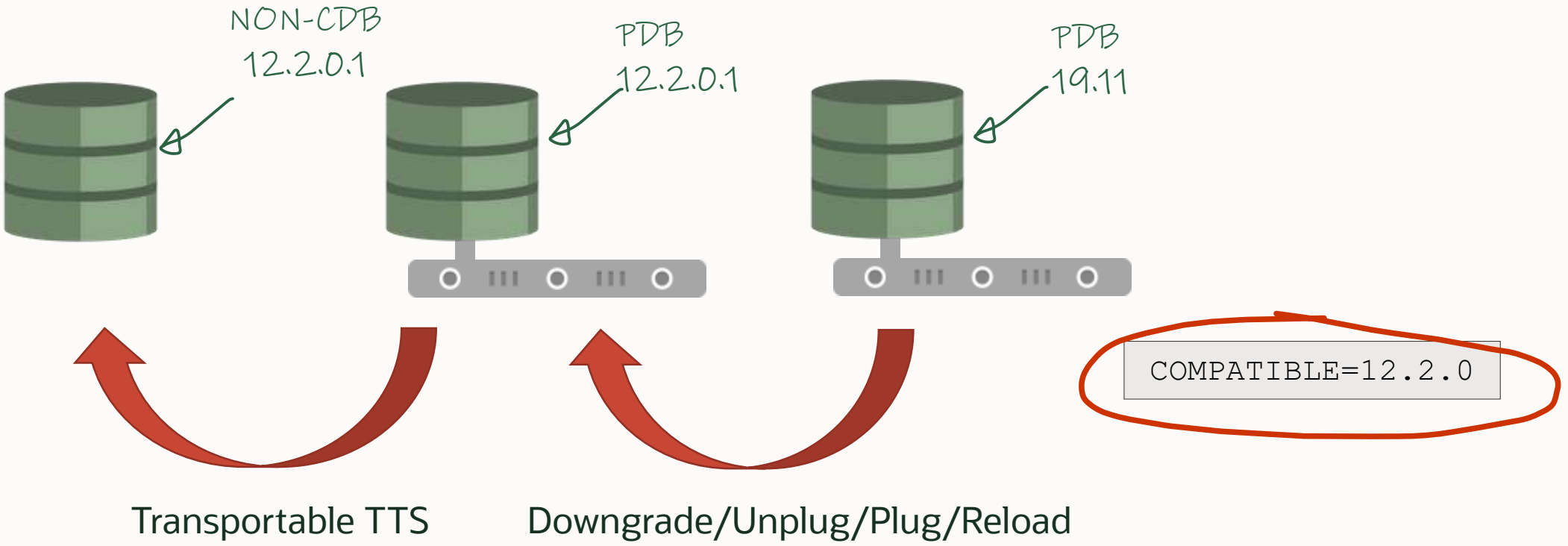
Move to Multitenant architecture = Migration

1. Usual fallback techniques don't work
   - No downgrade
   - No flashback to Restore Point

2. Only possible fallback options
   - Data Pump
   - GoldenGate
   - Transportable Tablespaces (only same version)
   - Plug into a source-version CDB




