ORACLE

AutoUpgrade Deep Dive

UKOUG Spring Tech Summit 2021

Daniel Overby Hansen
Senior Principal Product Manager

Mike Dietrich
Distinguished Product Manager

Photo by Karl Moran on Unsplash

Mike Dietrich

Distinguished Product Manager

Database Upgrade
and Migrations

https://MikeDietrichDE.com

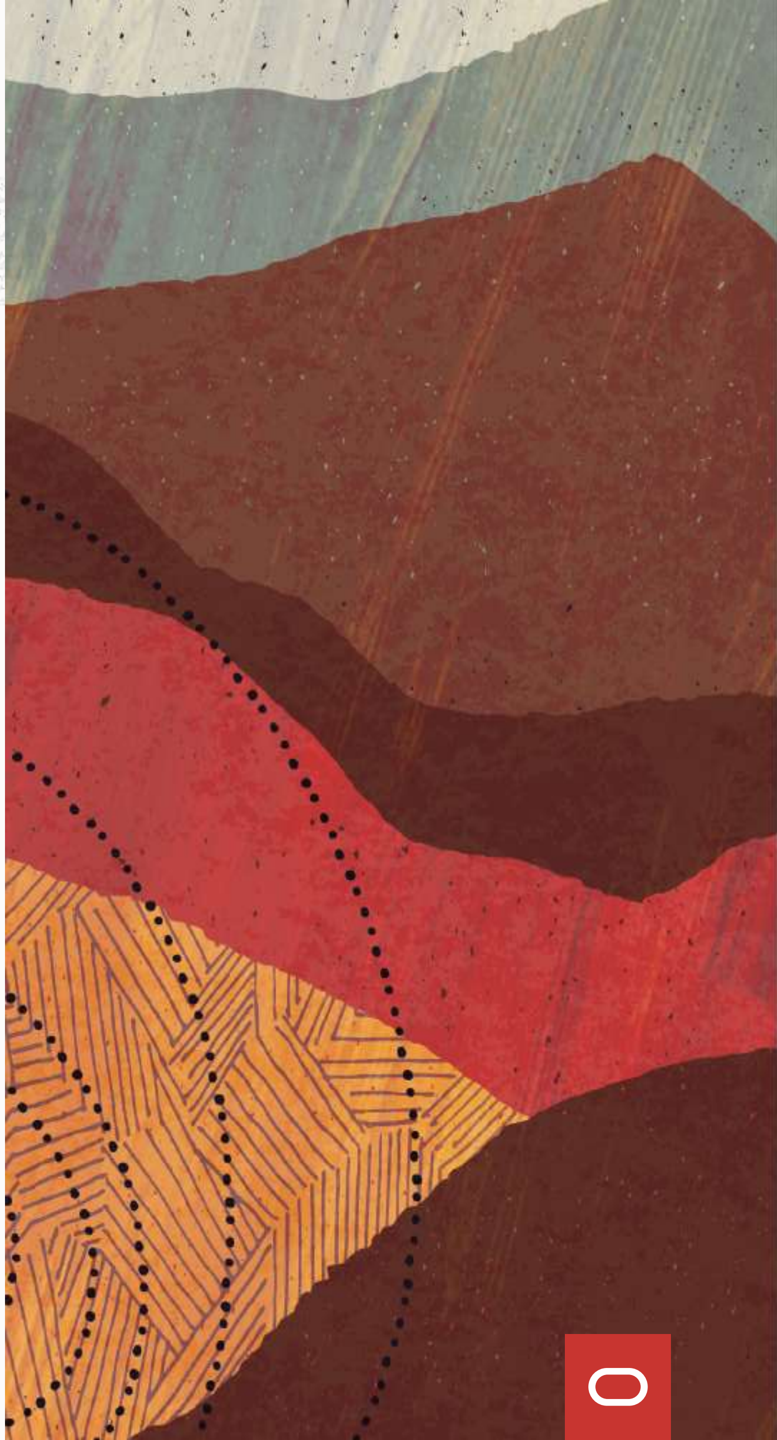
@MikeDietrichDE

mikedietrich

Daniel Overby Hansen

Senior Principal Product Manager
Cloud Migration

https://dohdatabase.com
@dohdatabase
dohdatabase



AutoUpgrade Essentials

Photo by Daniel Cheung on Unsplash

AutoUpgrade | Essentials

Download

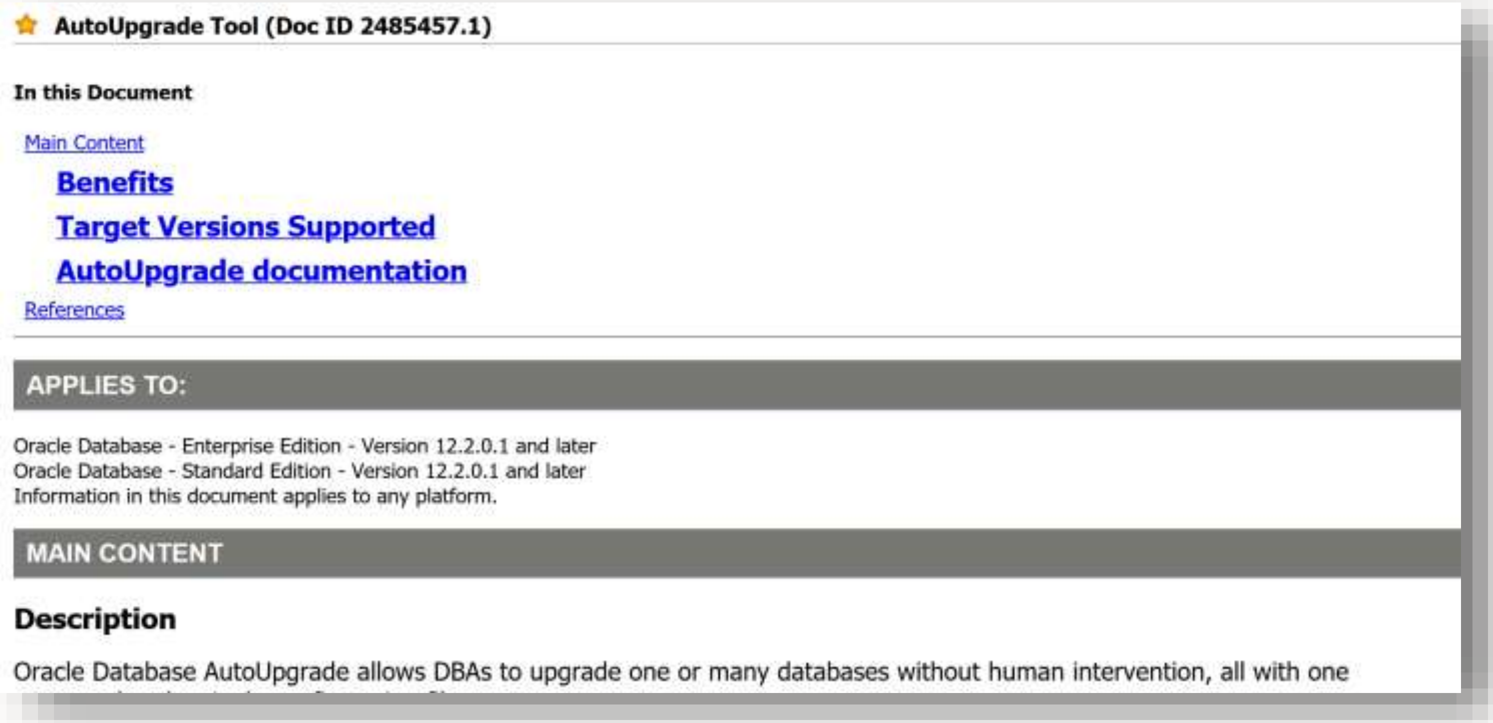
Configure

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

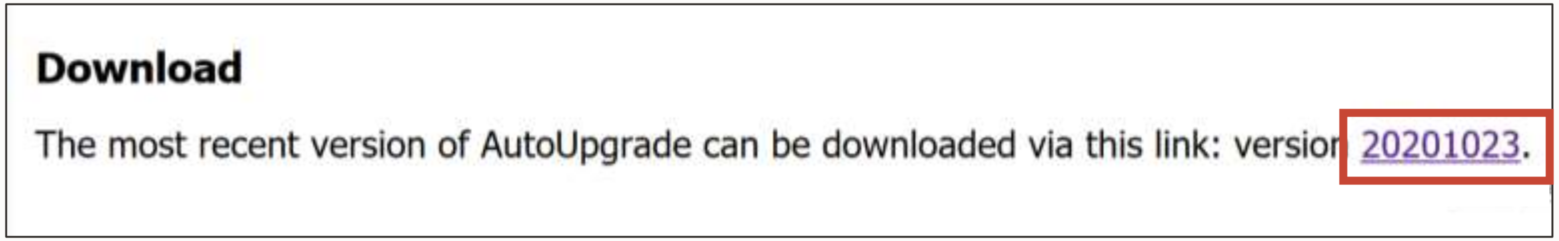
Download
Configure
Analyze
Check
Upgrade

