



ORACLE

AutoUpgrade Deep Dive

Upgrade to Oracle Database 19c

Mike Dietrich

Distinguished Product Manager
Database Upgrade and Migrations

Daniel Overby Hansen

Senior Principal Product Manager
Database Cloud Migrations



Mike Dietrich

Distinguished Product Manager
Database Upgrade and Migrations

 <https://MikeDietrichDE.com>

 MikeDietrich

 @MikeDietrichDE



Daniel Overby Hansen

Senior Principal Product Manager
Database Cloud Migrations

 <https://dohdatabase.com>

 dohdatabase

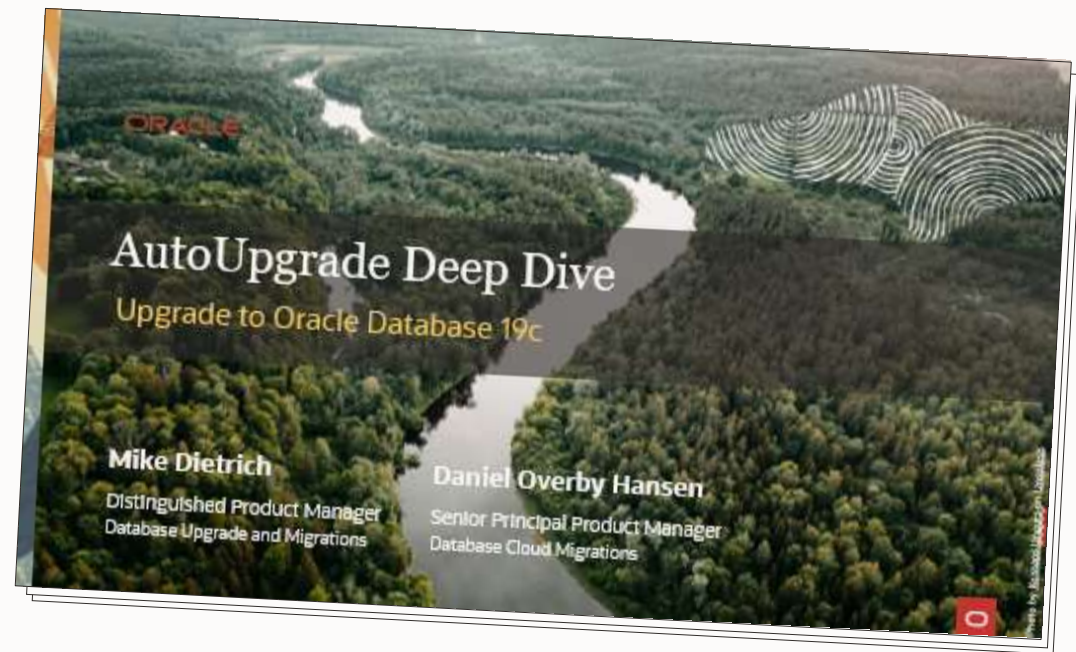
 @dohdatabase



Get the slides

<https://dohdatabase.com/slides>

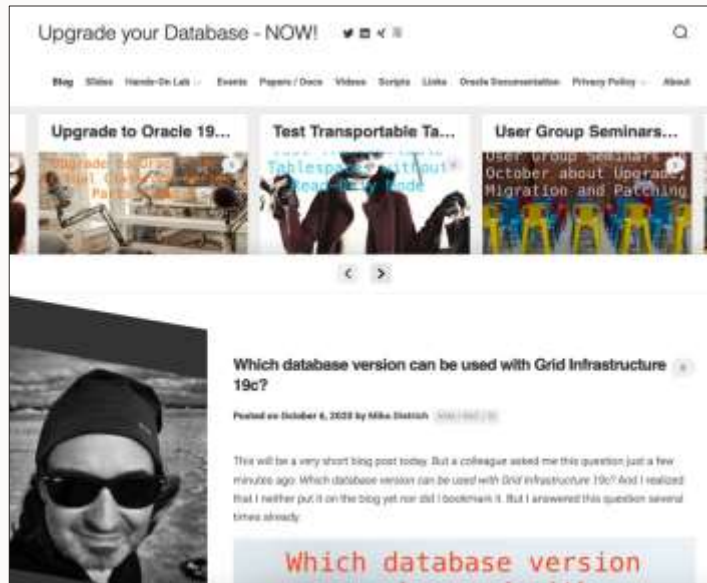
<https://MikeDietrichDE.com/slides>



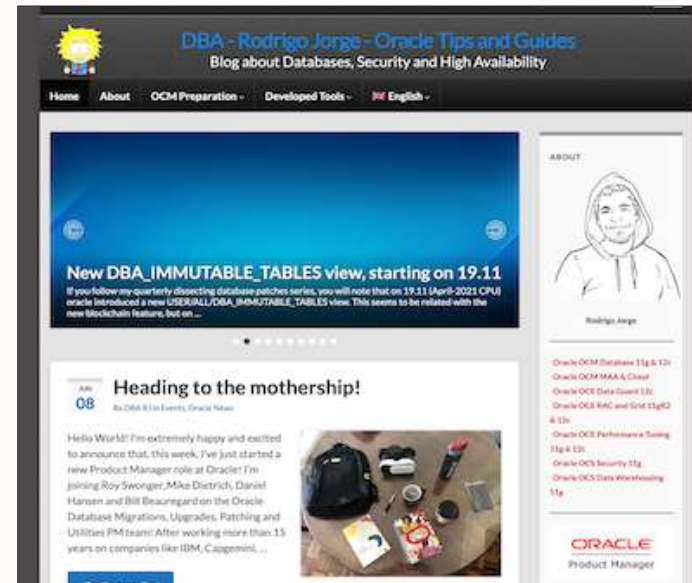
Visit our Blogs



<https://MikeDietrichDE.com>



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



<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/>

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



Chapter 1

Release and Patching Strategy



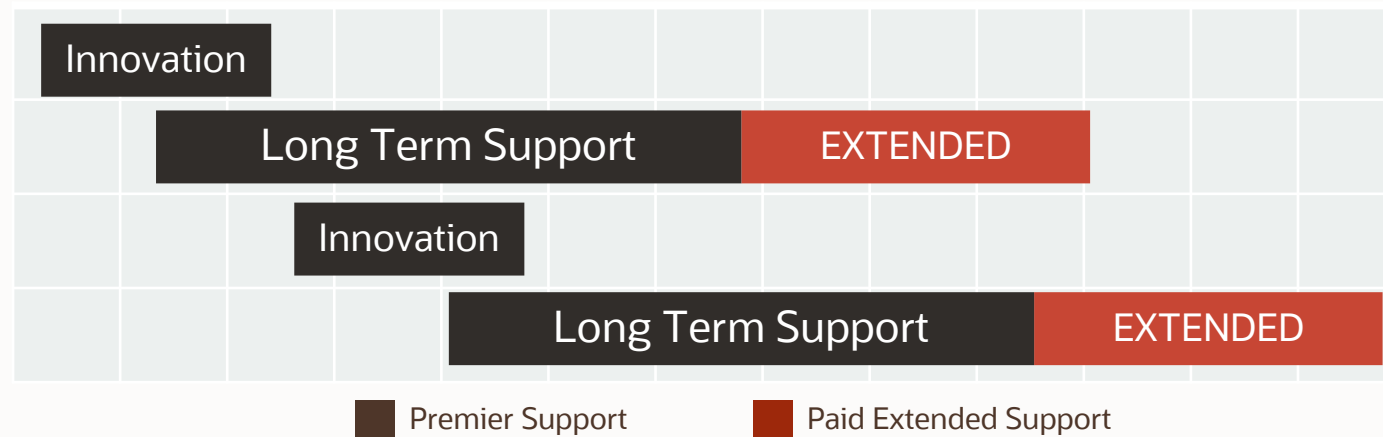
Release Types | Long Term Support vs Innovation Releases