# Fallback | Lower Release Non-CDB

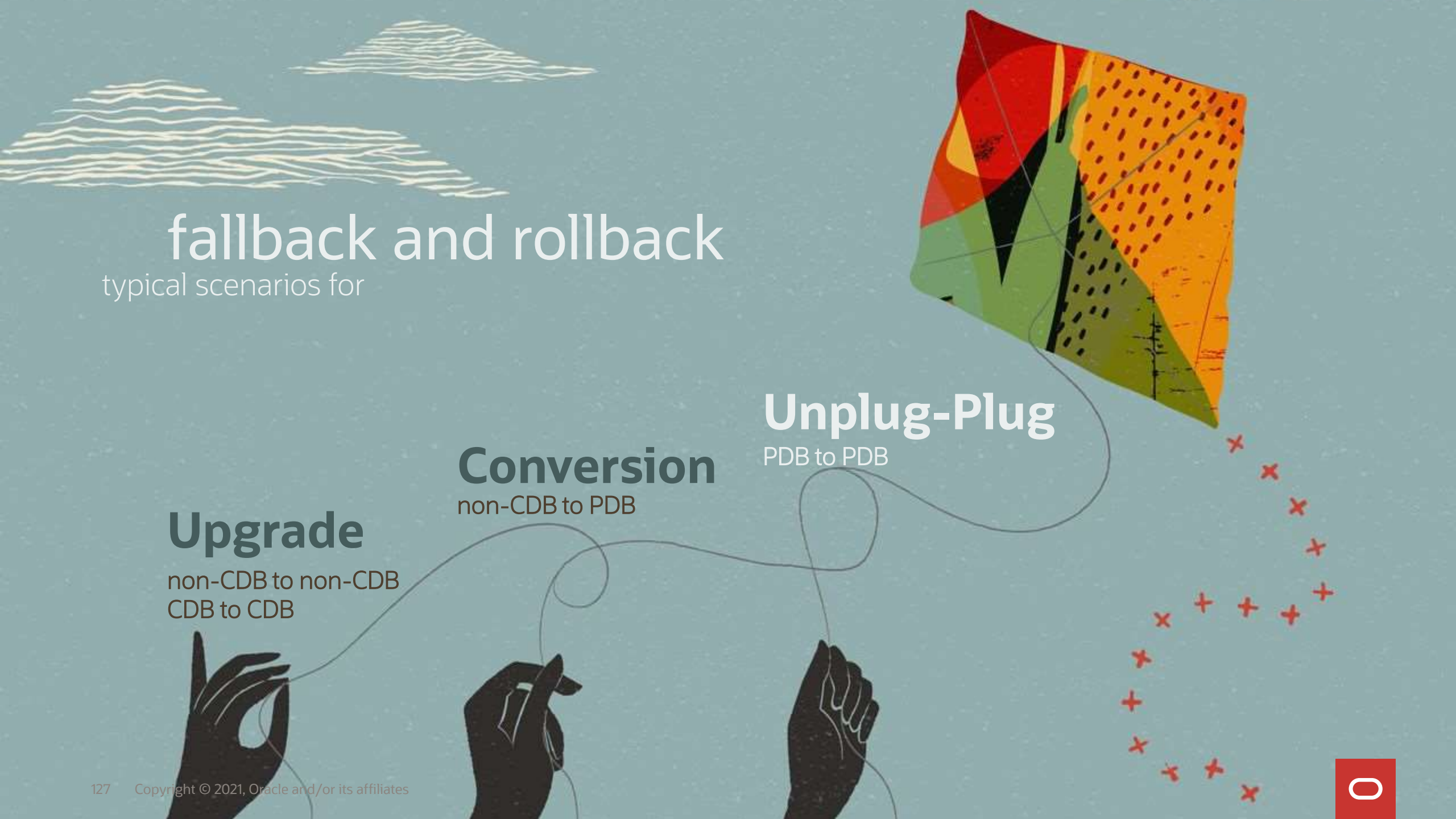


# Fallback | Full Process


Watch on YouTube

# fallback and rollback

typical scenarios for



## Upgrade

non-CDB to non-CDB
CDB to CDB

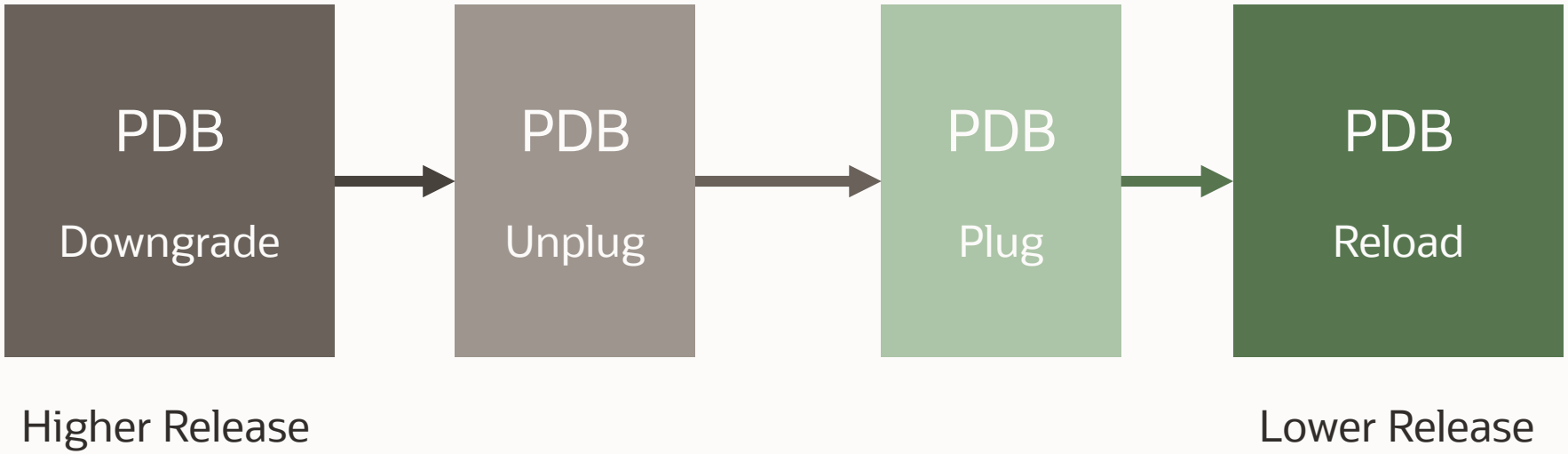
## Conversion

non-CDB to PDB

## Unplug-Plug

PDB to PDB

# Downgrade-Unplug-Plug | Concept



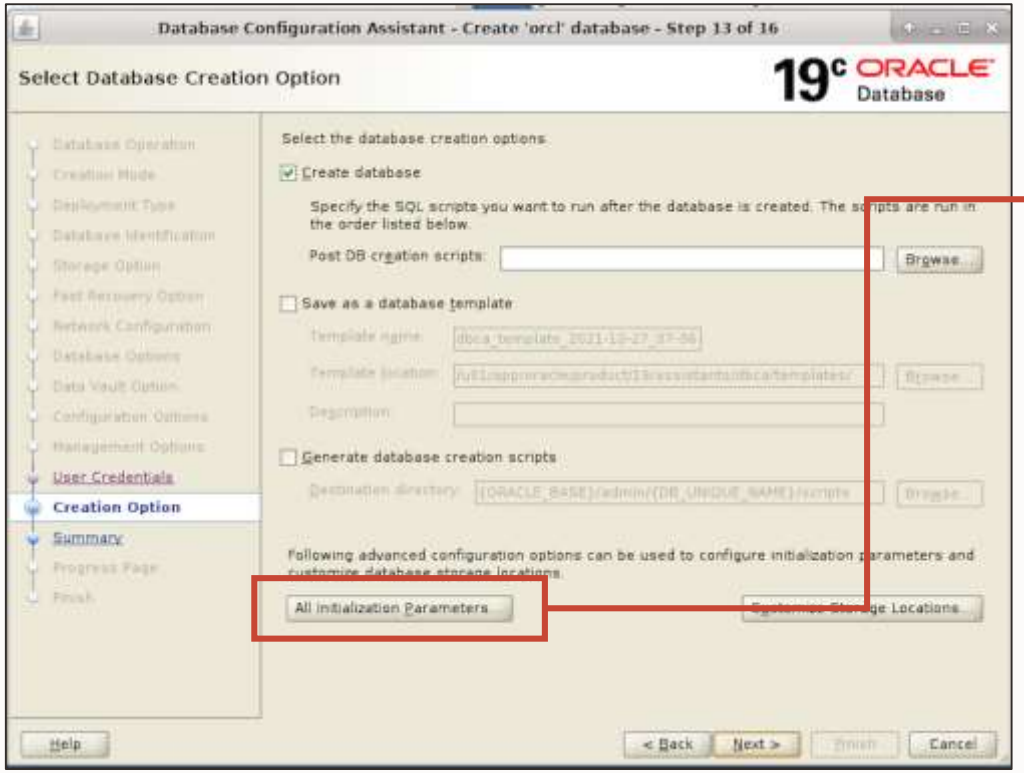
# Downgrade-Unplug-Plug | Silent Compatible Change