Check your version

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

build.hash 04dd9f2
build.version 19.7.5
build.date 2020/02/11 15:28:49
build.max_target_version 19
build.type production
```

Compare to latest version on MOS



Download

The most recent version of AutoUpgrade can be downloaded via this link: version 20201023.

AutoUpgrade | Essentials

- **Download**
- Configure
- Analyze
- Check
- Upgrade

AutoUpgrade handles older releases as well

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

build.hash 8ee6880
build.version 21.1.1
build.date 2020/12/14 14:41:34
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

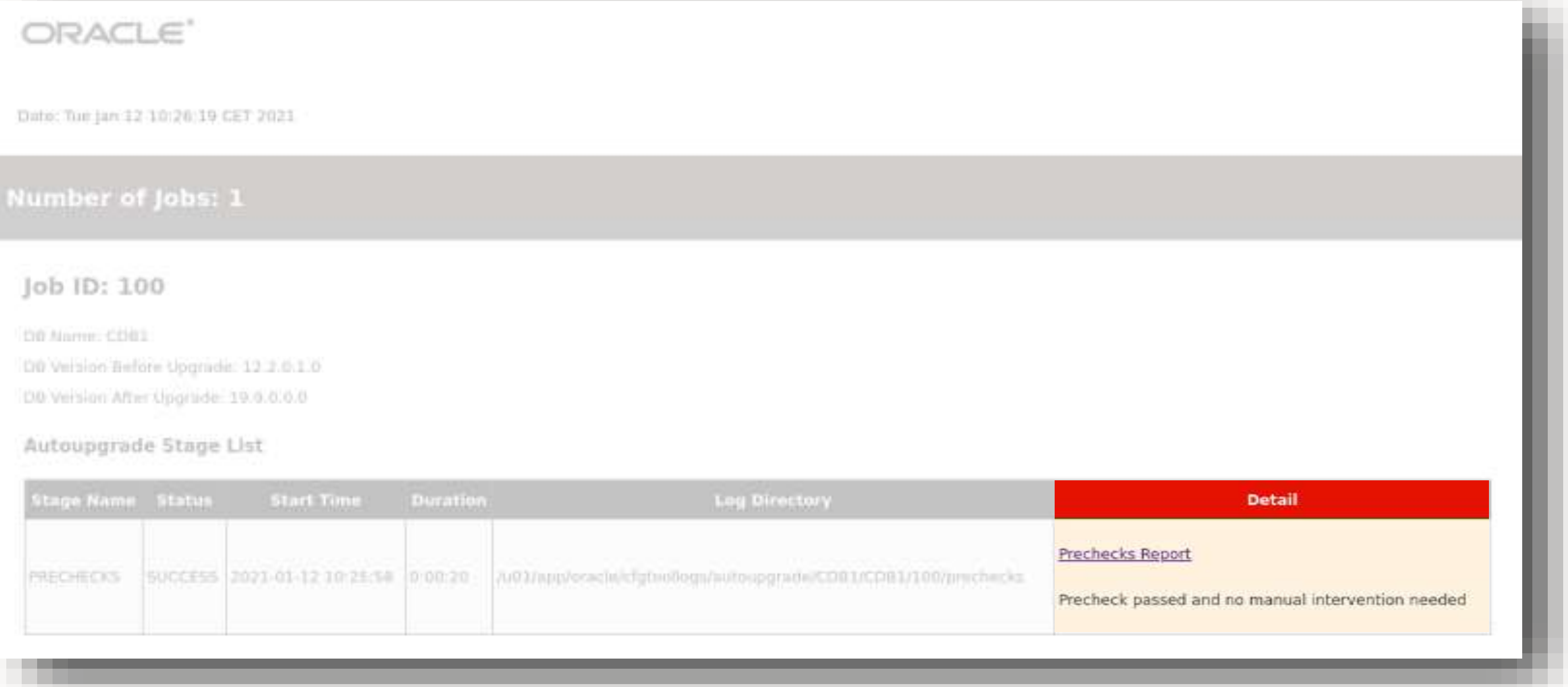
Configure

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

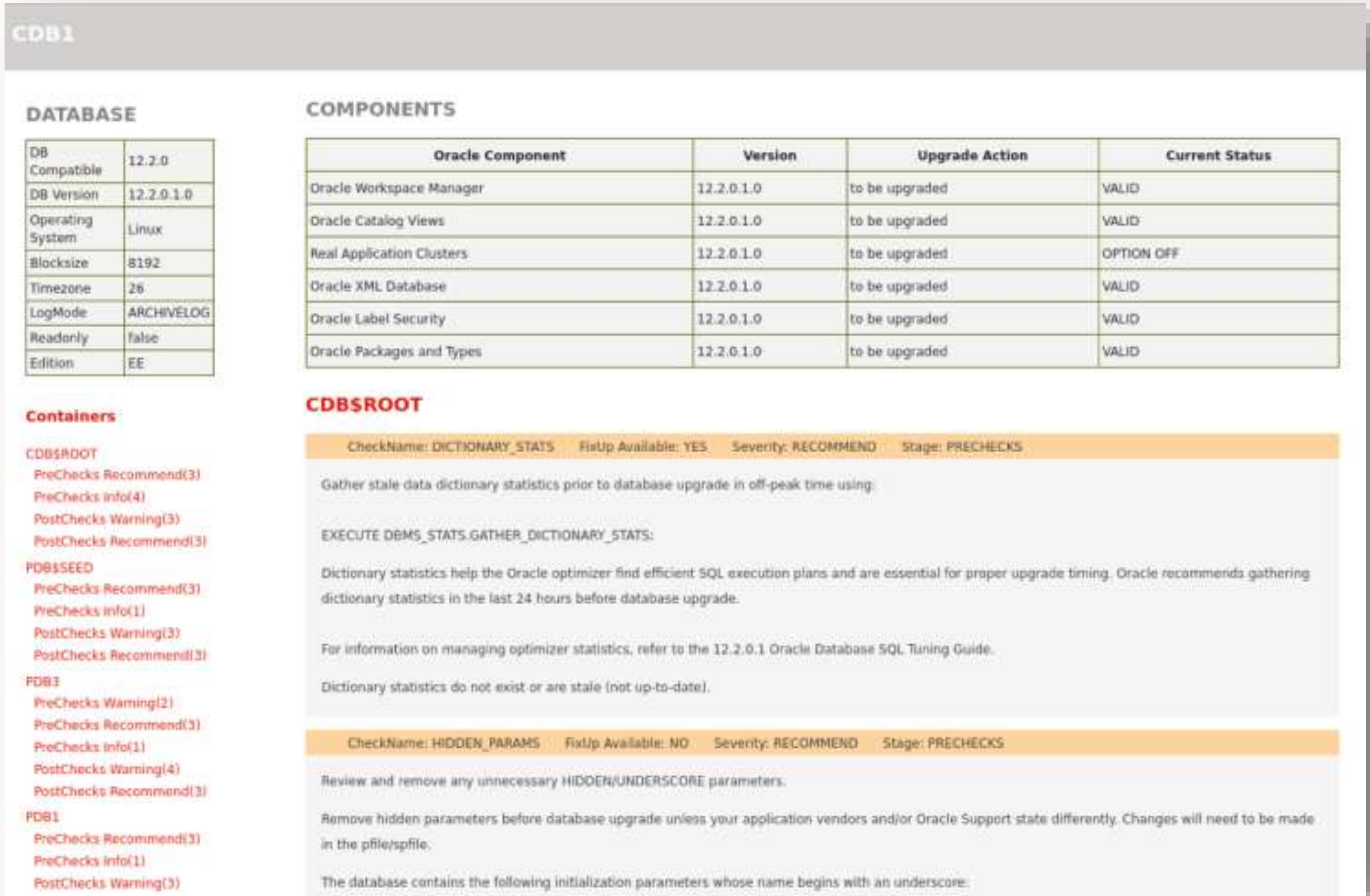
Download

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

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)

PDB3
- PreChecks Warning(2)
- PreChecks Recommend(3)
- PreChecks Info(1)
- PostChecks Warning(4)
- PostChecks Recommend(3)

PDB1
- PreChecks Recommend(3)
- PreChecks Info(1)
- PostChecks Warning(3)

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

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 (utlrp.sql)
- Time zone file upgrade
- Postupgrade fixups
- … and so much more

AutoUpgrade | Essentials

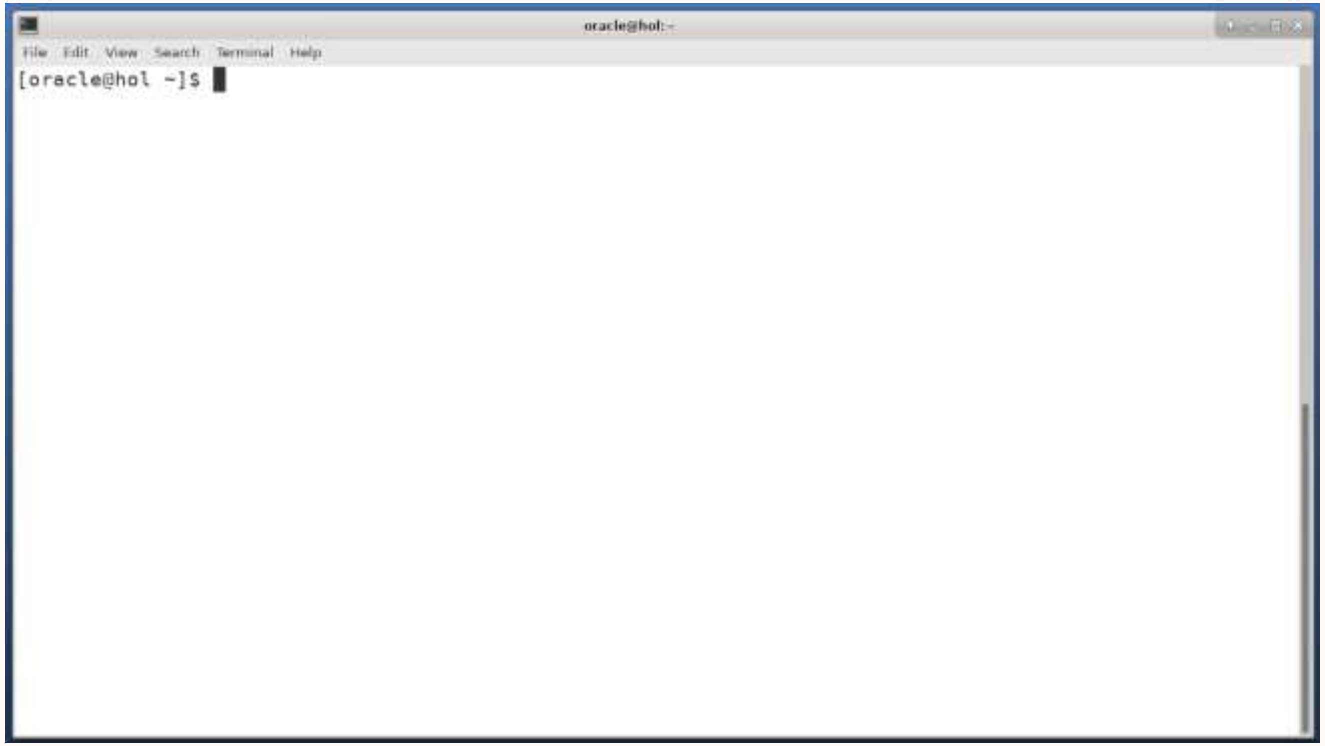
Download

Configure

Analyze

Check

Upgrade



```
[oracle@hol ~]$
```

Watch on YouTube

AutoUpgrade | Essentials

One-liner using environment variables