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



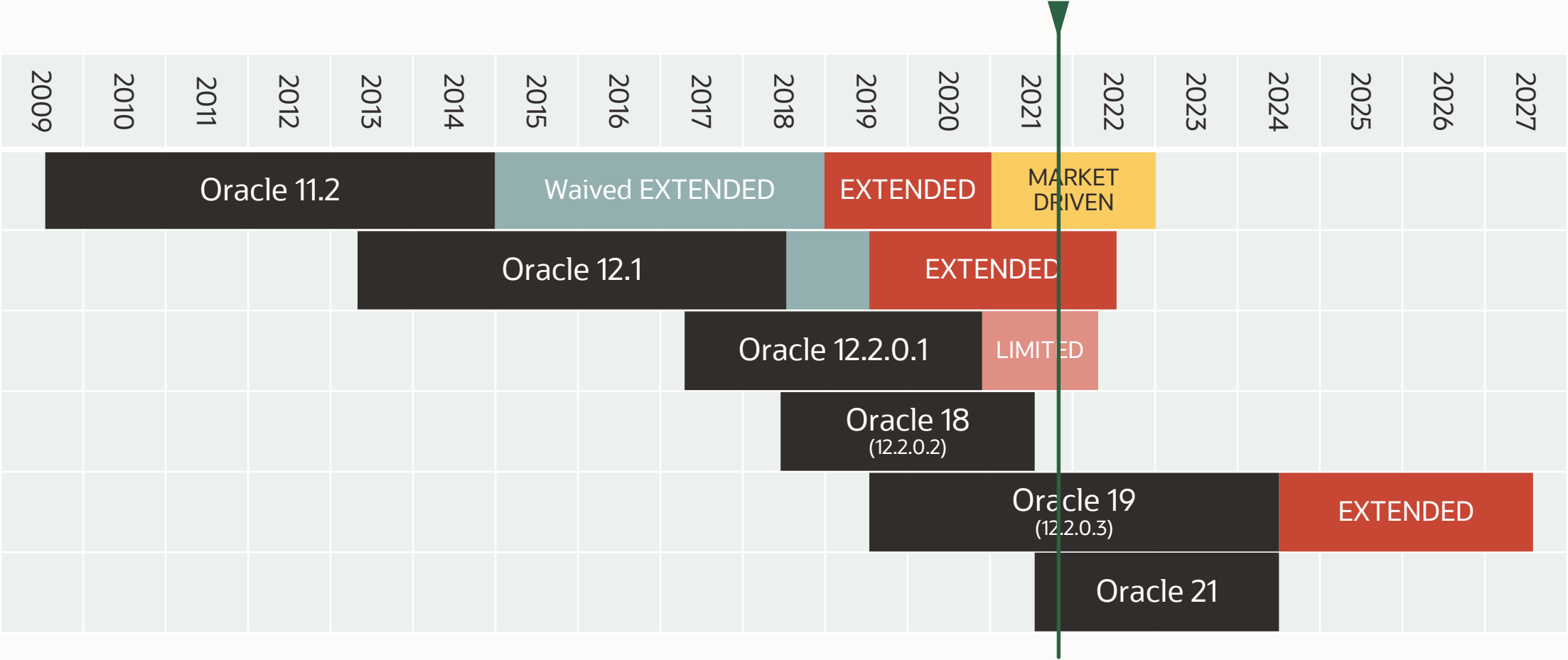
Recommendation: Production environments should go from LTS to LTS

Release Types | **Long Term** vs **Innovation**



Move production databases
from one **Long Term Support** release to the next

Lifetime Support Policy



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



Lifetime Support Policy

Different Support Periods

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

Bug fixing support regardless of severity

Extra cost extension, **10% / 20% extra cost**
Included in ULA/PULA contracts

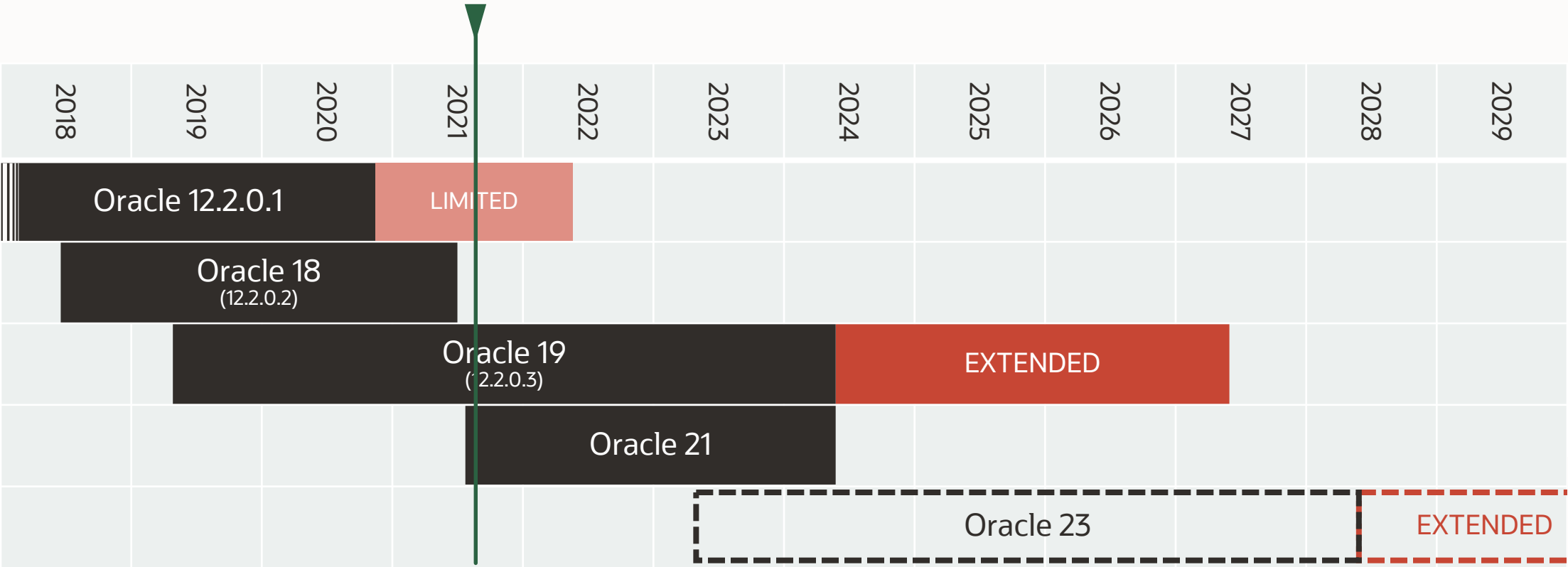
Extended support gets waived to everybody
having a valid Support contract for the product

Extra cost extension after Extended Support
Fixes done only for critical and security issues

Extension for Oracle 12.2.0.1 at **no extra cost**
Only applicable for Sev.1 and security issues

Oracle Support assists as long as the customer
is using the product – but no new fixes will be delivered

Oracle Database 12.2 and beyond



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





Photo by Jose Fontano on Unsplash

Security

The most important reason
to upgrade and patch

Security | Do We Really Need To Say This?

Source: <https://www.nytimes.com/2019/07/22/business/equifax-settlement.html?module=inline>

[nytimes.com](https://www.nytimes.com)

Equifax to Pay at Least \$650 Million in Largest-Ever Data Breach Settlement

By Stacy Cowley

"My database is not facing the internet"



92%

of malware gets
delivered via email



Check | OJVM

Is OJVM installed?

```
Select comp_id, comp_name, version from DBA_REGISTRY order by 1;
```

COMP_ID	COMP_NAME	VERSION
CATALOG	Oracle Database Catalog Views	19.13.0.0.0
CATJAVA	Oracle Database Java Packages	19.13.0.0.0
CATPROC	Oracle Database Packages and Types	19.13.0.0.0
JAVAVM	JServer JAVA Virtual Machine	19.13.0.0.0
OLS	Oracle Label Security	19.13.0.0.0
ORDIM	Oracle Multimedia	19.13.0.0.0
OWM	Oracle Workspace Manager	19.13.0.0.0
XDB	Oracle XML Database	19.13.0.0.0
XML	Oracle XDK	19.13.0.0.0

Oracle Java Virtual Machine | OJVM

OJVM Security Fixes in Quarterly Patches: $\varnothing$ 7.9





Chapter 2

Upgrade to Oracle Database 19c

your key to

Successful Database Upgrades

Step 1

Download and
install **Oracle 19c**

[eDelivery.oracle.com](https://edelivery.oracle.com)

Step 2

Download and
install **newest RU**

MOS Note: 2118136.2

Step 3

Download and use
AutoUpgrade

MOS Note: 2485457.1

Step 4

Performance Stability
with SPM, STA and RAT



your key to

Successful Database Upgrades

Step 1

Download and
Install Oracle 19c

[eDelivery](#) or [CD](#)

Step 2

Download and
install newest RU

[MOS Note 2113136.2](#)

Step 3

Download and use
AutoUpgrade

[MOS Note 249541.1](#)