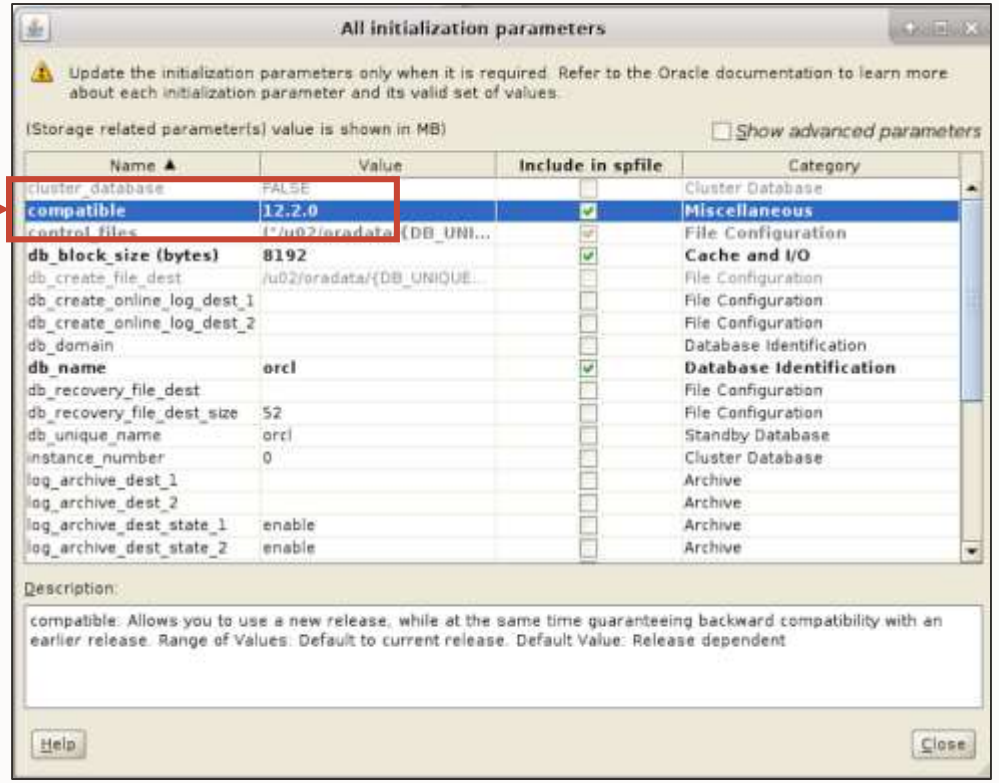
Beware of the silent `COMPATIBLE` change in Multitenant

# Downgrade-Unplug-Plug | Preparation

Target CDB must have the same `COMPATIBLE` setting

- CUSTOM database creation



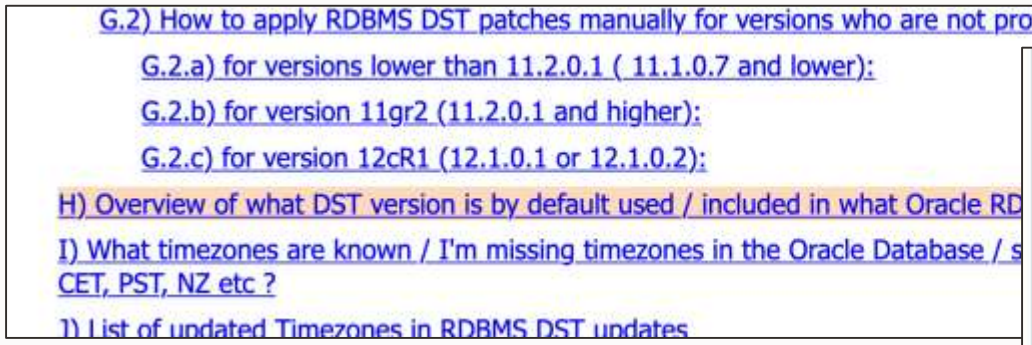
# Downgrade-Unplug-Plug | Preparation

Source and target CDB must have the identical time zone files present

- Check time zone file in source and target CDBs

```
select * from V$TIMEZONE_FILE;
```

- Install matching time zone patch to source or target home
  - MOS Note:412160.1




```
Version 26 - tzdata2016d update - patch 22873635
Version 27 - tzdata2016f update - patch 23614158
Version 28 - tzdata2016g update - patch 24701840
Version 29 - tzdata2016j update - patch 25173124
Version 30 - tzdata2017b update - patch 25881255
Version 31 - tzdata2017c update - patch 27015449
Version 32 - tzdata2018e update - patch 28125601
Version 33 - tzdata2018g update - patch 28852325
Version 34 - tzdata2019b update - patch 29997937
Version 35 - tzdata2020a update - patch 31335037
version 36 - tzdata2020e update - patch 32327201
```

# Downgrade-Unplug-Plug | Preparation

Source and target CDB must have the identical time zone files present

- Apply time zone patch to CDB$ROOT

```
SQL> start $ORACLE_HOME/rdbms/admin/utltz_upg_check.sql
SQL> start $ORACLE_HOME/rdbms/admin/utltz_upg_apply.sql
```

  - Attention: Restart will happen

- Check time zone file version again

```
select * from V$TIMEZONE_FILE;
```

# Downgrade-Unplug-Plug | No Compatible Change

Both CDBs must have identical `COMPATIBLE` settings

- Downgrade possibility

# Downgrade-Unplug-Plug | PDB Downgrade

Downgrade PDB in higher release CDB

- Cleanup unified audit trail

```
SELECT COUNT(*) FROM UNIFIED_AUDIT_TRAIL;
```

```
exec DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL(DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED, FALSE);
```

# Downgrade-Unplug-Plug | PDB Downgrade

Downgrade PDB in higher release CDB

- Shutdown PDB
- Start PDB in `DOWNGRADE` mode

```
alter pluggable database PDB1 open downgrade;
```

# Downgrade-Unplug-Plug | PDB Downgrade

Downgrade the PDB

- Logs default: `$ORACLE_HOME/cfgtoollogs/downgrade`

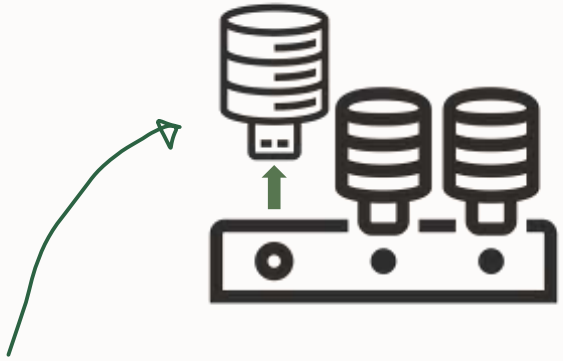
```
$> chmod +x $ORACLE_HOME/bin/dbdowngrade
```

```
$> dbdowngrade -c 'PDB1'
```