```
export ORACLE_SID=CDB1
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1
export ORACLE_TARGET_HOME=/u01/app/oracle/product/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/product/12.2.0.1,target_home=/u01/app/oracle/product/19" \
   -mode analyze
```

AutoUpgrade Advanced Options

Photo by Ciprian Boiciuc on Unsplash

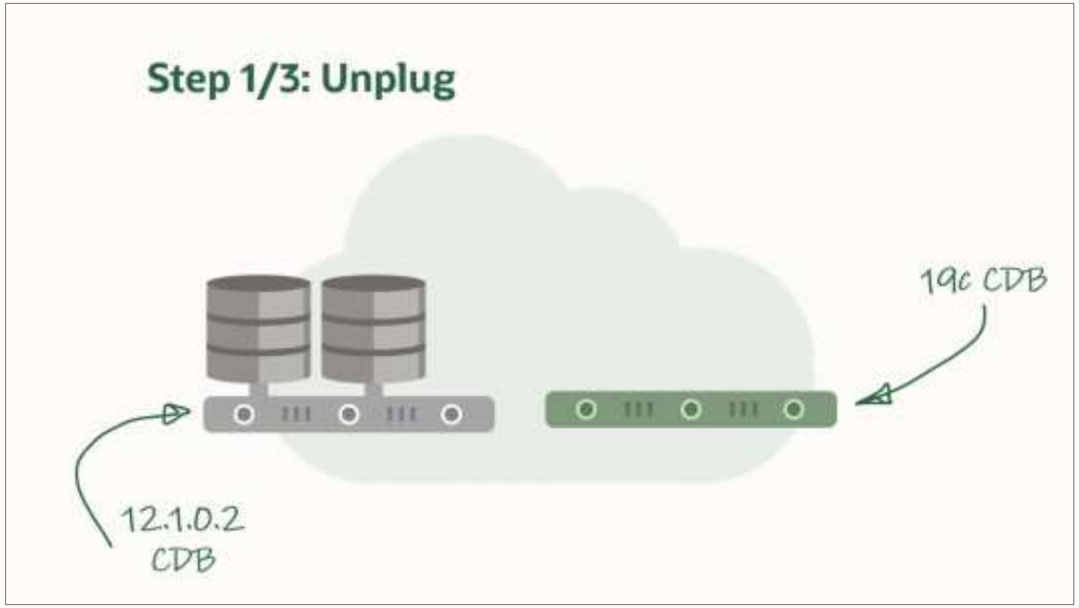
Photo by Katarzyna Pe on Unsplash

Unplug-Plug Upgrade

Unplug-plug | Overview

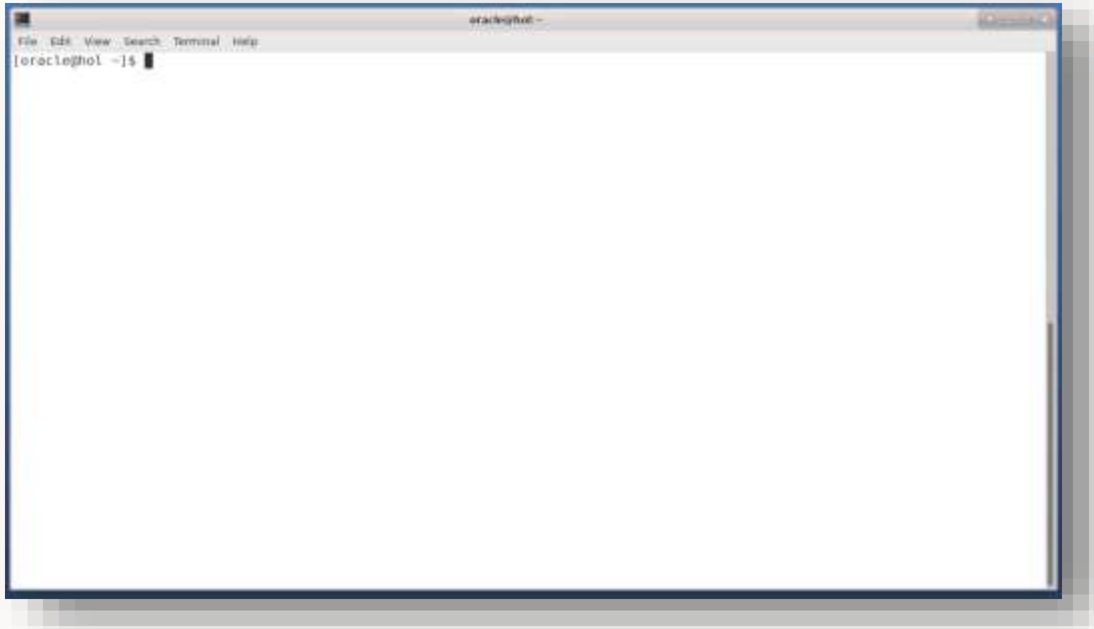
Upgrade a single PDB

- Faster
- More flexible
- Requires compatible target CDB
- Not compatible with Flashback Database
  - Consider using Refreshable PDBs
  - Copy data files (`target_pdb_copy_option`)



Unplug-plug | Demo

```
upg1.sid=CDB12102
upg1.target_cdb=CDB19
upg1.pdbs=pdb1
upg1.source_home=/u01/app/oracle/product/12102
upg1.target_home=/u01/app/oracle/product/19
```



```
[oracle@hol ~]$
```

Watch on YouTube

Unplug-plug | Options

Upgrade several PDBs

```
upg1.pdbs=pdb1,pdb2,pdb3
```

Rename a PDB

```
upg1.pdbs=pdb1
upg1.target_pdb_name.pdb1=sales
```

Copy data files on plug-in

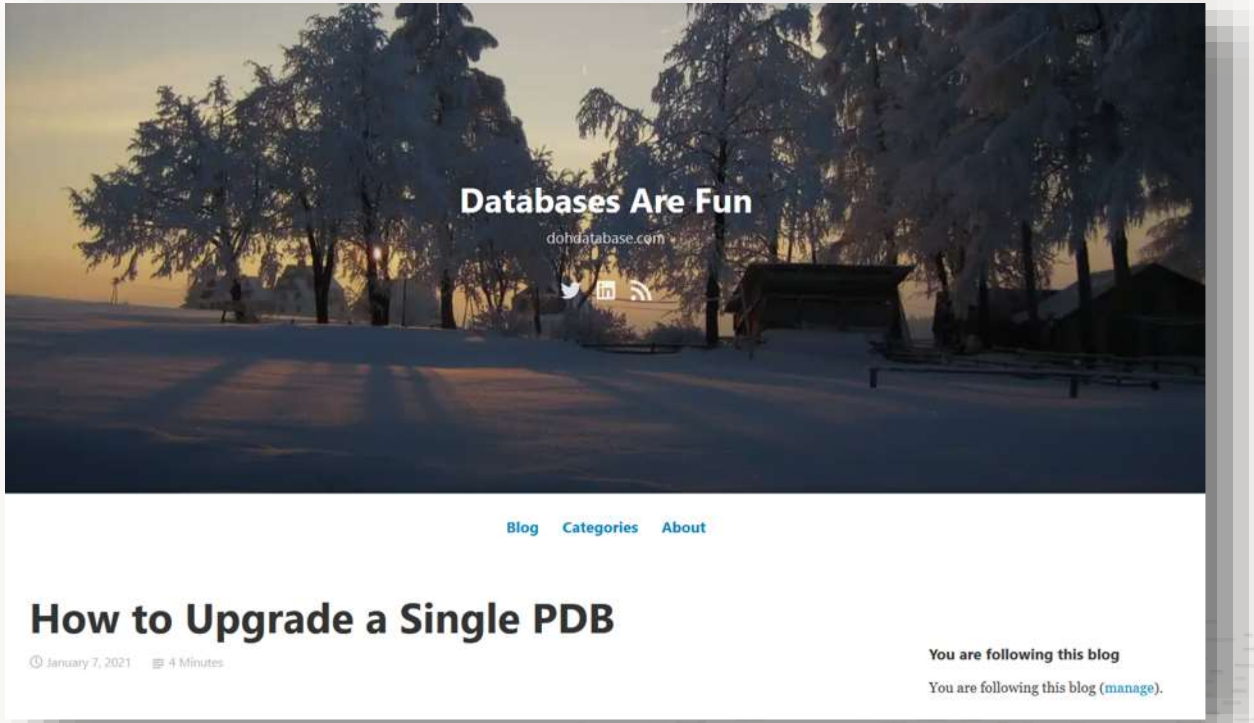
```
upg1.pdbs=pdb1
upg1.target_pdb_copy_option.pdb1=file_name_convert=('pdb1','sales')
```

Unplug-plug | Good To Know

Current limitations:

- Does not support Data Guard
- Does not support TDE Tablespace Encryption

Unplug-plug | Further Reading



Databases Are Fun

dohdatabase.com

Blog Categories About

How to Upgrade a Single PDB

January 7, 2021 4 Minutes

You are following this blog

You are following this blog (manage).

https://dohdatabase.com/how-to-upgrade-a-single-pdb

Photo by Danilo Alvesd on Unsplash

PDB Conversion

PDB Convert | Re-use data files

Fully automated plug-in, re-use data files

```
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
```

PDB Convert | Copy data files

Fully automated plug-in, copy data files