Step 4

Performance Stability
with SPM, STA and RAT



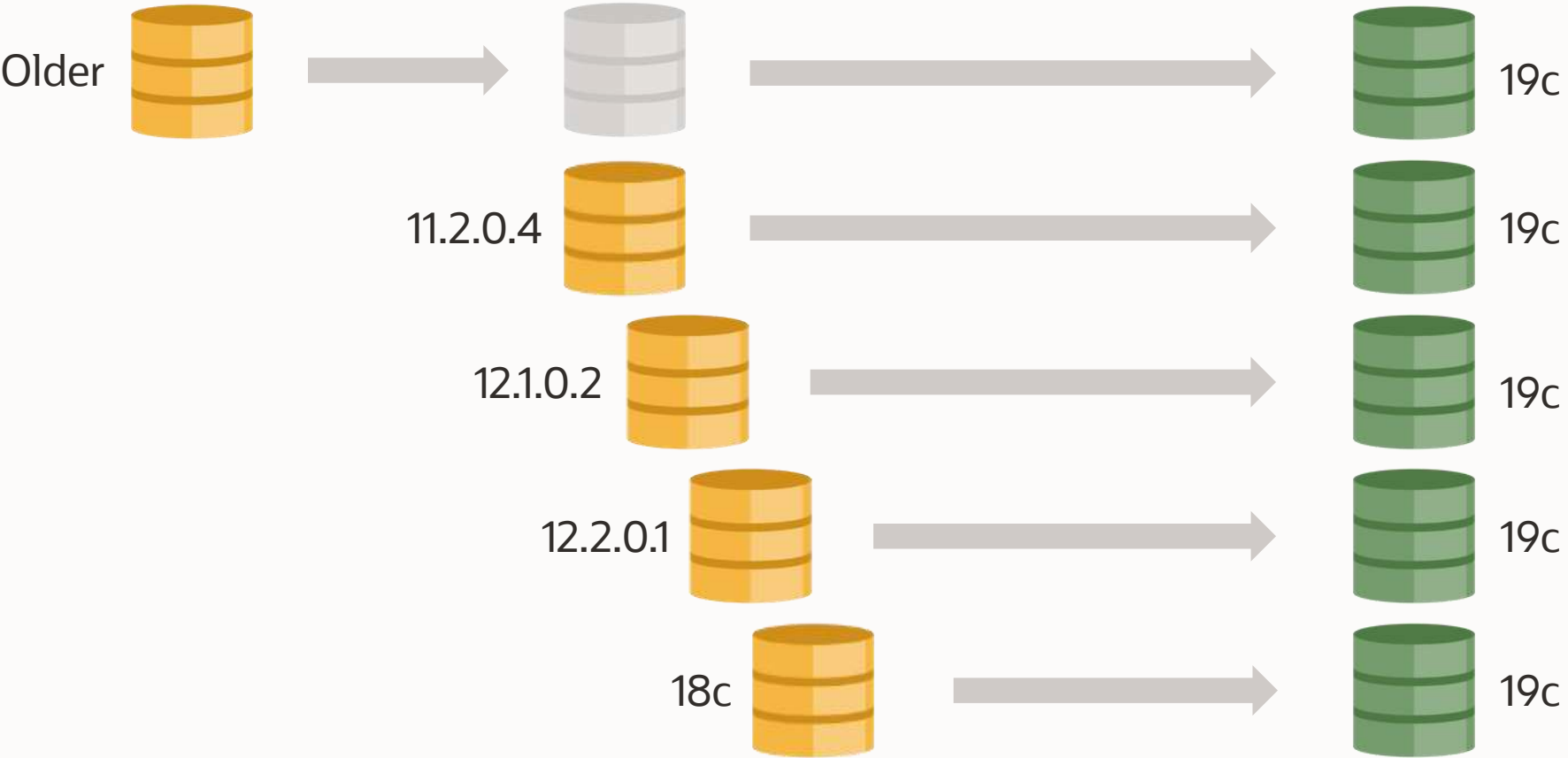


Check

Before Upgrade

Photo by [Diego PH](#) on [Unsplash](#)

Database Upgrade | Supported Releases



Upgrade 19c | **Speed it up**



Save time by gathering statistics in advance

Upgrade 19c | Speed it up

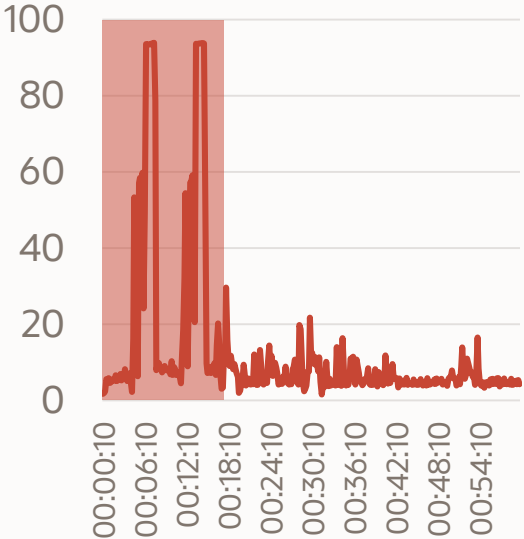
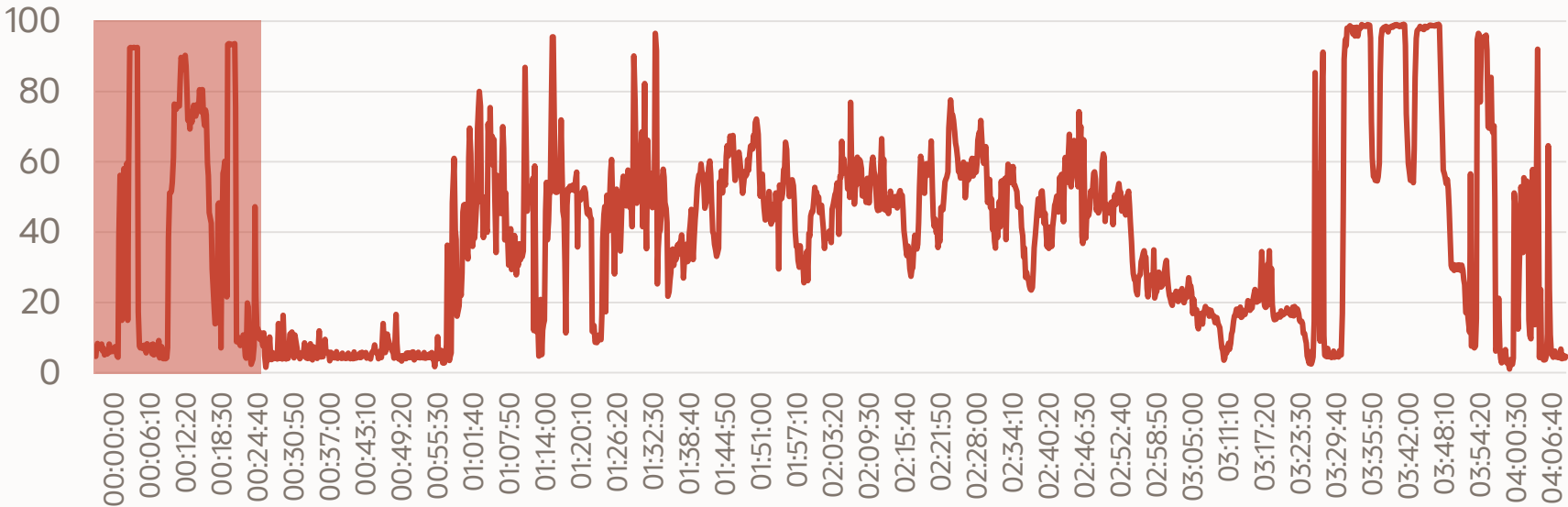
Non-CDB or PDB

```
SQL> begin
        dbms_stats.gather_schema_stats('SYS');
        dbms_stats.gather_schema_stats('SYSTEM');
end;
/
```

Entire CDB

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
-l /tmp \
-b gatherstats -- \
--x"begin dbms_stats.gather_schema_stats('SYS');
dbms_stats.gather_schema_stats('SYSTEM'); end;"
```

Upgrade 19c | Gather Stats In Advance



Gathering stats in advance saves 12 minutes

Health Check | **hcheck.sql**

If your database is highly important, do a health check

- Lightweight, non-intrusive script
- Checks consistency of selected dictionary relationships
- [hcheck.sql - Script to Check for Known Problems \(Doc ID 136697.1\)](#)

