ORACLE

AutoUpgrade 2.0

A big step into the future of database upgrades

Daniel Overby Hansen
Senior Principal Product Manager

Mike Dietrich
Senior Director





MIKE DIETRICH
Senior Director
Database Upgrade and Migrations

mikedietrich

@mikedietrichde

https://mikedietrichde.com





DANIEL OVERBY HANSEN
Senior Principal Product Manager
Cloud Migration

dohdatabase

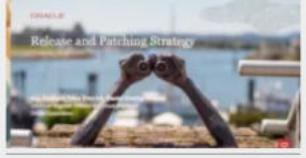
@dohdatabase

https://dohdatabase.com

NEW Episode 1

Release and Patching Strategy

105 minutes – Feb 4, 2021



NEW Episode 2

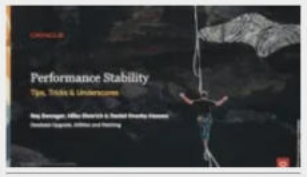
AutoUpgrade to Oracle Database 19c

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

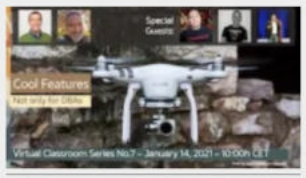
Move to the Cloud – Not only for techies

115 minutes – Apr 8, 2021

NEW Episode 7

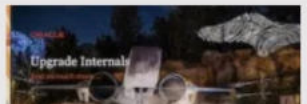
Cool Features – Not only for DBAs

110 minutes – Jan 14, 2021



NEW Episode 8

Database Upgrade Internals – and so much more



Recorded Web Seminars

https://MikeDietrichDE.com/videos/

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

ORACLE

CloudWorld

October 17-20 Las Vegas

#OCW22

oracle.com/cloudworld

Always use the latest version of AutoUpgrade

Download from My Oracle Support (2485457.1)

```
$ java -jar autoupgrade.jar -version

build.version 22.4.220712
build.date 2022/07/12 11:27:00 -0400
build.hash 161fde38
build.hash_date 2022/07/12 06:09:51 -0400
build.supported_target_versions 12.2,18,19,21
build.type production
```


Agenda

1

Patching

Database patching
Simple
Easy
Familiar

2

Refreshable Clone

3

Encryption

4

Distributed Upgrade

5

Usability

We made upgrading easy.
Now we make patching just as easy.

AutoUpgrade functionality extended to patching

Patching

1

Install Oracle Home including Release Update and additional patches (MOS Doc ID 555.1)

2

Create a simple configuration file

3

Start AutoUpgrade in deploy mode

```
$ cat DB19.cfg

patch1.source_home=/u01/app/oracle/product/19.0.0.0/dbhome_19_15_0
patch1.target_home=/u01/app/oracle/product/19.0.0.0/dbhome_19_16_0
patch1.sid=DB19


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


Patching

USE

Familiar interface

Console

Logging

ANALYZE

Prechecks

Summary report

PROTECT

Resumable

Restoration

Restore point

Fallback

AUTOMATE

`srvctl`

`/etc/oratab`

Files

Datapatch

Patching

Encryption

Hot clone

Refreshable clone

RAC

Proactive fixups

Distributed upgrade

...

Patching

What's missing

Windows

RAC rolling

Data Guard standby-first

Patching

AutoUpgrade

Automate your patching process and benefit from the familiar AutoUpgrade

Fleet Patching and Provisioning

Go fleet scale with FPP and benefit from additional functionality like deployment of Oracle Home

Agenda

1

Patching

2

Refreshable Clone

Unplug-plug upgrades
Non-CDB to PDB
Minimal downtime

3

Encryption

4

Distributed Upgrade

5

Usability

Non-CDB to PDB conversion is irreversible

What are your fallback options?

FALLBACK

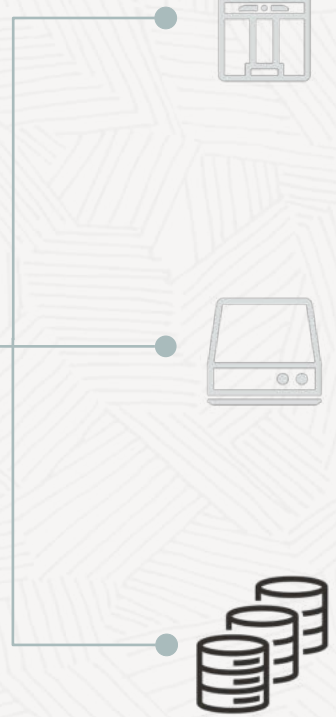
Backup / restore

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

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

--Specify relative start time
--upg1.start_time=+1h30m
```