```
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
#Copy files and perform search/replace on file names
upg1.target_pdb_copy_option=file_name_convert=('DB19','SALES')
#Copy files and generate new OMF file names
upg1.target_pdb_copy_option=file_name_convert=none
```

Command

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

PDB Convert | Plug in and upgrade

Upgrade - and plug in

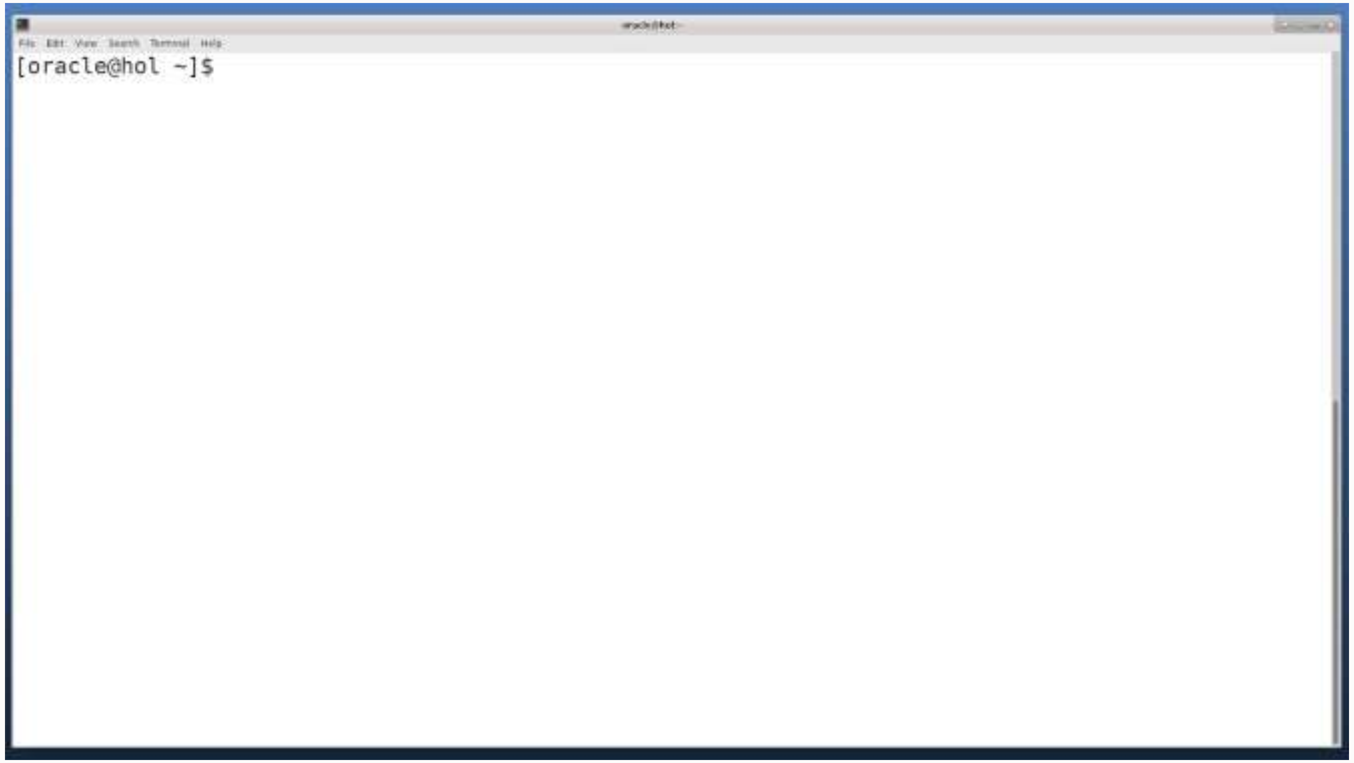
```
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
upg1.target_cdb=CDB2
#Optionally, rename PDB
#upg1.target_pdb_name=SALES
```

Command

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

Pro tip: You can find more details in Oracle AutoUpgrade between two servers – and Plugin?

PDB Convert | Demo



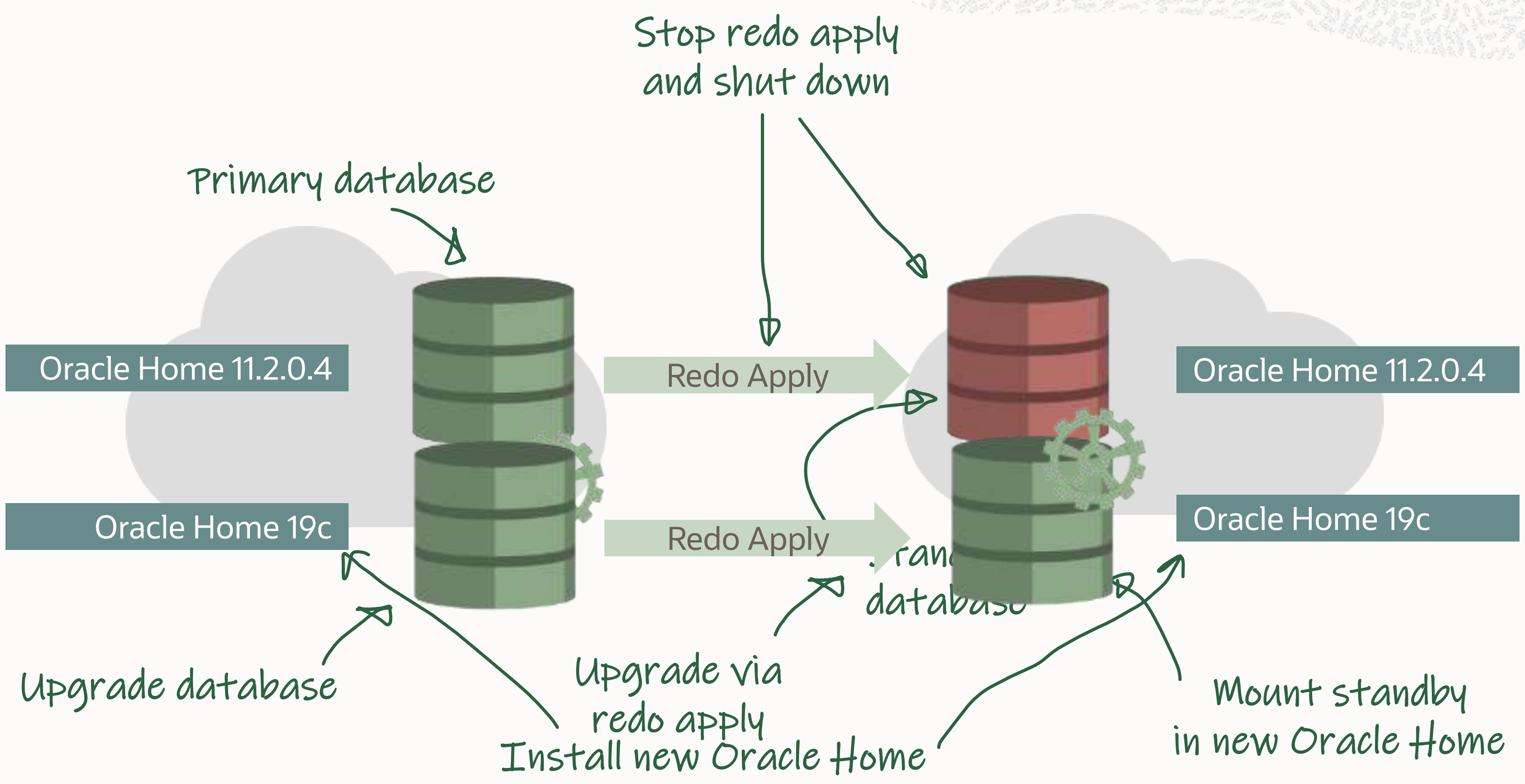
```
[oracle@hol ~]$
```

Watch on YouTube

Photo by Adam Muise on Unsplash

AutoUpgrade with Data Guard

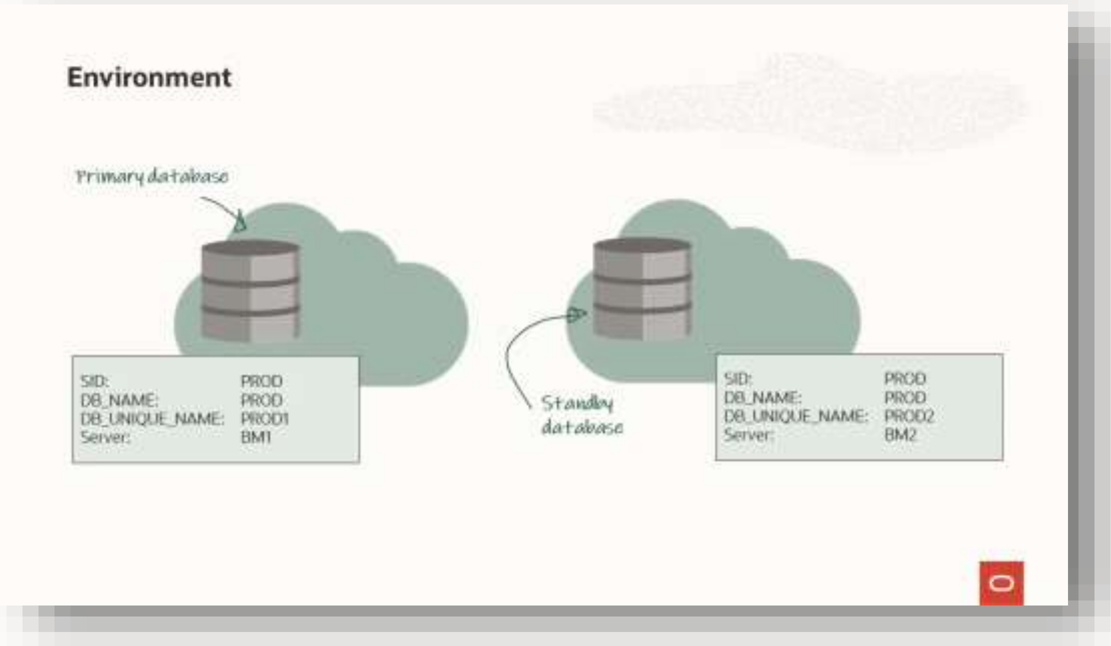
Data Guard | Concept



Remember use latest
Release Update

Data Guard | Overview

- Detected by AutoUpgrade automatically
- Works for broker-managed and manual Data Guard environments
- Primary database handled by AutoUpgrade Standby database handled manually



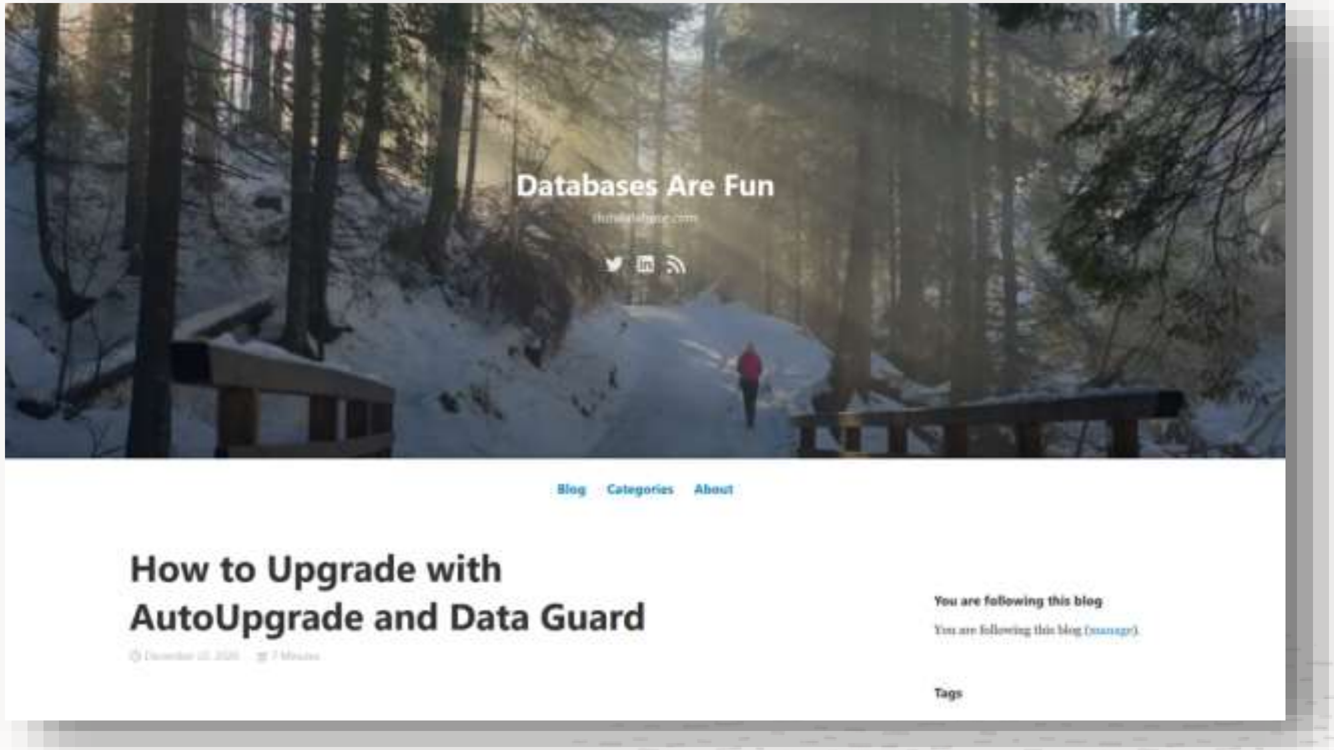
Environment

SID:	PROD
DB_NAME:	PROD
DB_UNIQUE_NAME:	PROD1
Server:	BM1

SID:	PROD
DB_NAME:	PROD
DB_UNIQUE_NAME:	PROD2
Server:	BM2

Watch on YouTube

Data Guard | Further Reading



Databases Are Fun

dohdatabase.com

Blog Categories About

How to Upgrade with AutoUpgrade and Data Guard

December 10, 2020 7 Minutes

You are following this blog

You are following this blog (manage).

Tags

https://dohdatabase.com/how-to-upgrade-with-autoupgrade-and-data-guard

Data Guard | YouTube Video

https://www.youtube.com/watch?v=JcIMlEQLWd0

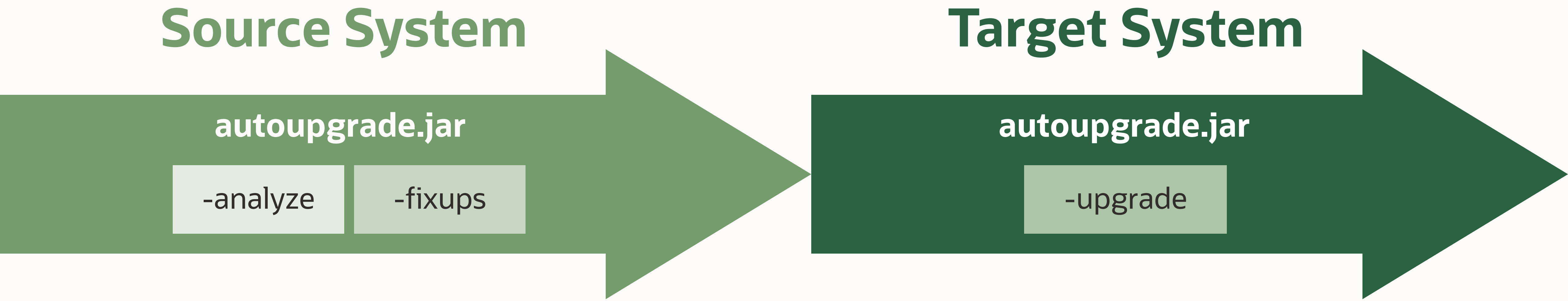
Data Guard | MAA Approach

- Keep standby online during upgrade
- Allows for faster go-live after upgrade
- Move standby database to new Oracle Home before upgrade
- Keep redo transport and redo apply on during upgrade

Photo by Hello I'm Nik 🎞 on Unsplash

AutoUpgrade to a New Server

Upgrade and Move to 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 and Move to 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

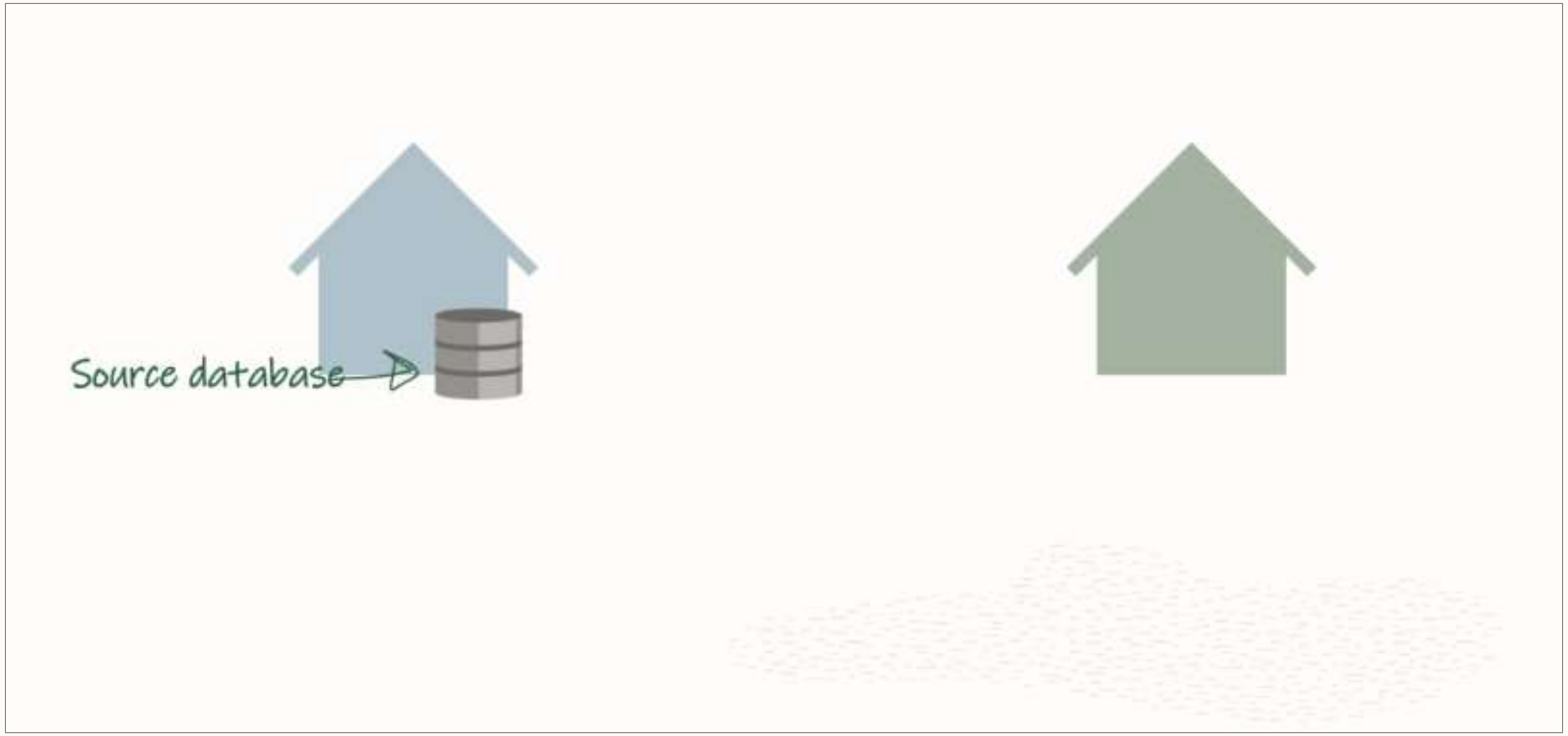
- Update `/etc/oratab`
- Prepare `?/network/admin` files
- `STARTUP UPGRADE`