# Downgrade-Unplug-Plug | PDB Unplug

Unplug the PDB

- Shutdown
- Unplug



```
alter pluggable database PDB1 close;
```

```
alter pluggable database PDB1 unplug into '/tmp/pdb1.xml';
```

# Downgrade-Unplug-Plug | PDB Plugin

Plugin the PDB into the source CDB

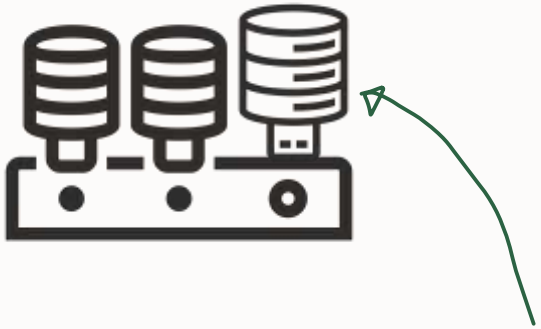
- Cleanup potentially existing files in destination
- `FILE_NAME_CONVERT` will trigger the copy operation

```
create pluggable database PDB1 using '/tmp/pdb1.xml'
  file_name_convert=('CDB2','CDB1');
```

# Downgrade-Unplug-Plug | PDB Reload

Reload all packages and code

- Open the PDB in `UPGRADE` mode
- Start reload script `catrelod.sql`



```
alter pluggable database PDB1 open upgrade;
```

```
alter session set container=PDB1;
```

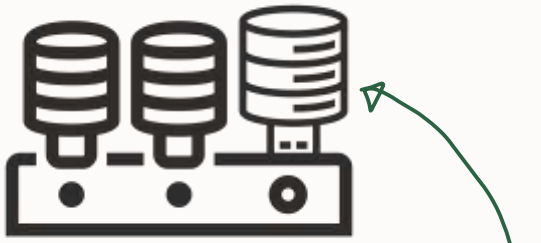
```
start ?/rdbms/admin/catrelod.sql
```

Pro tip: Spool the output of `catrelod.sql` into a logfile

# Downgrade-Unplug-Plug | PDB Recompilation

Recompilation

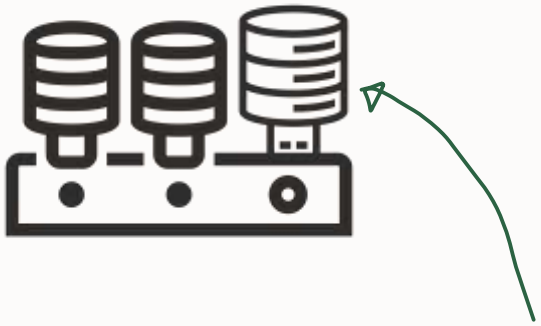
- Start `utlrp.sql`



```
@$ORACLE_HOME/rdbms/admin/utlrp.sql
```

# Downgrade-Unplug-Plug | Finalize

Stop and restart the PDB



```
alter pluggable database PDB1 close;
```

```
alter pluggable database PDB1 open;
```

Plan carefully and ensure `COMPATIBLE` and time zone are equal between source and target CDBs – and test it!

Can you use a refreshable clone PDB for downgrades?

# Downgrade | Refreshable Clone PDB

PDB has been plugged in and upgrade to 19c

- Refreshable clone back into previous release

```
SQL> create pluggable database PDB1 from PDB1@clonemypdb
     REFRESH MODE MANUAL file_name_convert=('CDB2','CDB1');
create pluggable database PDB1 from PDB1@clonemypdb REFRESH MODE MANUAL ...
*
ERROR at line 1:
ORA-65156: pluggable database version 19.0.0.0.0 not allowed
```