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

--Specify relative start time
--upg1.start_time=+1h30m
```

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=22/10/2022 02:00:00
--Specify relative start time
--upg1.start_time=+1h30m
```

Refreshable Clone


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

```
upg1.start_time=22/10/2022 02:00:00
```

The source non-CDB stays intact to allow fallback

Works for unplug-plug upgrades as well

DEMO

Unplug-plug upgrade via refreshable clone PDB

Watch on YouTube

Agenda

1

Patching

2

Refreshable Clone

3

Encryption

Fully supported
Tablespace Encryption
Dedicated keystore

4

Distributed Upgrade

5

Usability

Upgrading and converting
encrypted databases are fully supported

Encryption

Certain database operations require passwords or secrets

```
CREATE PLUGGABLE DATABASE ... KEYSTORE IDENTIFIED BY <password>

ALTER PLUGGABLE DATABASE ... UNPLUG INTO ... ENCRYPT USING <secret>

CREATE PLUGGABLE DATABASE ... DECRYPT USING <secret>

ADMINISTER KEY MANAGEMENT ... KEYSTORE IDENTIFIED BY <password>
```


Secure External Password Store

Operator stores database keystore password
in a Secure External Password Store

AutoUpgrade Keystore

Operator loads database keystore password
into AutoUpgrade keystore ahead of upgrade

Encryption

To configure an AutoUpgrade keystore

```
$ cat DB12.cfg

global.keystore=/etc/oracle/keystores/autoupgrade/DB12
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
```


Encryption

Analyze the database for upgrade readiness

```
$ java -jar autoupgrade.jar -config PDB1.cfg -mode analyze
```

Summary report will show which keystore passwords are needed

```
REQUIRED ACTIONS
================
      1.  Perform the specified action ...
      ORACLE_SID                      Action Required
      ------------------------------  -------------------------
      CDB1                            Add TDE password
      CDB2                            Add TDE password
```