```
SQL> @/tmp/hcheck
H.Check Version 4.4 on 01-MAR-2018 23:46:27
-----
Catalog Version 11.2.0.4.0 (1102000400)
db_name: UPR
-----
Procedure Name      Catalog    Fixed
Result             Version    Vs Release    Timestamp
-----
-----
.- LobNotInObj      ... 1102000400 <= *All Rel* 03/01 23:46:27 PASS
.- MissingOIDOnObjCol ... 1102000400 <= *All Rel* 03/01 23:46:27 PASS
.- SourceNotInObj   ... 1102000400 <= *All Rel* 03/01 23:46:27 FAIL
-----
HCKE-0003: SOURCE$ for OBJ# not in OBJ$ (Doc ID 1360233.1)
SOURCE$ has 4 rows for 1 OBJ# values not in OBJ$
.- OversizedFiles   ... 1102000400 <= *All Rel* 03/01 23:46:27 PASS
```



AutoUpgrade

The **ONLY** recommended way to upgrade databases

AutoUpgrade Essentials



AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Always download latest version from MOS

★ AutoUpgrade Tool (Doc ID 2485457.1)

In this Document

[Main Content](#)

[Benefits](#)

[Target Versions Supported](#)

[AutoUpgrade documentation](#)

[References](#)

APPLIES TO:

Oracle Database - Enterprise Edition - Version 12.2.0.1 and later

Oracle Database - Standard Edition - Version 12.2.0.1 and later

Information in this document applies to any platform.

MAIN CONTENT

Description

Oracle Database AutoUpgrade allows DBAs to upgrade one or many databases without human intervention, all with one

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Check your version

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

build.version 21.2.210721
build.hash 680914c
build.date 2021/07/21 11:14:54
build.max_target_version 21
build.supported_target_versions 12.2,18,19,21
build.type production
```

Compare to latest version on MOS

Download

The most recent version of AutoUpgrade can be downloaded via this link: version [20210721.](#)

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

AutoUpgrade handles older releases as well

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

```
build.version 21.2.210721
```

```
build.hash 680914c
```

```
build.date 2021/07/21 11:14:54
```

```
build.max_target_version 21
```

```
build.supported_target_versions 12.2,18,19,21
```

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Shortest possible config file version

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=CDB1
```

Or, generate a sample config file

```
$ java -jar autoupgrade.jar -create_sample_file config  
  
Created sample configuration file /home/oracle/sample_config.cfg
```

Pro tip: *upg1* is a prefix that you decide.
Use it to define multiple databases

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Analyze your database

```
$ java -jar autoupgrade.jar -config CDB1.cfg -mode analyze

...

upg> Job 100 completed

Please check the summary report at:
/u01/app/oracle/cfgtoollogs/autoupgrade/cfgtoollogs/upgrade/auto/status/status.html
/u01/app/oracle/cfgtoollogs/autoupgrade/cfgtoollogs/upgrade/auto/status/status.log
```

Pro tip: Analyze is similar to running `preupgrade.jar`

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Summary report - text

```
=====
                        Autoupgrade Summary Report
=====
[Date]                  Tue Jan 12 10:26:19 CET 2021
[Number of Jobs] 1
=====
[Job ID] 100
=====
[DB Name]                CDB1
[Version Before Upgrade] 12.2.0.1.0
[Version After Upgrade]  19.9.0.0.0
-----
[Stage Name]    PRECHECKS
[Status]        SUCCESS
[Start Time]    2021-01-12 10:25:58
[Duration]      0:00:20
[Log Directory] /u01/app/oracle/upg/CDB1/100/prechecks
[Detail]        /u01/app/oracle/upg/CDB1/100/prechecks/cdb1_preupgrade.log
                Precheck passed and no manual intervention needed
-----
```

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Summary report - HTML

ORACLE®					
Date: Tue Jan 12 10:26:19 CET 2021					
Number of Jobs: 1					
Job ID: 100					
DB Name: CDB1					
DB Version Before Upgrade: 12.2.0.1.0					
DB Version After Upgrade: 19.0.0.0.0					
Autoupgrade Stage List					
Stage Name	Status	Start Time	Duration	Log Directory	Detail
PRECHECKS	SUCCESS	2021-01-12 10:25:58	0:00:20	/u01/app/oracle/cfgtoollogs/autoupgrade/CDB1/CDB1/100/prechecks	Prechecks Report Precheck passed and no manual intervention needed

AutoUpgrade | Essentials

Download
Configure
Analyze
Check
Upgrade

CDB1

DATABASE

DB Compatible	12.2.0
DB Version	12.2.0.1.0
Operating System	Linux
Blocksize	8192
Timezone	26
LogMode	ARCHIVELOG
Readonly	false
Edition	EE

COMPONENTS

Oracle Component	Version	Upgrade Action	Current Status
Oracle Workspace Manager	12.2.0.1.0	to be upgraded	VALID
Oracle Catalog Views	12.2.0.1.0	to be upgraded	VALID
Real Application Clusters	12.2.0.1.0	to be upgraded	OPTION OFF
Oracle XML Database	12.2.0.1.0	to be upgraded	VALID
Oracle Label Security	12.2.0.1.0	to be upgraded	VALID
Oracle Packages and Types	12.2.0.1.0	to be upgraded	VALID

Containers

CDB\$ROOT

PreChecks Recommend(3)
PreChecks Info(4)
PostChecks Warning(3)
PostChecks Recommend(3)

PDB\$SEED

PreChecks Recommend(3)
PreChecks Info(1)
PostChecks Warning(3)
PostChecks Recommend(3)

PDB1

PreChecks Warning(2)
PreChecks Recommend(3)
PreChecks Info(1)
PostChecks Warning(4)
PostChecks Recommend(3)

PDB2

PreChecks Recommend(3)
PreChecks Info(1)
PostChecks Warning(3)

CDB\$ROOT

CheckName: DICTIONARY_STATS FixUp Available: YES Severity: RECOMMEND Stage: PRECHECKS

Gather stale data dictionary statistics prior to database upgrade in off-peak time using:

EXECUTE DBMS_STATS.GATHER_DICTIONARY_STATS;

Dictionary statistics help the Oracle optimizer find efficient SQL execution plans and are essential for proper upgrade timing. Oracle recommends gathering dictionary statistics in the last 24 hours before database upgrade.

For information on managing optimizer statistics, refer to the 12.2.0.1 Oracle Database SQL Tuning Guide.

Dictionary statistics do not exist or are stale (not up-to-date).

CheckName: HIDDEN_PARAMS FixUp Available: NO Severity: RECOMMEND Stage: PRECHECKS

Review and remove any unnecessary HIDDEN/UNDERSCORE parameters.

Remove hidden parameters before database upgrade unless your application vendors and/or Oracle Support state differently. Changes will need to be made in the pfile/spfile.

The database contains the following initialization parameters whose name begins with an underscore:

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Preupgrade report comes in:

- HTML
- Text
- JSON

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Upgrade

```
$ java -jar autoupgrade.jar -config CDB1.cfg -mode deploy
```



Have a cup of coffee and wait, or ...

AutoUpgrade | Essentials

- Download
- Configure
- Analyze
- Check

Upgrade

Monitor

```
upg> lsj
```

Job#	DB_NAME	STAGE	OPERATION	STATUS	START_TIME	UPDATED	MESSAGE
101	CDB1	PREFIXUPS	EXECUTING	RUNNING	20/11/24 13:38	13:39:26	Remaining 12/13

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

All the details

```
upg> status -job 101

Progress
-----
Start time:      20/11/24 13:38
Elapsed (min):   13
Last update:     2020-11-24T13:48:52.139
Stage:           DBUPGRADE
Operation:       EXECUTING
Status:          RUNNING
Stage summary:
  SETUP          <1 min
  GRP             <1 min
  PREUPGRADE     <1 min
  PRECHECKS      <1 min
  PREFIXUPS      8 min
  DRAIN          <1 min
  DBUPGRADE      3 min (IN PROGRESS)

Job Logs Locations
-----
Logs Base:       /home/oracle/autoupg_default/CDB1/CDB1
Job logs:        /home/oracle/autoupg_default/CDB1/CDB1/101
Stage logs:      /home/oracle/autoupg_default/CDB1/CDB1/101/dbupgrade
TimeZone:        /home/oracle/autoupg_default/CDB1/CDB1/temp
```

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

All the details - continued

...

Additional information

Details:

[Upgrading] is [0%] completed for [cdb1-cdb\$root]

CONTAINER	PERCENTAGE
CDB\$ROOT	UPGRADE [12%]
PDB\$SEED	UPGRADE PENDING
PDB3	UPGRADE PENDING

Error Details:

None

AutoUpgrade | Essentials

Download

Configure

Analyze

Check

Upgrade

Success

```
upg> Job 101 completed
----- Final Summary -----
Number of databases          [ 1 ]

Jobs finished successfully    [1]
Jobs failed                   [0]
Jobs pending                  [0]
----- JOBS FINISHED SUCCESSFULLY -----
Job 101 for CDB1