```
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

Upgrade and Move to New Server | Demo



Watch on YouTube

Photo by Joshua Fernandez on Unsplash

Compatible

Compatible | Recommendation

When should you change `COMPATIBLE`?

A week or two after the upgrade - requires a database restart

Caution: When you change `COMPATIBLE` you can't:

- Flashback to restore point
- Downgrade

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 | AutoUpgrade

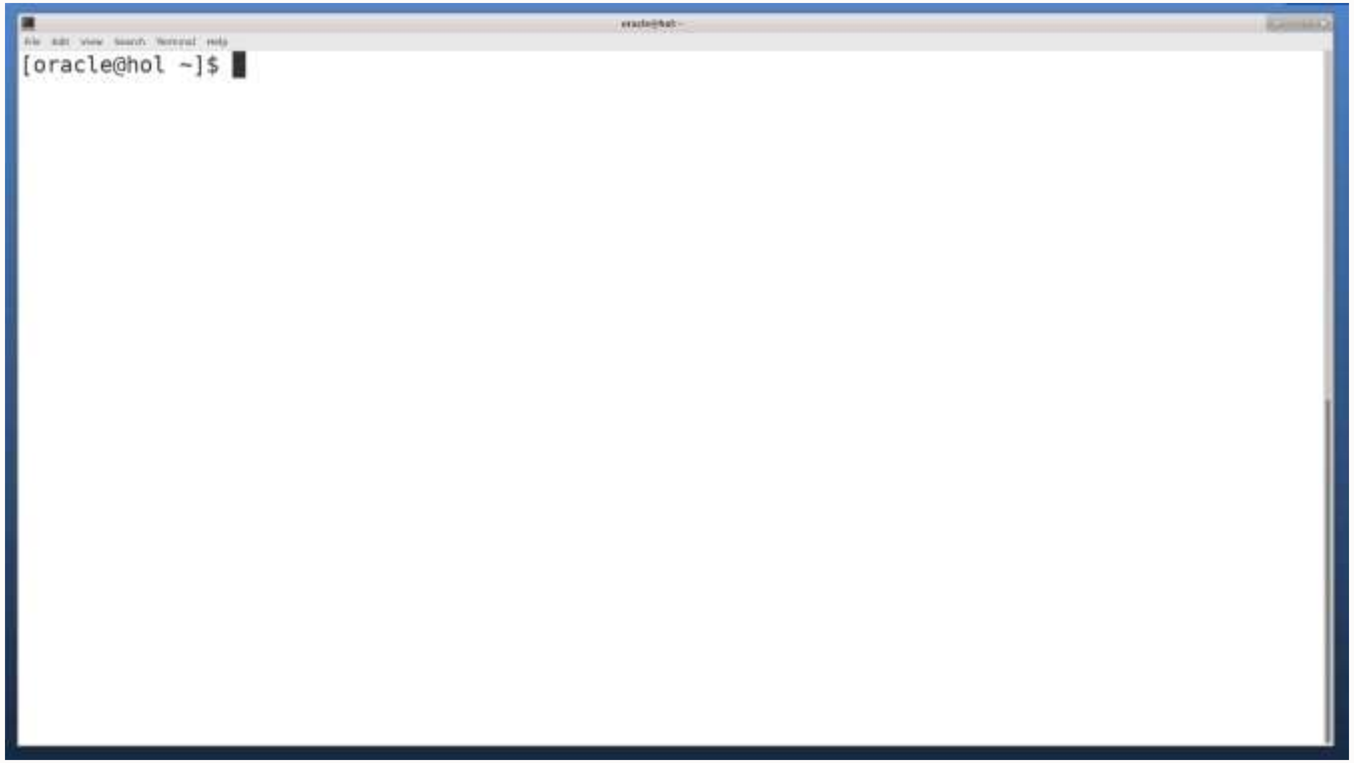
AutoUpgrade does not change `COMPATIBLE`