DEMO

Upgrading an encrypted database and converting to PDB

Watch on YouTube

Agenda

1

Patching

2

Refreshable Clone

3

Encryption

4

Distributed Upgrade

RAC feature
Performance
PDB availability

5

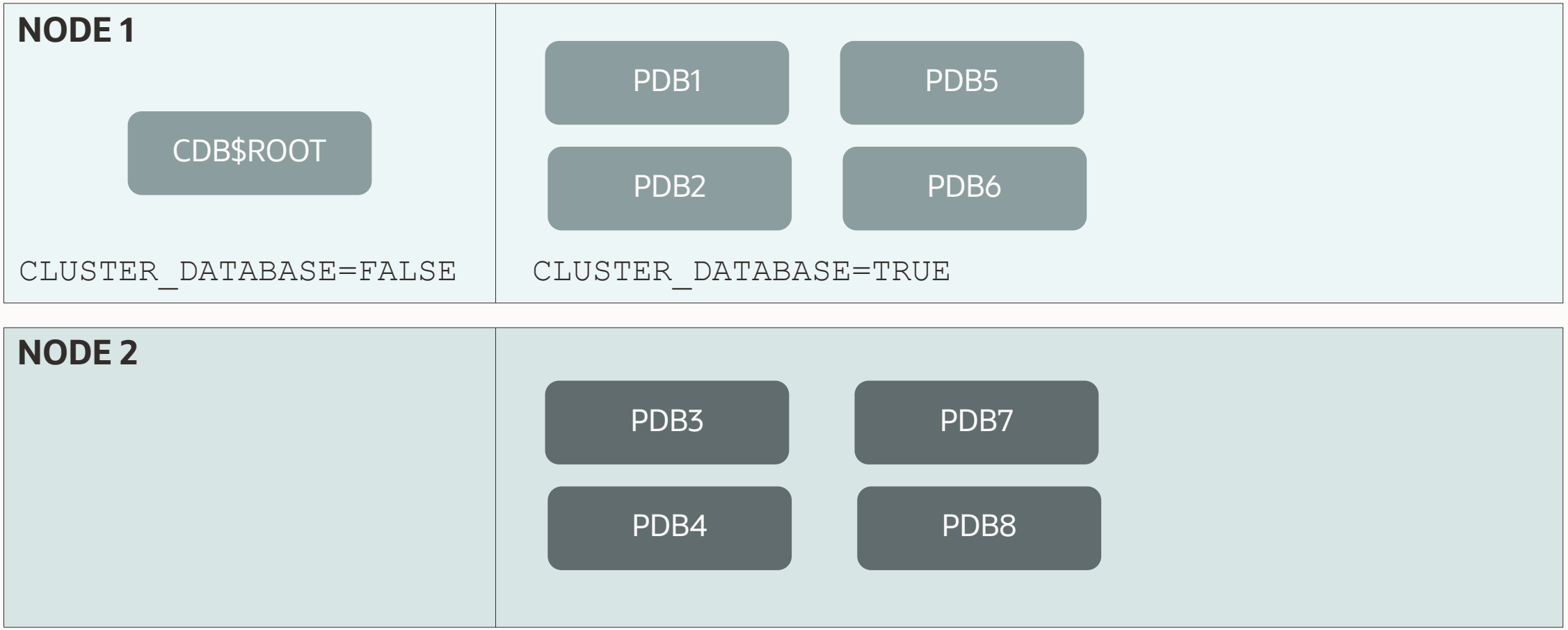
Usability

Distributed upgrade uses all nodes resulting in faster upgrades of CDBs

Applies to RAC and multitenant architecture only

Distributed Upgrade

Distributed Upgrade



Distributed Upgrade

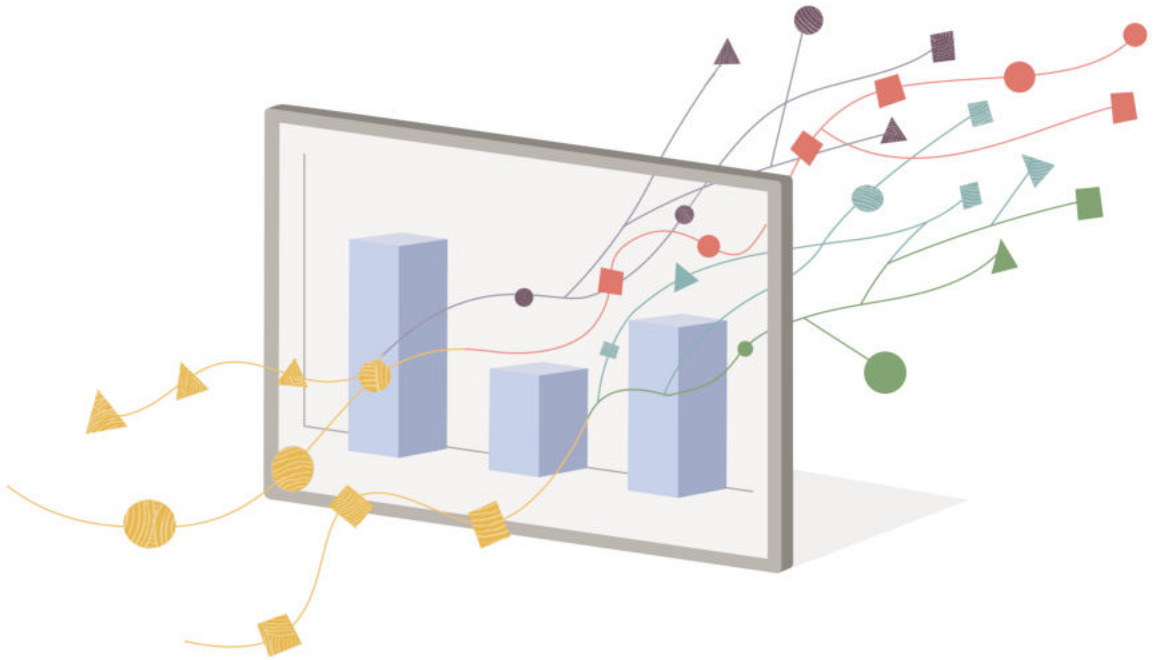
To enable distributed upgrade

```
$ cat RACDB.cfg

global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
upg1.log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade/RACDB
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true


$ java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

41%

In benchmark, time saved
by using distributed upgrade

- 2 node RAC database
- 4 CPUs each
- CDB with 8 PDBs



By default,
AutoUpgrade uses two nodes

Distributed Upgrade

To enable more nodes