---- Drop GRP at your convenience once you consider it is no longer needed ----
Drop GRP from CDB1: drop restore point AUTOUPGRADE_9212_CDB1122010
```

And it includes:

- Recompilation (utlrb.sql)
- Time zone file upgrade
- Postupgrade fixups
- ... and so much more

AutoUpgrade | Essentials

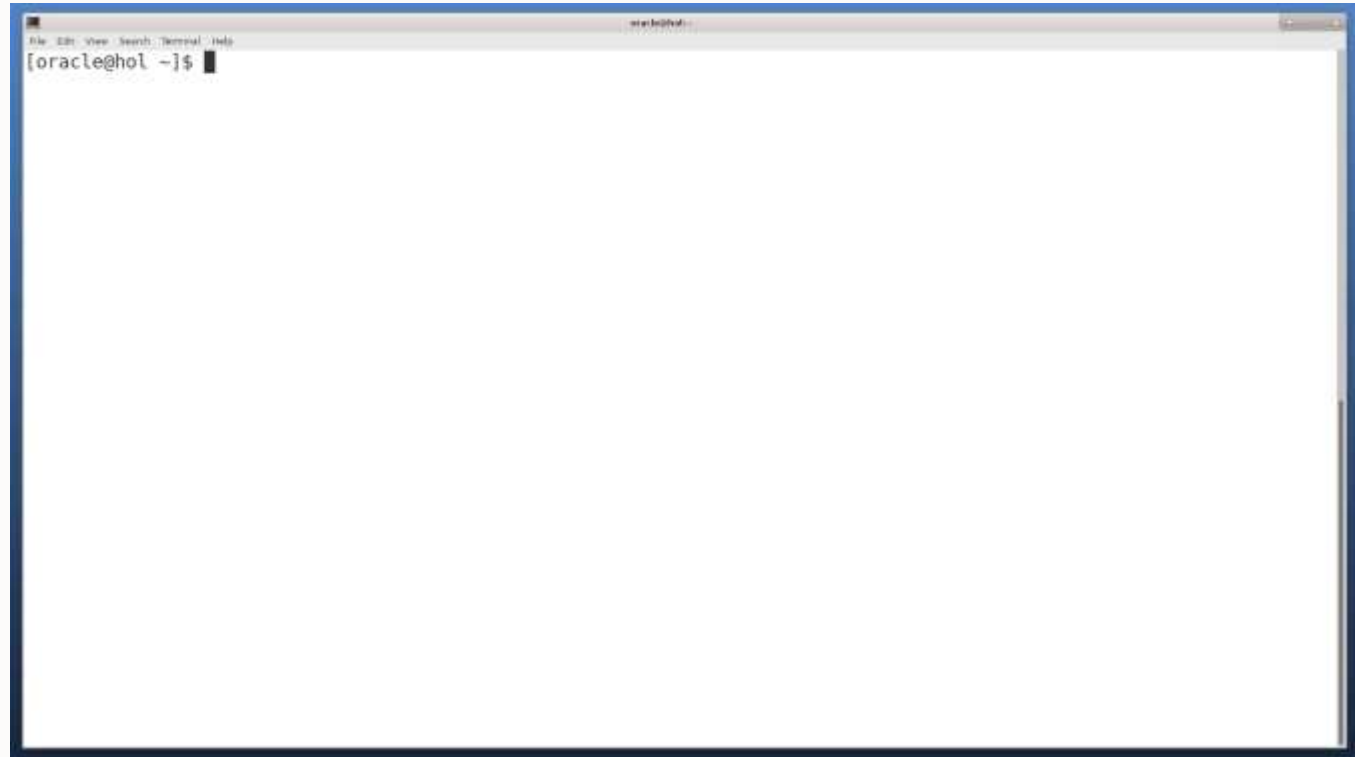
Download

Configure

Analyze

Check

Upgrade



[Watch on YouTube](#)

AutoUpgrade | Essentials

One-liner using environment variables

```
export ORACLE_SID=CDB1
export ORACLE_HOME=/u01/app/oracle/12.2
export ORACLE_TARGET_HOME=/u01/app/oracle/19

java -jar autoupgrade.jar -config_values -mode analyze
```

One-liner using config_values

```
java -jar autoupgrade.jar \
    -config_values
"sid=CDB1,source_home=/u01/app/oracle/12.2,target_home=/u01/app/oracle/19" \
    -mode analyze
```

AutoUpgrade Advanced Options



Photo by Ciprian Boiciuc on Unsplash

AutoUpgrade | Advanced Options

Many Databases

Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Upgrade one or many databases

One

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=CDB1
```

Many

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=CDB1
```

```
upg2.source_home=/u01/app/oracle/product/11.2.0.4  
upg2.target_home=/u01/app/oracle/product/19  
upg2.sid=DB11204
```

...

```
upgn.source_home=/u01/app/oracle/product/12.1.0.2  
upgn.target_home=/u01/app/oracle/product/19  
upgn.sid=HR
```

Pro tip: You can also start multiple instances of AutoUpgrade at the same time

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

Recompilation

Time Zone

Parallel

Monitoring

Upgrade only when `upgrade_node` matches hostname

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.upgrade_node=test_server01.mycorp.net
upg1.sid=CDB1

upg2.source_home=/u01/app/oracle/product/12.2.0.1
upg2.target_home=/u01/app/oracle/product/19
upg2.upgrade_node=prod_server01.mycorp.net
upg2.sid=CDB2
```

- Database `upg1` will only be upgraded when AutoUpgrade gets executed on server `test_server01.mycorp.net`

AutoUpgrade | Advanced Options

Many Databases
Different Servers

PFILE

Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Update initialization parameters as part of the upgrade

You can:

- Add or remove parameters
- Before, during or after upgrade
- For a single or every database

AutoUpgrade | Advanced Options

Many Databases
Different Servers

PFILE

Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Remove a parameter during a specific upgrade

```
upg1.del_during_upgrade_pfile=/home/oracle/global_del_during.ora
```

Example: global_del_during.ora
optimizer_features_enable

Add parameters to all databases after upgrade

```
global.add_after_upgrade_pfile=/home/oracle/global_add_after.ora
```

Example: global_add_after.ora
deferred_segment_creation=false
_cursor_obsolete_threshold=1024
_sql_plan_directive_mgmt_control=0
_use_single_log_writer=true

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

Recompilation

Time Zone

Parallel

Monitoring

Batch-update parameters

```
global.del_during_upgrade_pfile=/home/oracle/global_del_during.ora
global.add_during_upgrade_pfile=/home/oracle/global_add_during.ora
global.del_after_upgrade_pfile=/home/oracle/global_del_during.ora
global.add_after_upgrade_pfile=/home/oracle/global_add_after.ora
```

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
```

```
upg1.add_after_upgrade_pfile=/home/oracle/upg1_add_after.ora
```

```
upg2.source_home=/u01/app/oracle/product/12.2.0.1
upg2.target_home=/u01/app/oracle/product/19
upg2.sid=CDB2
```

```
upg2.add_after_upgrade_pfile=/home/oracle/upg2_add_after.ora
```

Example: **global_add_after.ora**

```
deferred_segment_creation=false
_cursor_obsolete_threshold=1024
_sql_plan_directive_mgmt_control=0
_use_single_log_writer=true
```

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE

Shell Scripts

Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Execute your own scripts as part of the upgrade

You can:

- Before and after upgrade
- Halt or continue on error
- For a single or every database

Ideas:

- Enterprise Manager configuration
- Backup configuration
- Interact with apps using the database

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE

Shell Scripts

Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Shell script execution

```
global.before_action=/database/scripts/set_blackout.sh  
  
upg1.source_home=/u01/app/oracle/product/12.2.0.1  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=CDB1  
upg1.after_action=/database/scripts/start_level0.sh
```

- Permitted extension options:
 - Unix shell (.sh)
 - Microsoft Windows batch (.bat, .cmd)
 - Microsoft Windows PowerShell (.ps1)

Pro tip: If you want script execution for all upgrades use `global.before_action` and `global.after_action`



AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE

Shell Scripts

Restore Point
Underscores
Recompilation
Time Zone
Parallel
Monitoring

Shell script execution

Default - AutoUpgrade **does not react** on return code

```
upg1.before_action=/database/scripts/run_this_on_UPG1_before.sh
```

Optionally - AutoUpgrade **halts** on non-zero return code

```
upg1.before_action=/database/scripts/run_this_on_UPG1_before.sh Y
```

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

Recompilation

Time Zone

Parallel

Monitoring

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=no
```

```
upg2.source_home=/u01/app/oracle/product/12.2.0.1  
upg2.target_home=/u01/app/oracle/product/19  
upg2.sid=CDB2  
upg2.drop_grp_after_upgrade=yes
```