BREAKING NEWS

Next version of AutoUpgrade (21.1.3) supports changing compatible

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

Compatible | Demo



```
[oracle@hol ~]$ 
```

Watch on YouTube

Photo by Joshua Hoehne on Unsplash

Restore

Restore | Overview

Use AutoUpgrade to:

- Flashback the database
- Revert a plug-in operation (only when data files are copied)
- Revert a non-CDB to PDB conversion (only when data files are copied)

```
java -jar autoupgrade.jar -restore -jobs n
```

If you revert or restore in any other way, you must tell AutoUpgrade

Restore | Flashback

Standard technique in AutoUpgrade

- COMPATIBLE must not be changed

Pre Upgrade Environment	Post Upgrade Environment

CREATE RESTORE POINT **grpt**
GUARANTEE FLASHBACK DATABASE;	

UPGRADE	
	SHUTDOWN IMMEDIATE
	STARTUP MOUNT;
	FLASHBACK DATABASE TO RESTORE POINT **grpt**;
	SHUTDOWN IMMEDIATE
STARTUP MOUNT;	
ALTER DATABASE OPEN RESETLOGS;	
DROP RESTORE POINT **grpt**;	

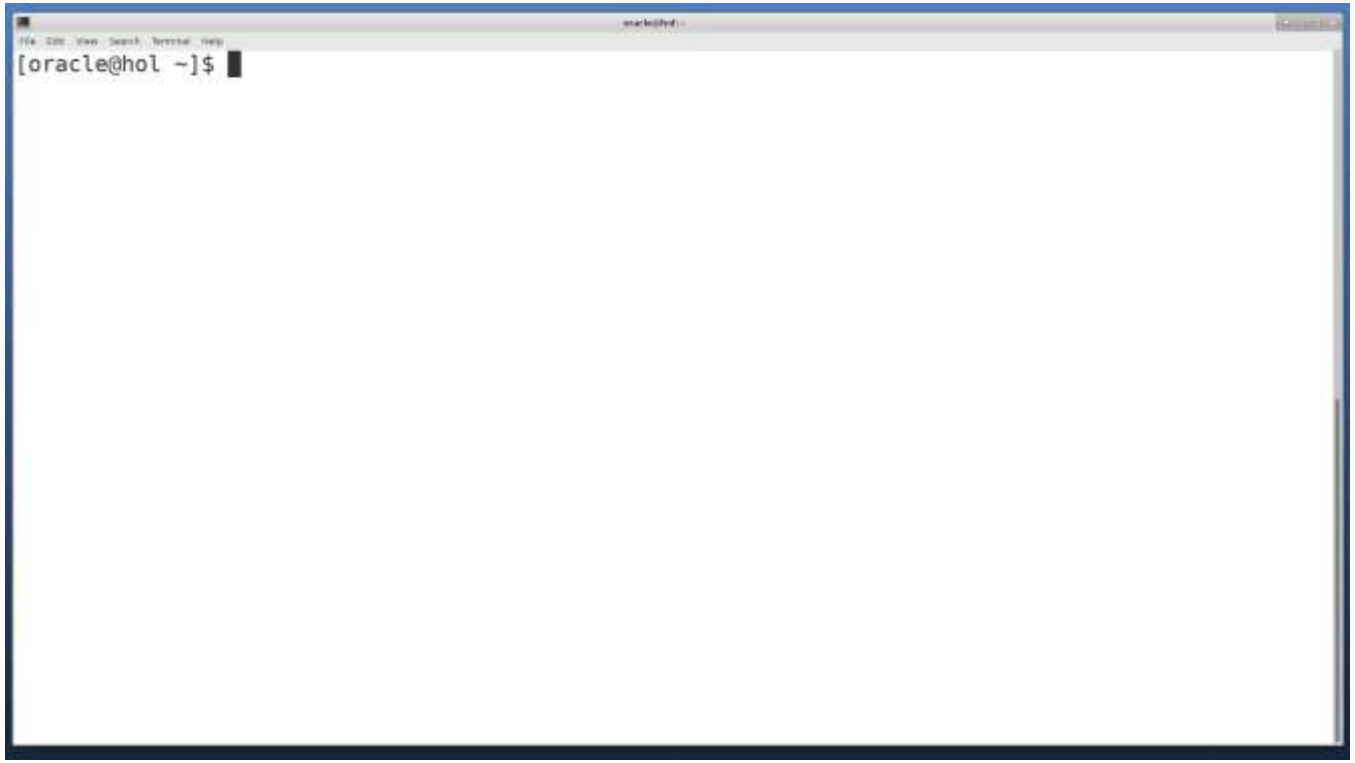
Restore | Overview

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

Restore | Demo



```
[oracle@hol ~]$ 
```

Watch on YouTube

Restore | What if ... you need to restart?

If you revert or restore in any other way, you must tell AutoUpgrade

1. Clear recovery data for a specific job

```
java -jar autoupgrade.jar -config DB.cfg -clear_recovery_data -job n
```

2. Clear all recovery data

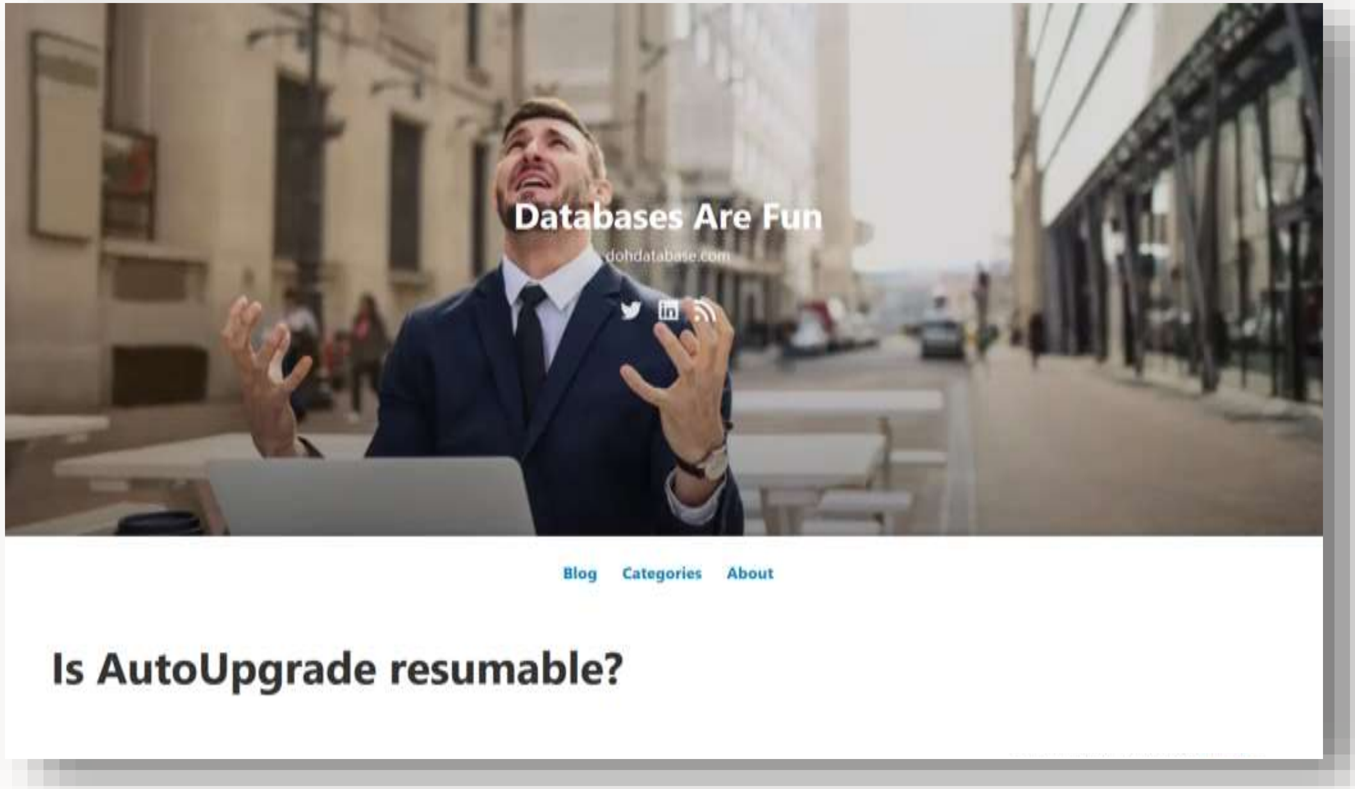
```
java -jar autoupgrade.jar -config DB.cfg -clear_recovery_data
```

3. Remove AutoUpgrade log directories - use with caution

```
global.autoupg_log_dir=/u01/app/oracle/cfgtoollogs/autoupgrade
upg1.log_dir=/u01/app/oracle/admin/DB1/upglogs

rm -rf /u01/app/oracle/cfgtoollogs/autoupgrade
rm -rf /u01/app/oracle/admin/DB1/upglogs
```

Restore | What if ... you need to restart?