```
$ cat RACDB.cfg

global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
upg1.log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade/RACDB
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=RACDB
upg1.tune_setting=distributed_upgrade=true,active_nodes_limit=n


$ java -jar autoupgrade.jar -config RACDB.cfg -mode deploy
```

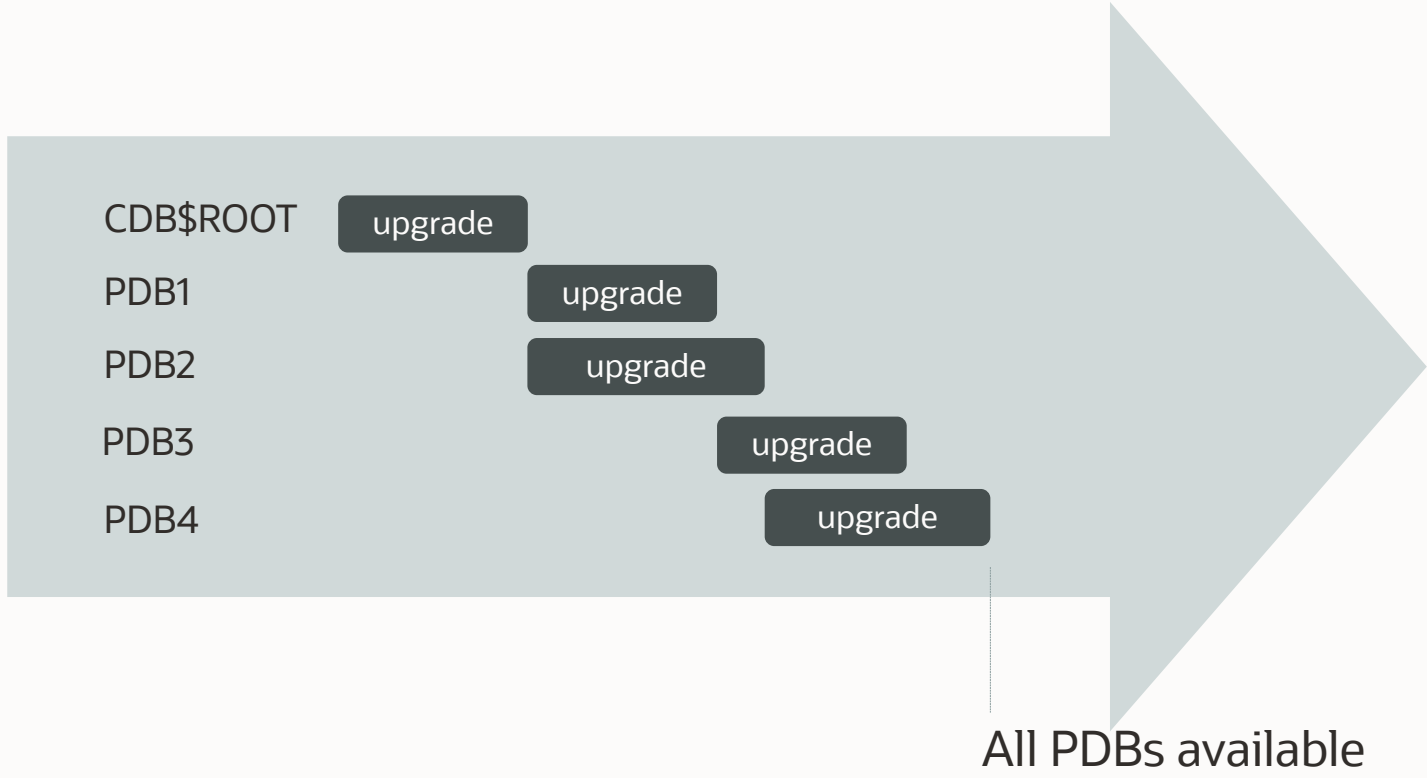
Some PDBs are more important

Control the order of upgrade

PDB Availability

DEFAULT

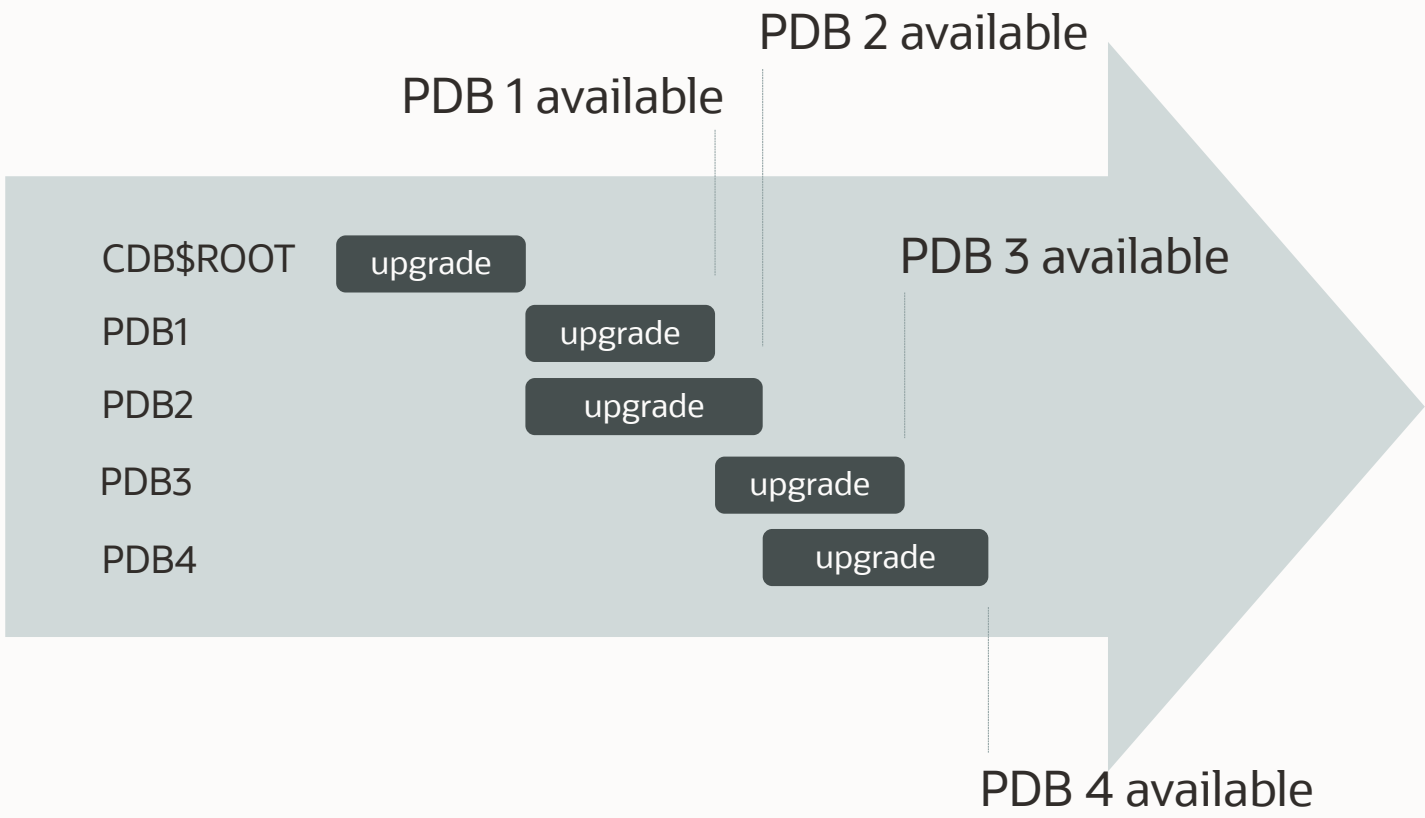
```
make_pdbs_available=false
```



PDB Availability

IMMEDIATELY AVAILABLE

`make_pdbs_available=true`

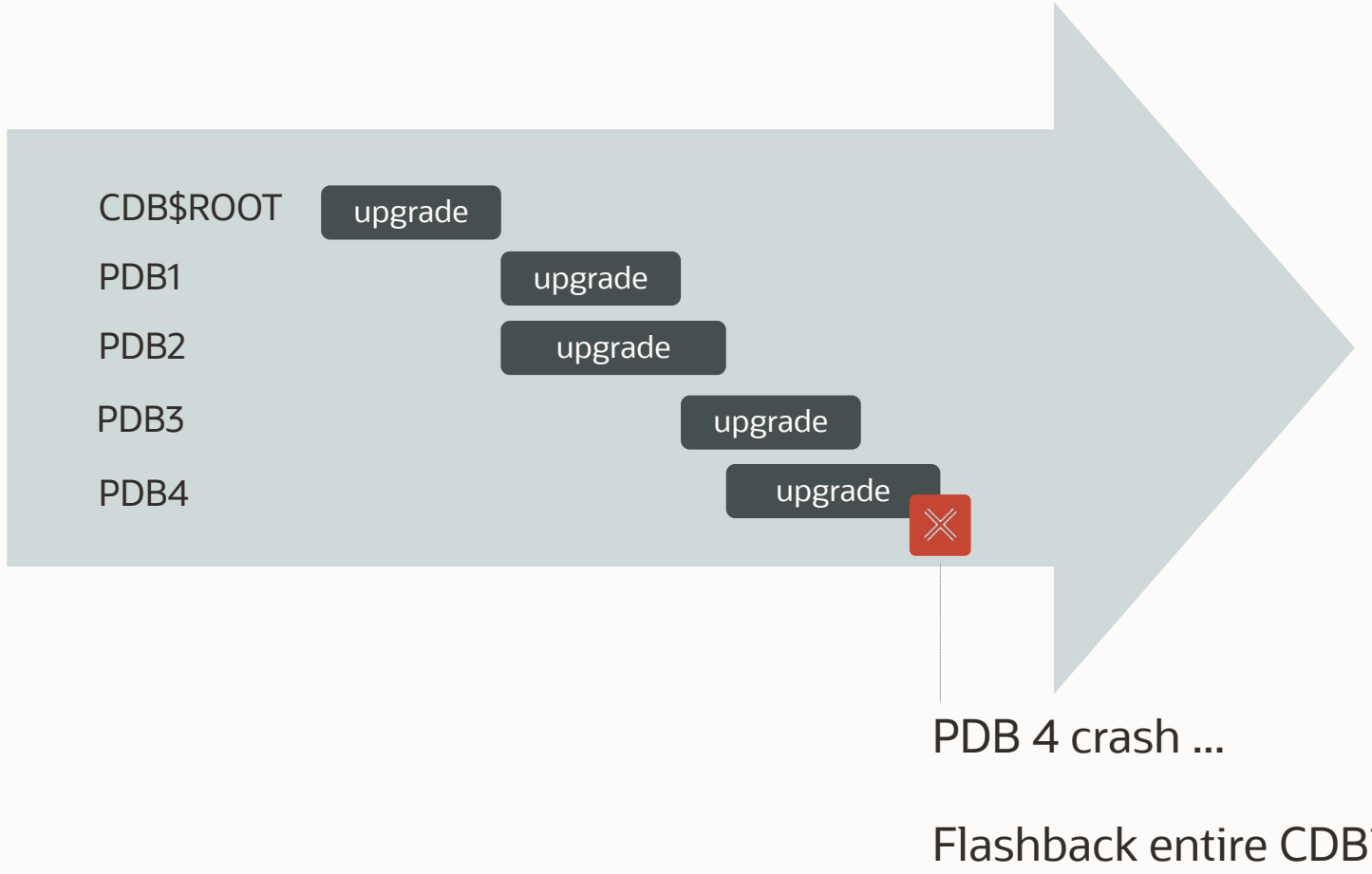