- 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

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

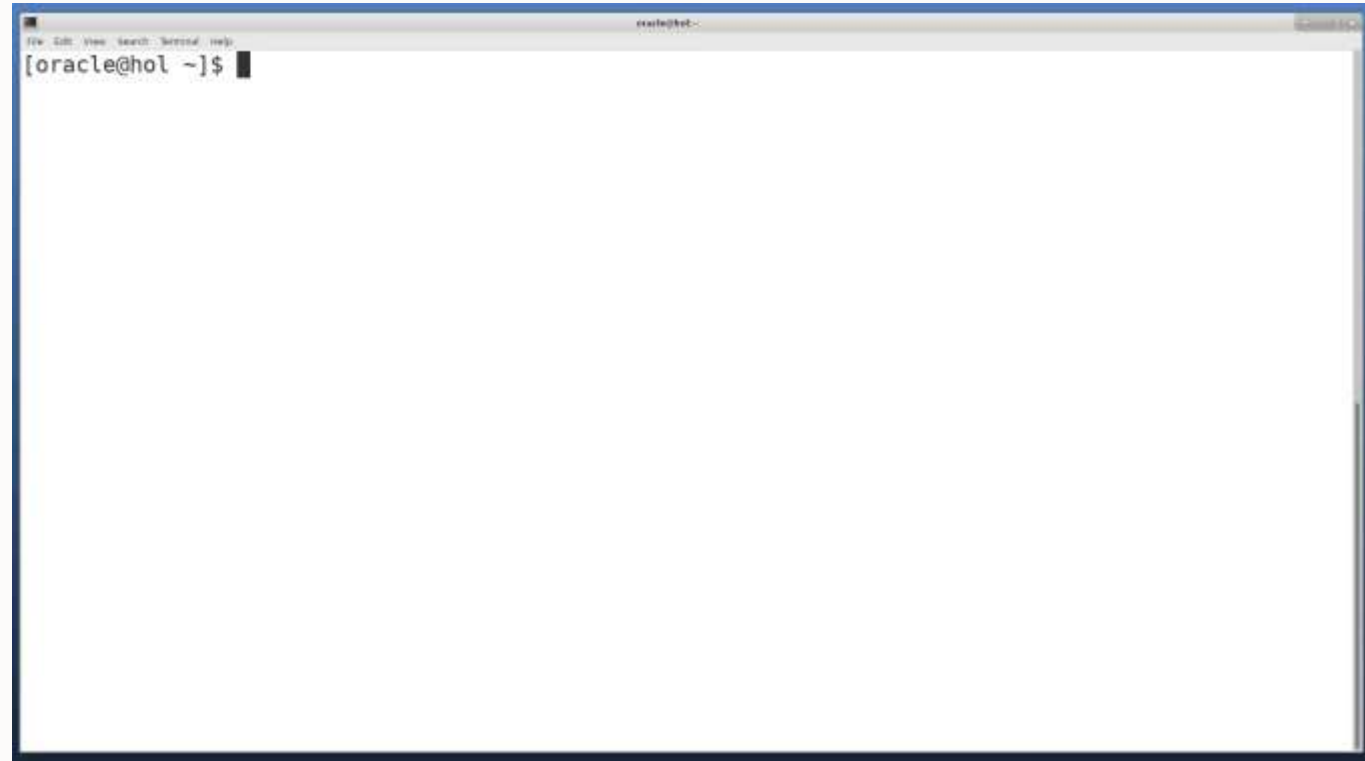
Underscores

Recompilation

Time Zone

Parallel

Monitoring



AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

Recompilation

Time Zone

Parallel

Monitoring

Underscore parameters and events

```
upg1.source_home=/u01/app/oracle/product/12.2.0.1  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=CDB1  
upg1.remove_underscore_parameters=yes
```

- Default behavior:
 - Underscores and events will be kept

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

Recompilation

Time Zone

Parallel

Monitoring

Postpone recompilation

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

- Default behavior:
 - Recompilation happens after the upgrade

AutoUpgrade | Advanced Options

Many Databases

Different Servers

PFILE

Shell Scripts

Restore Point

Underscores

Recompilation

Time Zone

Parallel

Monitoring

Skip 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.timezone_upg=no
```

- Default behavior:
 - Time zone adjustment happens post upgrade
 - Database will be restarted several times
 - Important when you use "Downgrade" as fallback strategy as time zone can't be downgraded

AutoUpgrade | Advanced Options

- Many Databases
- Different Servers
- PFILE
- Shell Scripts
- Restore Point
- Underscores
- Recompilation
- Time Zone
- Parallel**
- Monitoring

CDB