Databases Are Fun

dohdatabase.com

Is AutoUpgrade resumable?

https://dohdatabase.com/is-autoupgrade-resumable/

Photo by Kristine Tumanyan on Unsplash

Change Parameters

Change Parameters | Overview

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

Change Parameters | Examples

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
```

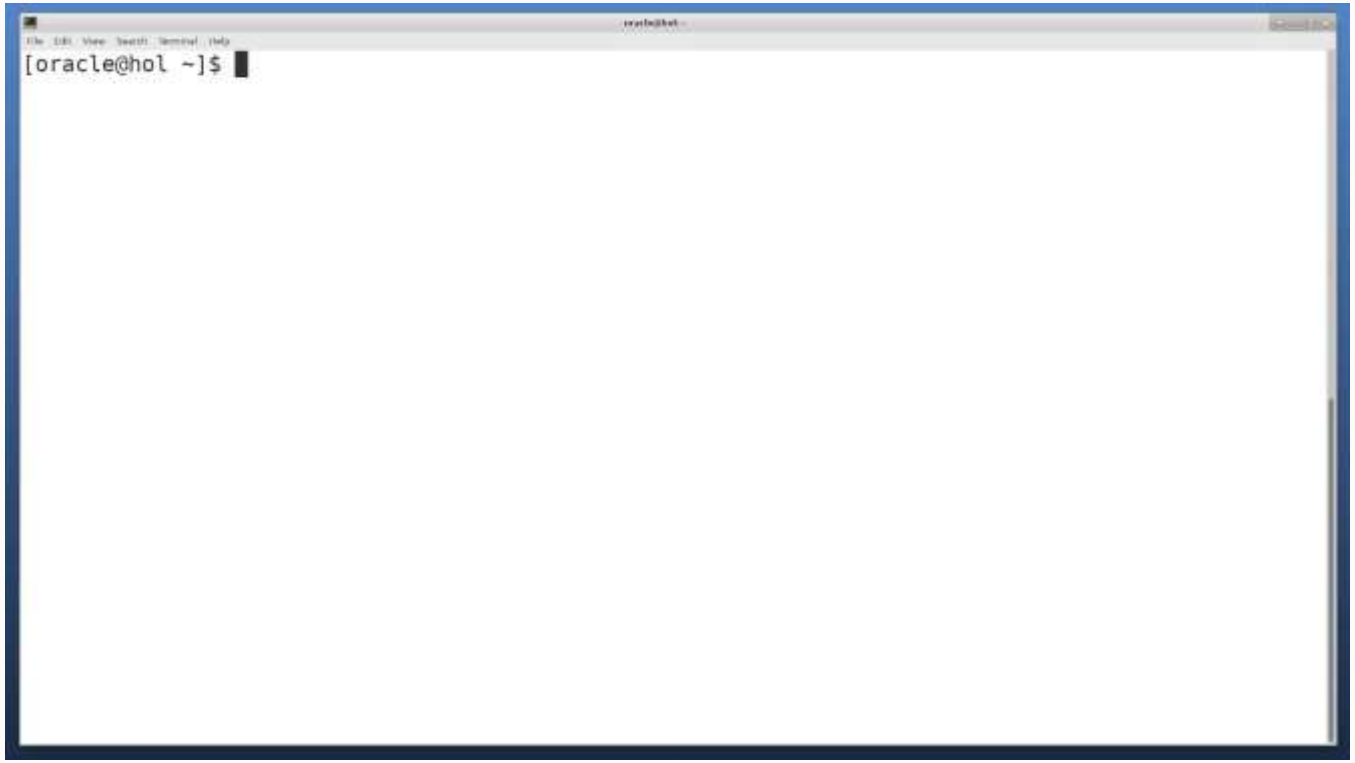
Add parameters to all databases after upgrade

```
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
```

Change Parameters | Demo



```
[oracle@hol ~]$ 
```

Watch on YouTube

Photo by Gita Krishnamurti on Unsplash

Execute Scripts

Execute Scripts | Overview

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

Execute Scripts | Example

```
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: `before_action` and `after_action` works on all upgrades or a specific upgrade

Execute Scripts | Return Code

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
```

Execute Scripts | Gathering Fixed Objects Stats

> ”*After an upgrade, or after other database configuration changes, Oracle strongly recommends that you regather fixed object statistics after you have run representative workloads on Oracle Database.*

Database 19c Upgrade Guide, chapter 7

Never run it right after upgrade

Execute Scripts | Gathering Fixed Objects Stats

Ask yourself: Do you remember this?

If not, DBMS_SCHEDULER to the rescue

Execute Scripts | Gathering Fixed Objects Stats

1. Create a .sql script

```
BEGIN
   DBMS_SCHEDULER.CREATE_JOB (
      job_name => '"SYS"."GATHER_FIXED_OBJECTS_STATS_ONE_TIME"',
          job_type => 'PLSQL_BLOCK',
          job_action => 'BEGIN DBMS_STATS.GATHER_FIXED_OBJECTS_STATS; END;',
          start_date => SYSDATE+7,
          auto_drop => TRUE,
          comments => 'Gather fixed objects stats after upgrade - one time'
   );
   DBMS_SCHEDULER.ENABLE (
      name => '"SYS"."GATHER_FIXED_OBJECTS_STATS_ONE_TIME"'
   );
END;
/
```

Execute Scripts | Gathering Fixed Objects Stats

2. Create a .sh script

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
   -n 4 -e \
   -C 'PDB$SEED' \
   -b sched_gfos -d /home/oracle/sched_gfos/ sched_gfos.sql
```

3. Execute .sh script after upgrade

```
upg1.after_action=/home/oracle/sched_gfos/sched_gfos.sh
```

Further information and non-CDB example in blog post

Photo by Mick Haupt on Unsplash

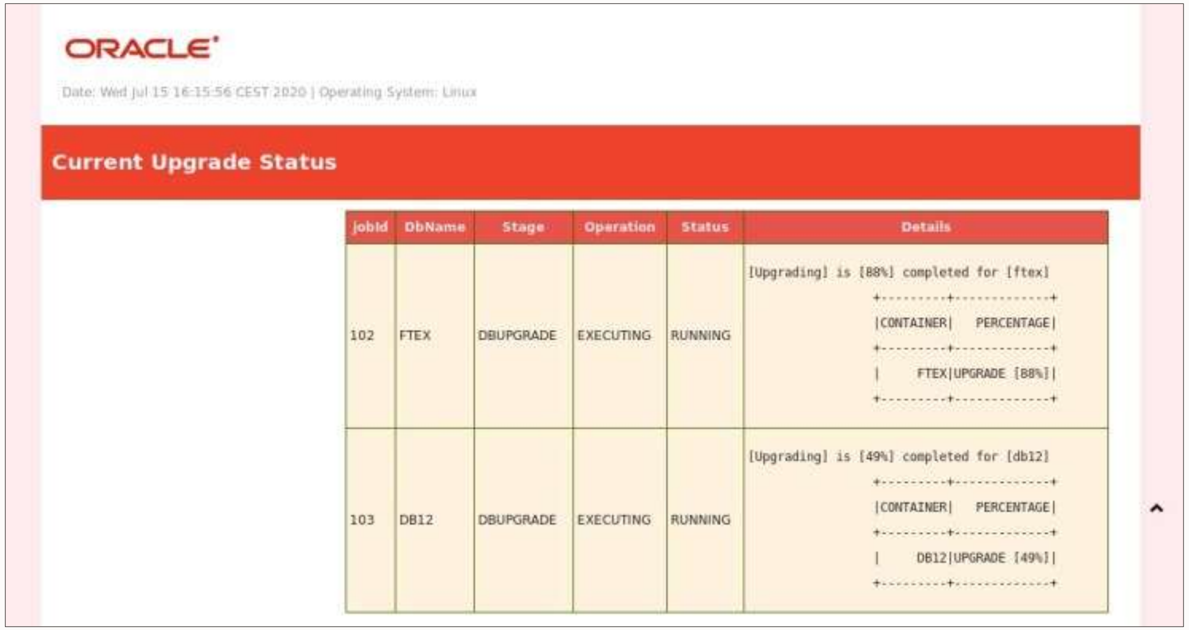
Monitoring

Monitoring | Status page

Use Python SimpleHTTPServer

```
$ cd <au_global_log_dir>/cfgtoollogs/upgrade/auto
$ python -m SimpleHTTPServer 8888
```

Then open your browser (`http://127.0.0.1:8000/state.html`)



ORACLE

Date: Wed Jul 15 16:15:56 CEST 2020 | Operating System: Linux

Current Upgrade Status

<table>
<tr><th>jobId</th><th>DbName</th><th>Stage</th><th>Operation</th><th>Status</th><th>Details</th></tr>
<tr><td>102</td><td>FTEX</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [88%] completed for [ftex]
+---------+-------------+
|CONTAINER| PERCENTAGE|
+---------+-------------+
| FTEX|UPGRADE [88%]|
+---------+-------------+</td></tr>
<tr><td>103</td><td>DB12</td><td>DBUPGRADE</td><td>EXECUTING</td><td>RUNNING</td><td>[Upgrading] is [49%] completed for [db12]
+---------+-------------+
|CONTAINER| PERCENTAGE|
+---------+-------------+
| DB12|UPGRADE [49%]|
+---------+-------------+</td></tr>
</table>

Pro tip: More details in blog post AutoUpgrade: Refresh Status Information Automatically

Monitoring | Demo

AutoUpgrade

One Single Command

Copyright © 2021, Oracle and/or its affiliates

[Watch on YouTube]