  - You can clone only to the same or a higher release CDB

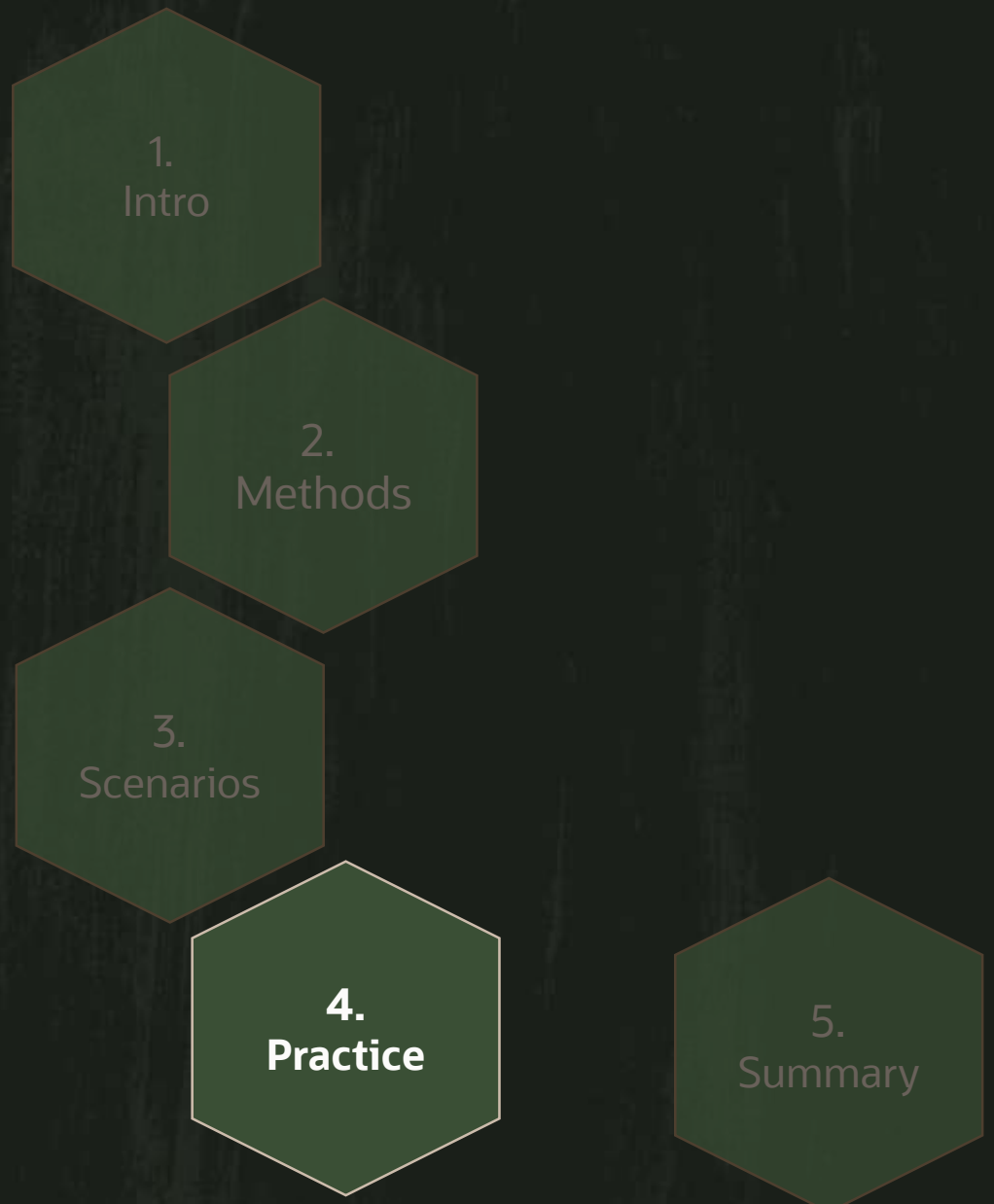



# Unplug-Plug-Downgrade | More Information

- MOS Note: 2421060.1
  How to Downgrade a Single Pluggable Oracle Database ( PDB ) from to previous release

- MOS Note: 2172185.1
  How to Downgrade a Single Pluggable Oracle Database ( PDB ) to previous release



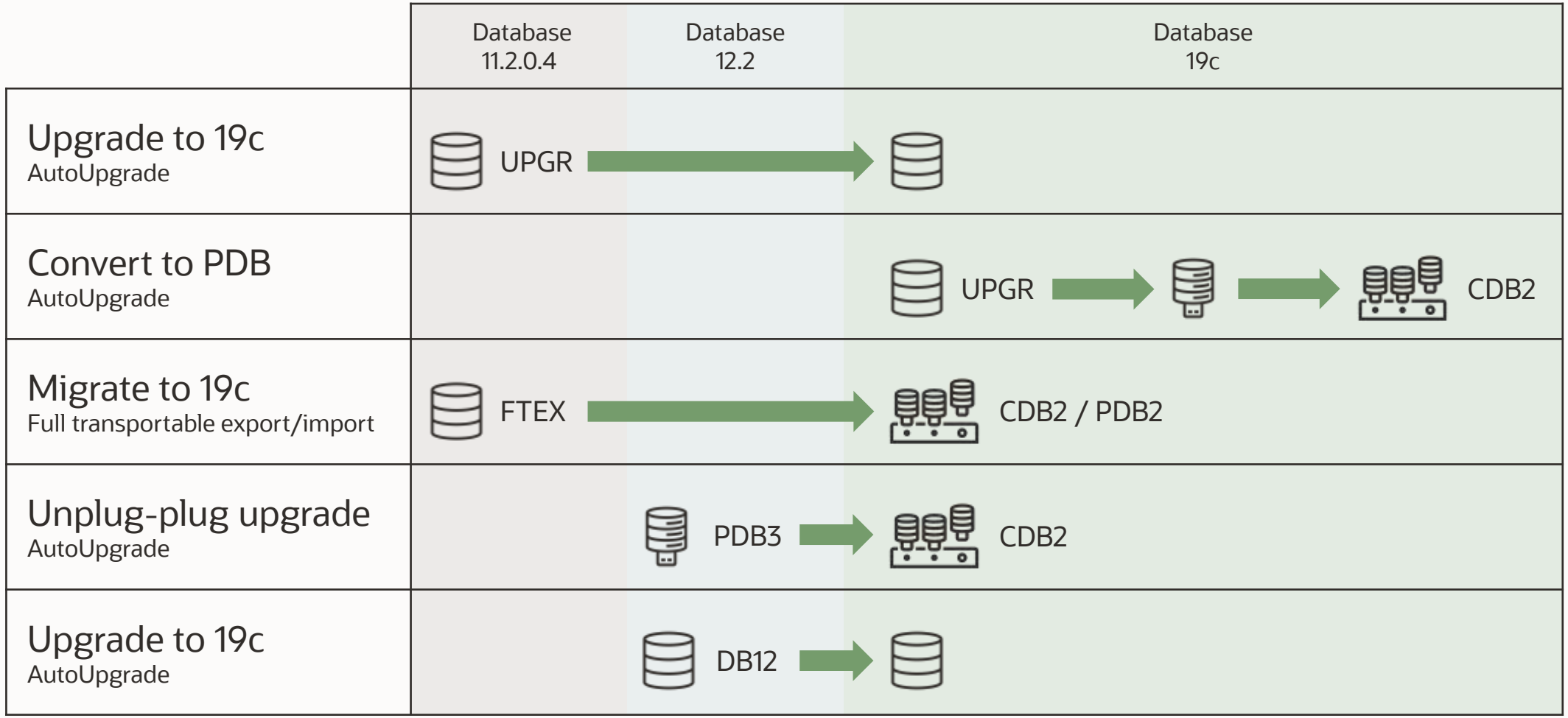



# fallback and rollback

## how to practice

Our Hands-On Lab has 15 guided exercise taking you from A-Z using five databases and three Oracle Homes