```
upg1.catctl_options=-n 64 -N 8
```

-n	Total number of parallel processes (min 4, max unlimited, default CPU_COUNT)
-N	Number of parallel processes per PDB (min 1, max 8, default 2)

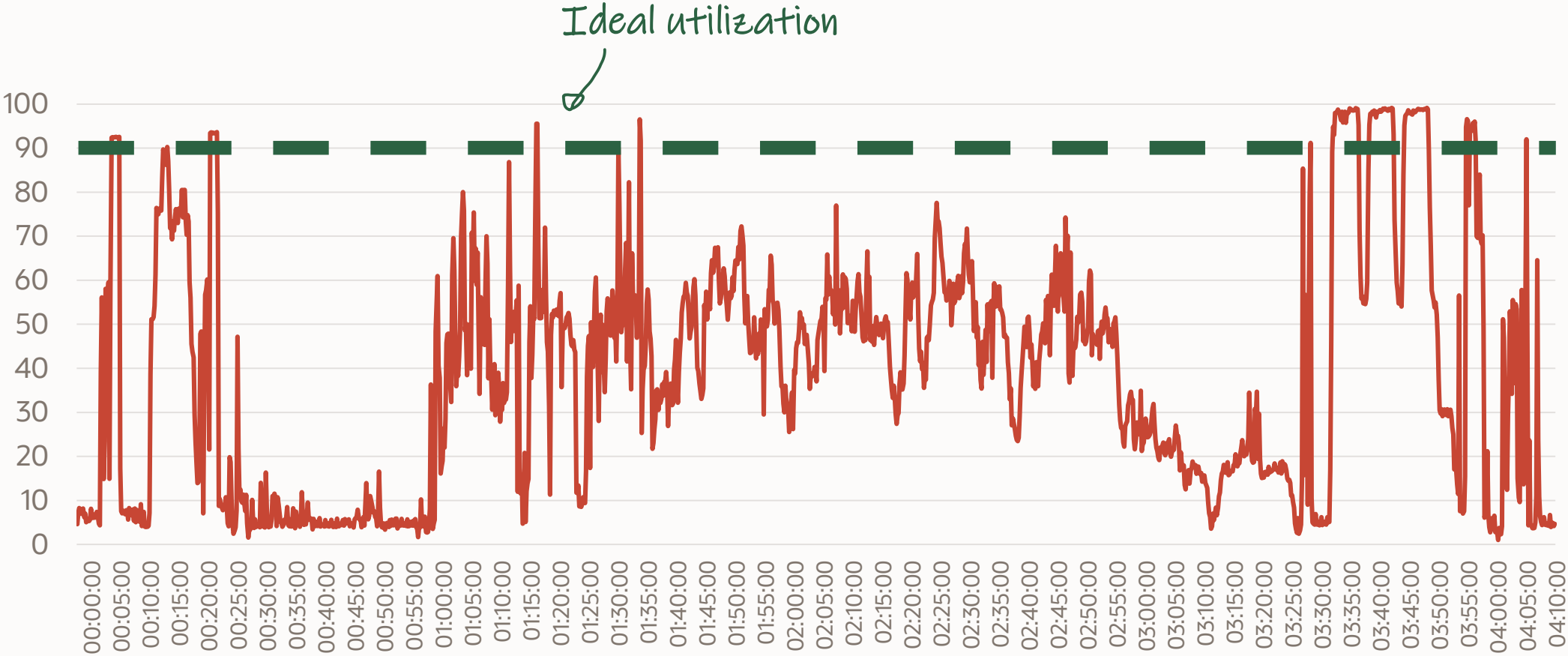
Concurrent PDB upgrades: n / N



AutoUpgrade | Parallel

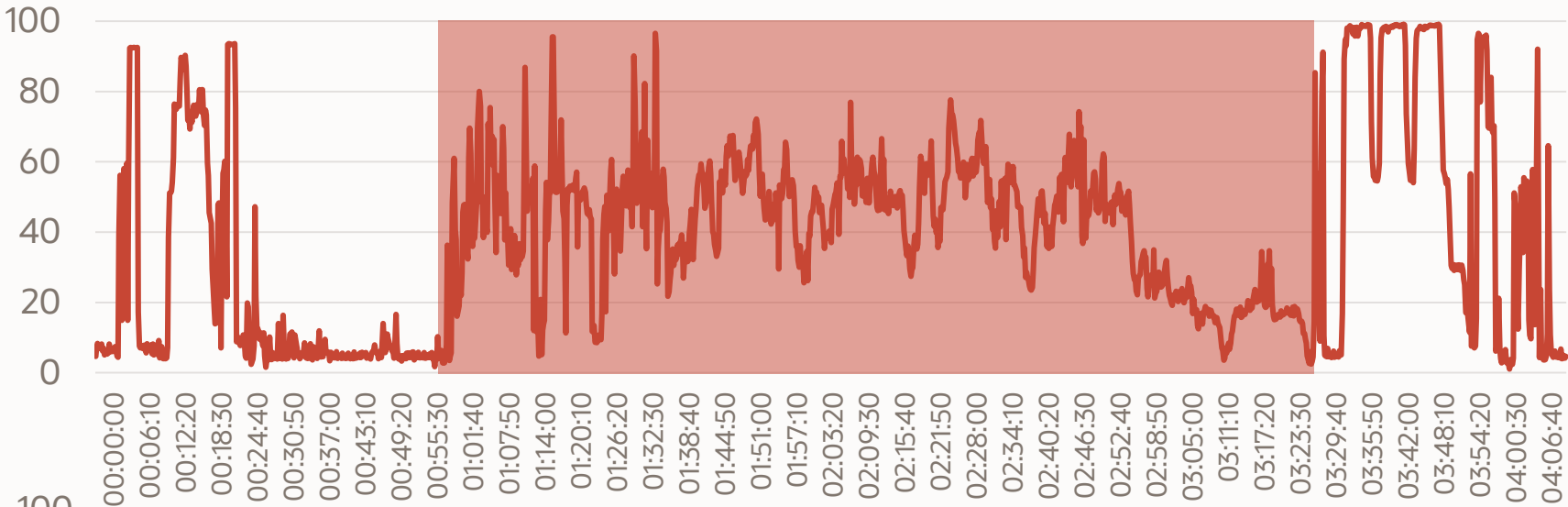
- OCI Bare Metal host
 - 16 OPCUs
 - 768 GB memory
 - NVMe disks
- CDB with 52 PDBs
 - CPU_COUNT = 32
 - SGA_TARGET = 80G
 - PGA_AGGREGATE_TARGET = 20G
- Many database components (17 in total)
- **Upgrade from 12.1.0.2 to 19**

AutoUpgrade | Parallel

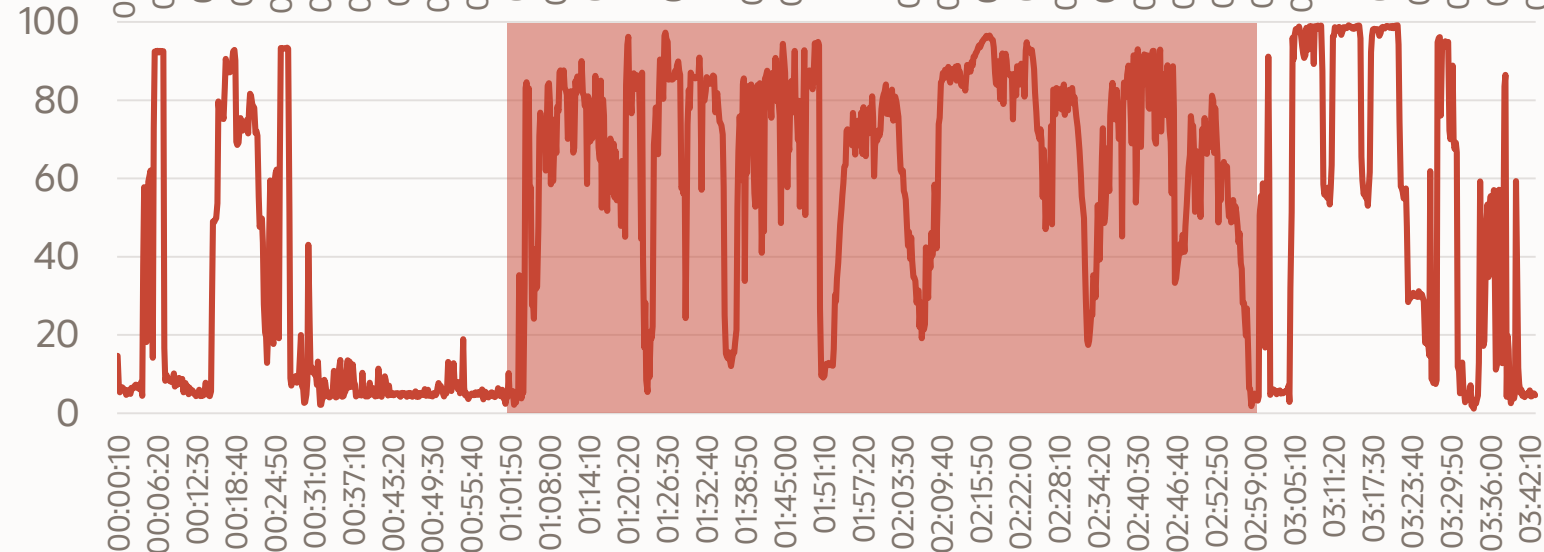


Total upgrade time: 4 hours 8 minutes

AutoUpgrade | Parallel



32 parallel processes



54 parallel processes
upg1.catctl_options=-n 54

26 minutes faster

AutoUpgrade | Advanced Options

Many Databases
Different Servers
PFILE
Shell Scripts
Restore Point
Underscores
Recompilation
Time Zone
Parallel

Monitoring

Open your browser:

- `http://<log_dir>/cfgtoollogs/upgrade/auto/state.html`



The screenshot shows the Oracle AutoUpgrade status page. At the top, the Oracle logo is displayed, followed by the date and time 'Date: Wed Jul 15 16:15:56 CEST 2020 | Operating System: Linux'. Below this is a red header bar with the text 'Current Upgrade Status'. The main content is a table with the following columns: JobId, DbName, Stage, Operation, Status, and Details. There are two rows of data. The first row shows JobId 102, DbName FTEX, Stage DBUPGRADE, Operation EXECUTING, and Status RUNNING. The Details column for this row shows '[Upgrading] is [88%] completed for [ftex]' followed by a table with columns 'CONTAINER' and 'PERCENTAGE', and a row 'FTEX|UPGRADE [88%]'. The second row shows JobId 103, DbName DB12, Stage DBUPGRADE, Operation EXECUTING, and Status RUNNING. The Details column for this row shows '[Upgrading] is [49%] completed for [db12]' followed by a table with columns 'CONTAINER' and 'PERCENTAGE', and a row 'DB12|UPGRADE [49%]'. A small upward arrow is visible on the right side of the table.

JobId	DbName	Stage	Operation	Status	Details
102	FTEX	DBUPGRADE	EXECUTING	RUNNING	[Upgrading] is [88%] completed for [ftex] +-----+-----+ CONTAINER PERCENTAGE +-----+-----+ FTEX UPGRADE [88%] +-----+-----+
103	DB12	DBUPGRADE	EXECUTING	RUNNING	[Upgrading] is [49%] completed for [db12] +-----+-----+ CONTAINER PERCENTAGE +-----+-----+ DB12 UPGRADE [49%] +-----+-----+

- AutoUpgrade: Refresh Status Information Automatically

Case by Case





Photo by Hello I'm Nik  on Unsplash

AutoUpgrade to a New Server

Upgrade to a new server | Overview

Source System

autoupgrade.jar

-analyze

-fixups

Target System

autoupgrade.jar

-upgrade

```
$ java -jar autoupgrade.jar -mode analyze ...
```

```
$ java -jar autoupgrade.jar -mode fixups ...
```

```
$ java -jar autoupgrade.jar -mode upgrade ...
```

Upgrade to a new server | Details

Source Server

```
upg1.source_home=/u01/app/oracle/product/12  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=DB12
```

- -mode analyze
- -mode fixups
- shutdown immediate

Copy database including redo logs, control files, SPFILE, password file

Target Server

- Update /etc/oratab
- Prepare ?/network/admin files

- STARTUP UPGRADE