```
alter pluggable database SALESPROD priority 1;

alter pluggable database SALESDEV priority 2;

alter pluggable database SALESUAT priority 2;

alter pluggable database SALESTEST priority 3;
```

PDB Availability

IMMEDIATELY AVAILABLE

`make_pdbs_available=true`



Agenda

1 Patching

2 Refreshable Clone

3 Encryption

4 Distributed Upgrade

5 **Usability**

Repeating commands
History
Fixup list

DEMO

Enhancing the console experience

Watch on YouTube

```
--Repeat lsj command every 10 seconds
lsj -a 10

--Repeat status command every 10 seconds
status –jobid n –a 10

--Repeat last command
/

--Show history of commands
h

--Repeat command number n from history
/n
```

```
--Show fixups for a job
fxlist –job n

--Disable a fixup
--Example: fxlist -job 100 -c DB12 alter OLD_TIME_ZONES_EXIST run no
fxlist -job n -c <container> alter <fixup> run no
```


1

AutoUpgrade 2.0

A big step into the future of database upgrades

Tuesday 20 September 09:15 Kopenhagen

2

No Slides Zone

Upgrade to Oracle 19c and migrate to Multitenant

Tuesday, 20 September 14:00 Kopenhagen

3

Patch me if you can!

Database Patching News, Improvements and Secrets

Wednesday, 21 September 10:00 Tokio


ORACLE