# Hands-On Lab



# TWO OPTIONS

## VIRTUAL BOX

Self-contained image

Runs on your laptop

70 GB

## LIVELABS

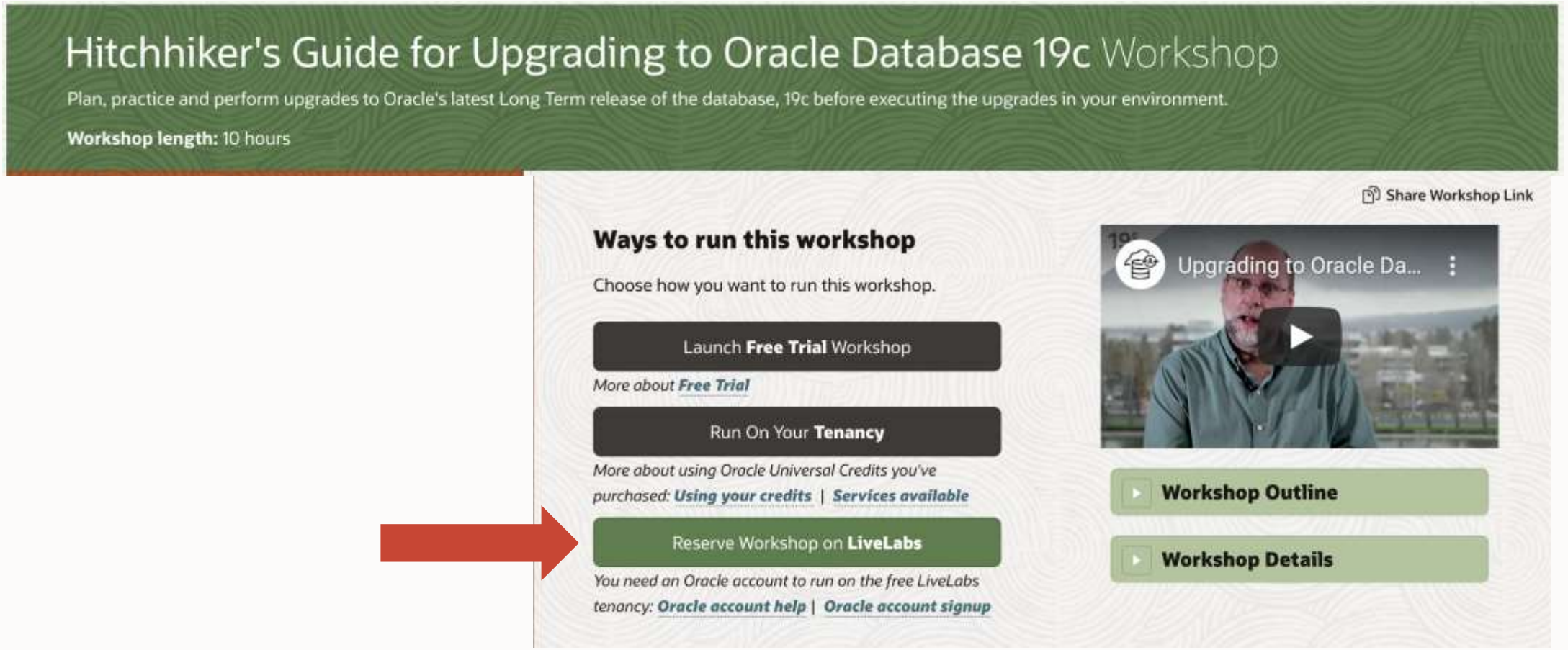
Runs from browser

Runs in OCI

Using Free Tier account

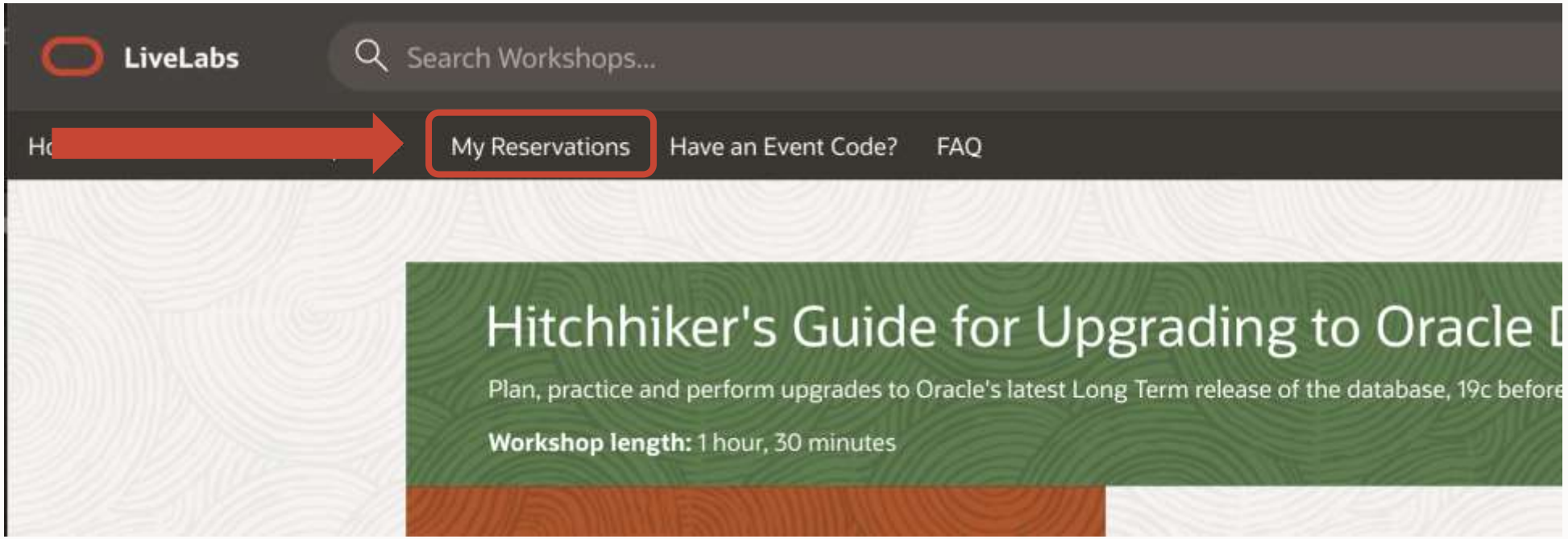
# LiveLabs

Hitchhiker's Guide to Database Upgrades



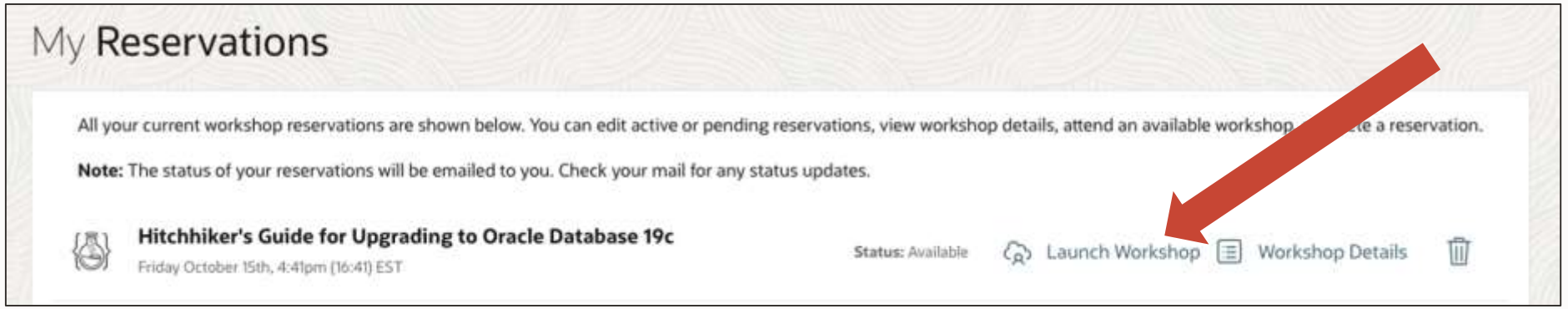
# LiveLabs

6-10 minutes later:



# LiveLabs

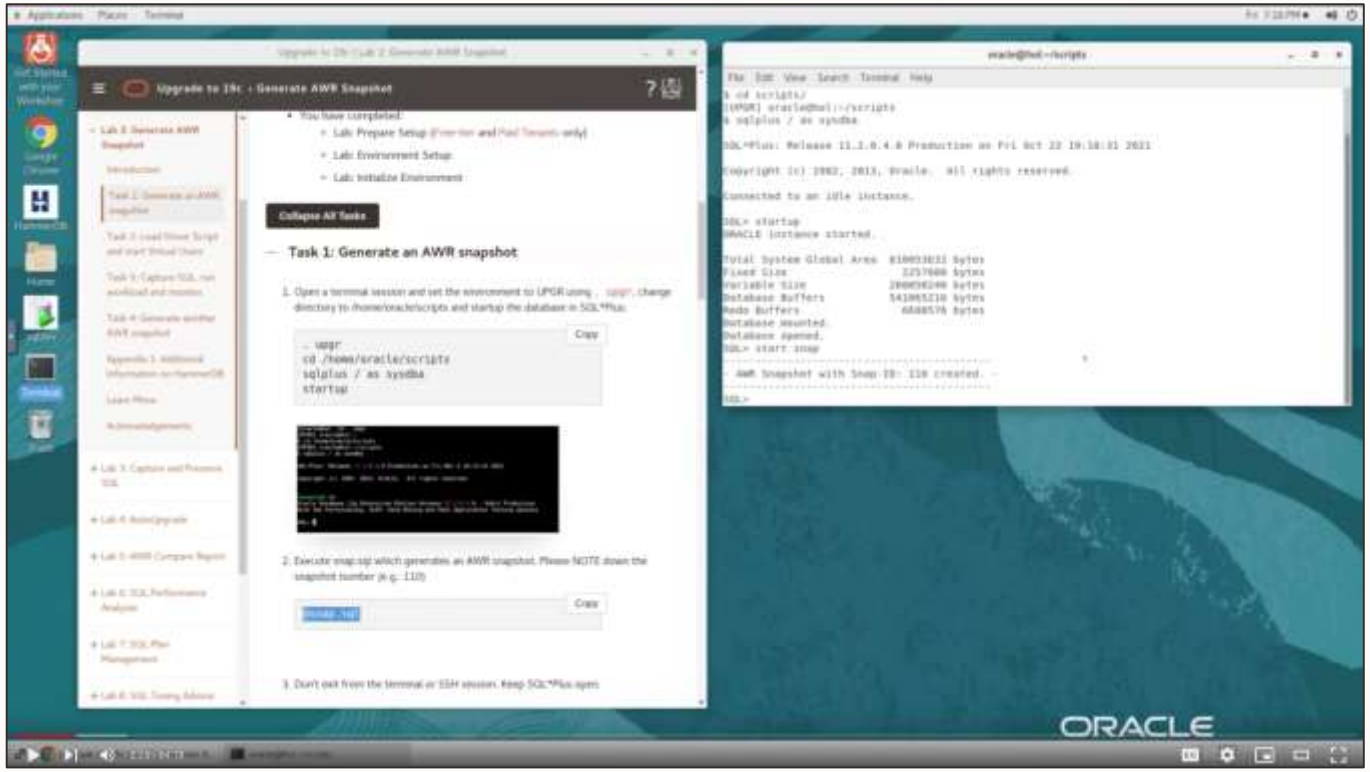
Then launch the workshop and access it within your browser (noVNC link)




# LiveLabs

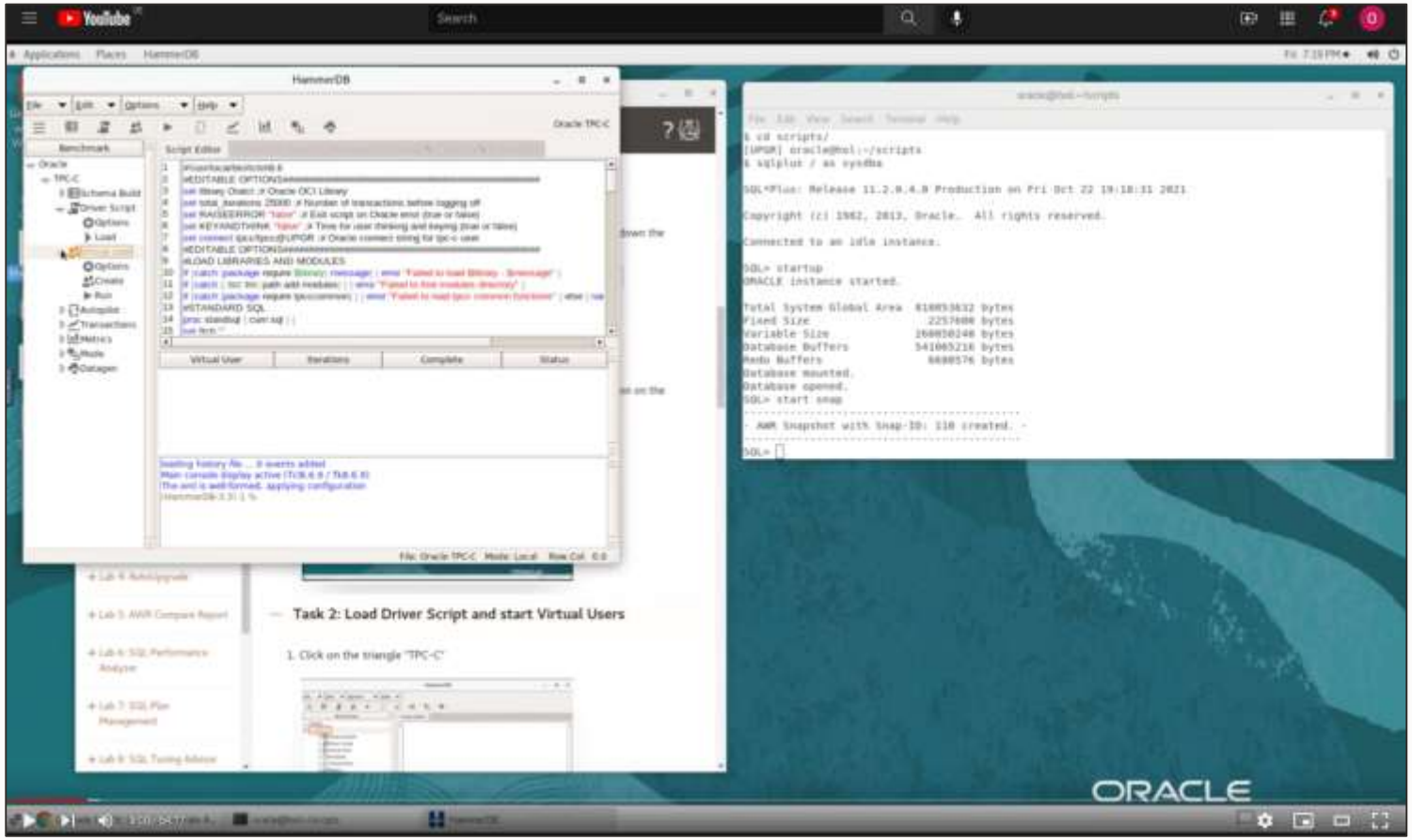
No installation of any tool is required if you use the Green Button lab