```
upg1.source_home=/tmp  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=DB12
```

- -mode upgrade

Pro tip: Find more details in blog post
[Oracle AutoUpgrade between two servers](#)



Photo by [Danilo Alves](#) on [Unsplash](#)

Unplug – Plug – Upgrade ... and even more

Unplug-Plug | Refreshable Clone PDB



COMING SOON!

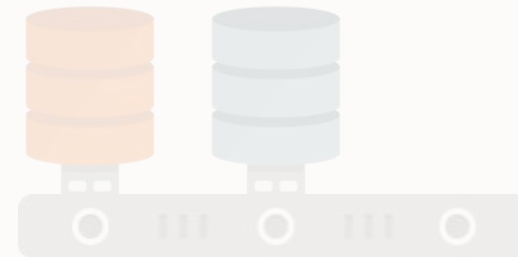
Upgrade via refreshable clone PDB

Unplug-Plug | Refreshable Clone PDB

Clone User

```
CREATE USER c##borg  
IDENTIFIED BY oracle  
DEFAULT TABLESPACE users  
TEMPORARY TABLESPACE temp  
CONTAINER=ALL;
```

```
GRANT  
CREATE SESSION,  
CREATE PLUGGABLE DATABASE,  
SELECT_ANY_CATALOG  
TO c##borg  
CONTAINER = ALL;
```

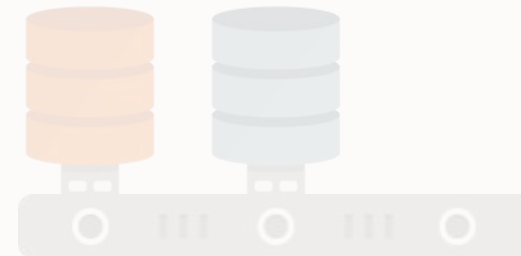


Unplug-Plug | Refreshable Clone PDB

Database link into source PDB



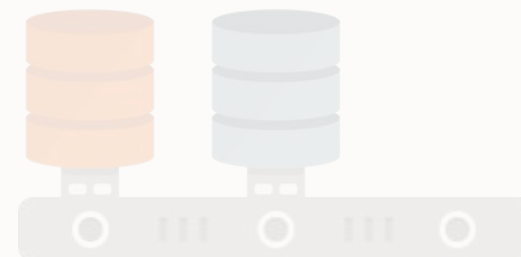
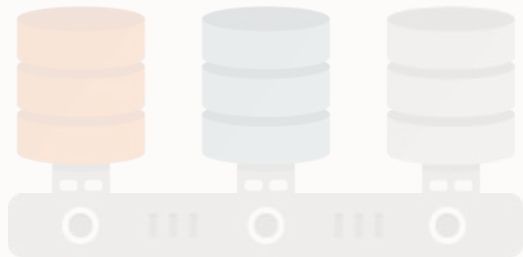
```
CREATE DATABASE LINK clonePDB1  
CONNECT TO c##borg  
IDENTIFIED BY oracle  
USING 'tns-or-ezconnect';
```



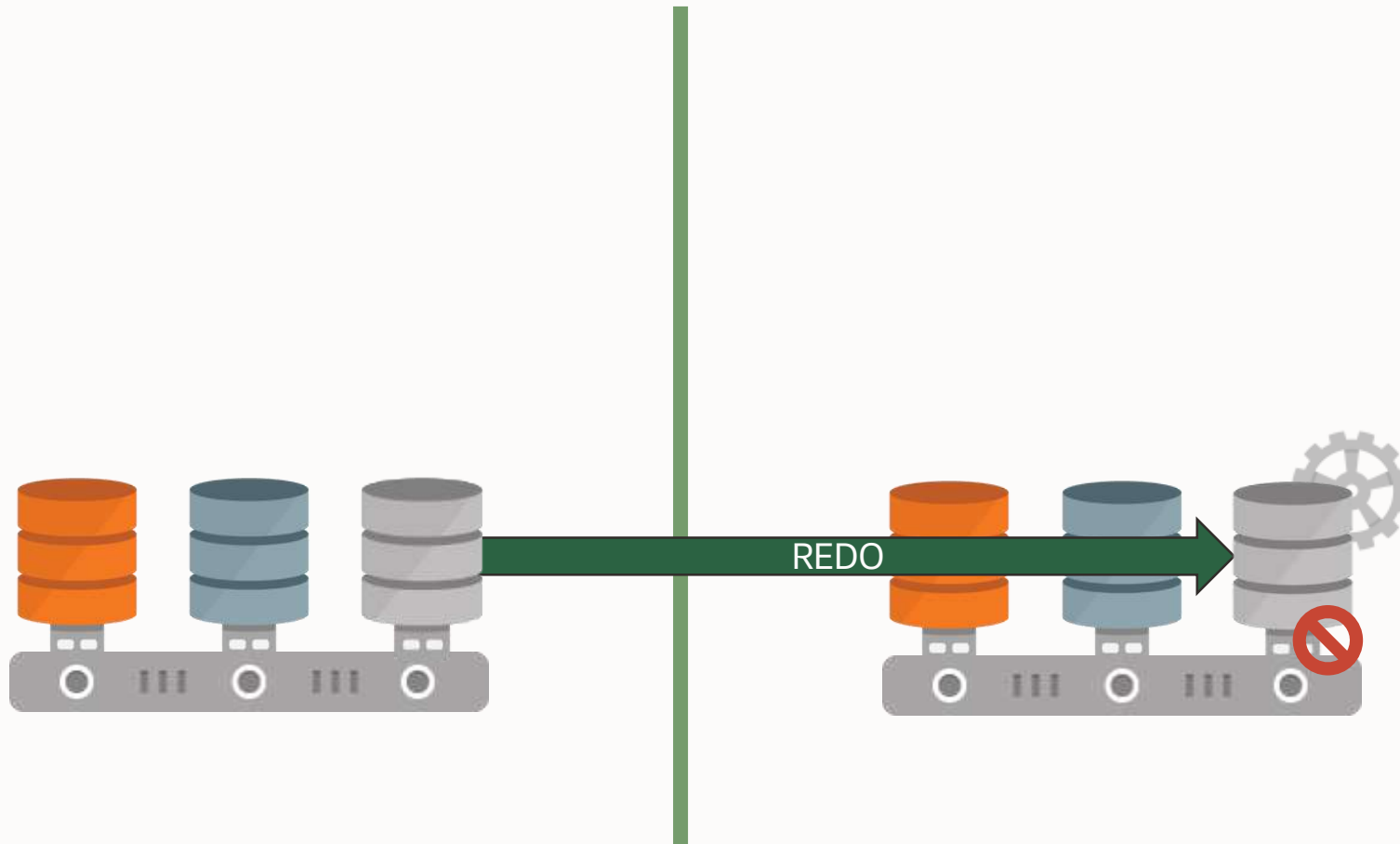
Unplug-Plug | Refreshable Clone PDB

Fully automated relocation with upgrade

```
upg1.source_home=/u01/app/oracle/product/12.2  
upg1.target_home=/u01/app/oracle/product/19  
upg1.sid=CDB1  
upg1.pdbs=PDB1  
upg1.target_cdb=CDB2  
upg1.source_dblink.PDB1=clonePDB1 600
```



Unplug-Plug | Refreshable Clone PDB



Unplug-Plug | Refreshable Clone PDB



AutoUpgrade uses
`CREATE PLUGGABLE DATABASE` statement
which automatically adjusts parallel degree



Photo by Philipp Katzenberger on Unsplash

Grid Infrastructure, Upgrade and Patching

AutoUpgrade | RAC

How to upgrade a RAC database

1. Upgrade Grid Infrastructure
 - Not covered by AutoUpgrade
 - Recommended to upgrade in advance
2. Upgrade Database
 - Upgrade with AutoUpgrade
 - Everything handled by AutoUpgrade

Recommendation: Keep GI and database patch level in sync



Photo by [Chris Briggs](#) on [Unsplash](#)

What if ...

Tips and Tricks and Workarounds

AutoUpgrade | What if ... AutoUpgrade fails

1. Create zip file

```
$ java -jar autoupgrade.jar -config config.cfg -zip
```

2. Optionally, add opatch lsinventory

```
$ $ORACLE_HOME/OPatch/opatch lsinventory > opatch.txt  
$ zip -r AUPG_210419_0735_461.zip opatch.txt
```

3. Upload it to My Oracle Support





*"We upgraded 735 databases to 19c,
and the task was mostly relatively
relaxed.*

*Start the AutoUpgrade tool and
monitor the progress from time to
time.*

*Sitting in front of the screen the
whole time is not necessary."*

Alain Fuhrer

(Former) Head IT Database Services

La Mobilière

Bern, Switzerland

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