- Instructions are in the browser inside the lab



# LiveLabs

Find a narrated 54 min video covering the entire lab on YouTube



# Oracle Cloud Infrastructure

Lift your database into the cloud and exercise fallback and rollback

- https://mikedietrichde.com/videos/

Episode 6

Move to the Cloud – Not only for techies

*115 minutes – Apr 8, 2021*


Episode 9

Performance Testing Using the Oracle Cloud for Upgrades and Migrations

*90 minutes – May 19, 2021*

*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

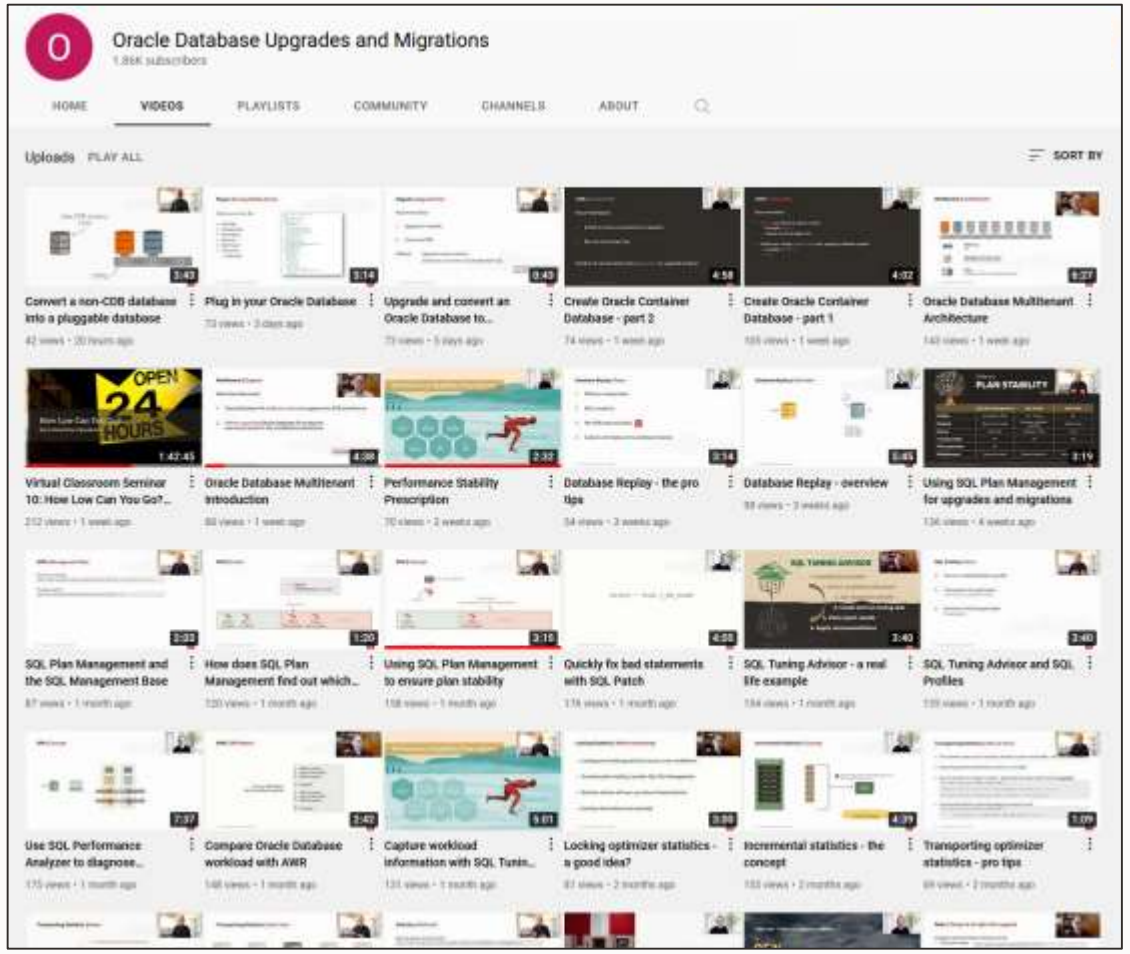
Database Upgrade Internals – and so much more

## Recorded Web Seminars

https://MikeDietrichDE.com/videos/

https://dohdatabase.com/webinars/

# YouTube | Oracle Database Upgrades and Migrations




Link

- 100+ videos
- New videos every week
- No marketing
- No buzzword
- 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:**

OracleDatabaseUpgradesandMigrations

# THANK YOU

**MIGRATING VERY LARGE DATABASES**

Dec 9, 2021 – 10:00h CET

# THANK YOU