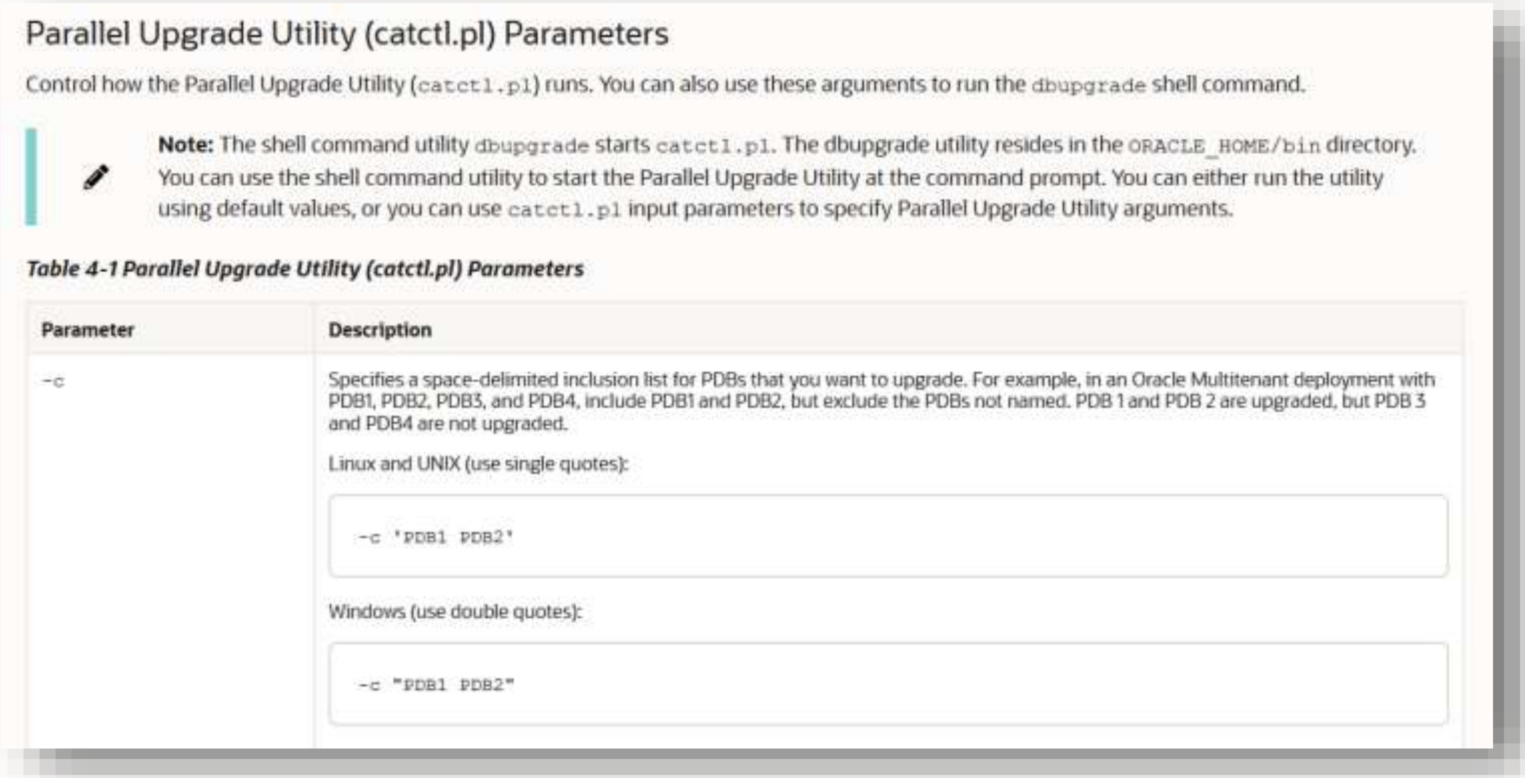
Photo by Danilo Alvesd on Unsplash

Catctl Options

Catctl Options | Overview

AutoUpgrades uses Parallel Upgrade Utility (catctl)

Catctl has advanced options



Parallel Upgrade Utility (catctl.pl) Parameters

Control how the Parallel Upgrade Utility (`catctl.pl`) runs. You can also use these arguments to run the `dbupgrade` shell command.

> **Note:** The shell command utility `dbupgrade` starts `catctl.pl`. The dbupgrade utility resides in the `ORACLE_HOME/bin` directory. You can use the shell command utility to start the Parallel Upgrade Utility at the command prompt. You can either run the utility using default values, or you can use `catctl.pl` input parameters to specify Parallel Upgrade Utility arguments.

Table 4-1 Parallel Upgrade Utility (catctl.pl) Parameters

Parameter	Description

`-c`	Specifies a space-delimited inclusion list for PDBs that you want to upgrade. For example, in an Oracle Multitenant deployment with PDB1, PDB2, PDB3, and PDB4, include PDB1 and PDB2, but exclude the PDBs not named. PDB 1 and PDB 2 are upgraded, but PDB 3 and PDB4 are not upgraded.
	Linux and UNIX (use single quotes):
	`-c 'PDB1 PDB2'`
	Windows (use double quotes):
	`-c "PDB1 PDB2"`

Catctl Options | AutoUpgrade

Some are available in AutoUpgrade

```
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

Catctl Options | Faster Upgrades

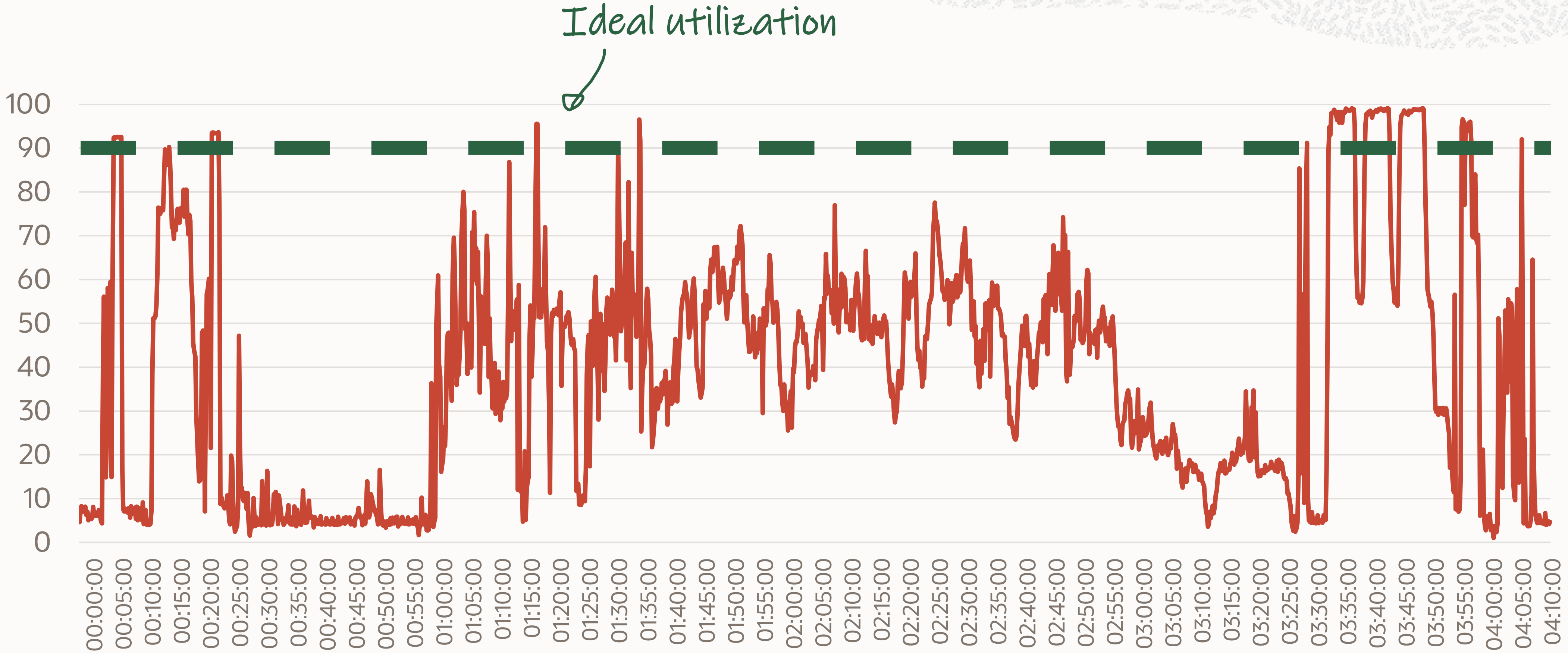
- OCI Bare Metal host
  - 16 OPCUs
  - 768 GB memory
  - NVMe disks

- CDB with 52 PDBs
  - `CPU_COUNT` = 32
  - `SGA_TARGET` = 80G
  - `PGA_AGGREGATE_TARGET` = 20G

- Many database components (17 in total)

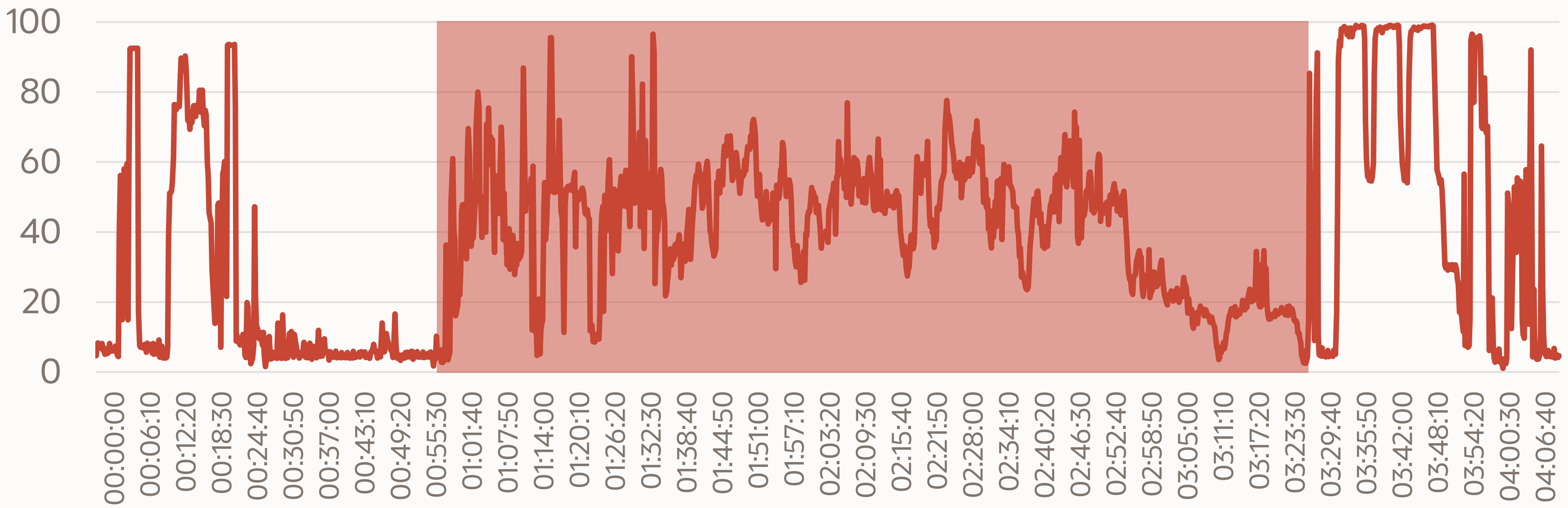
- **Upgrade from 12.1.0.2 to 19**

Catctl Options | Faster Upgrades



Total upgrade time: 4 hours 8 minutes

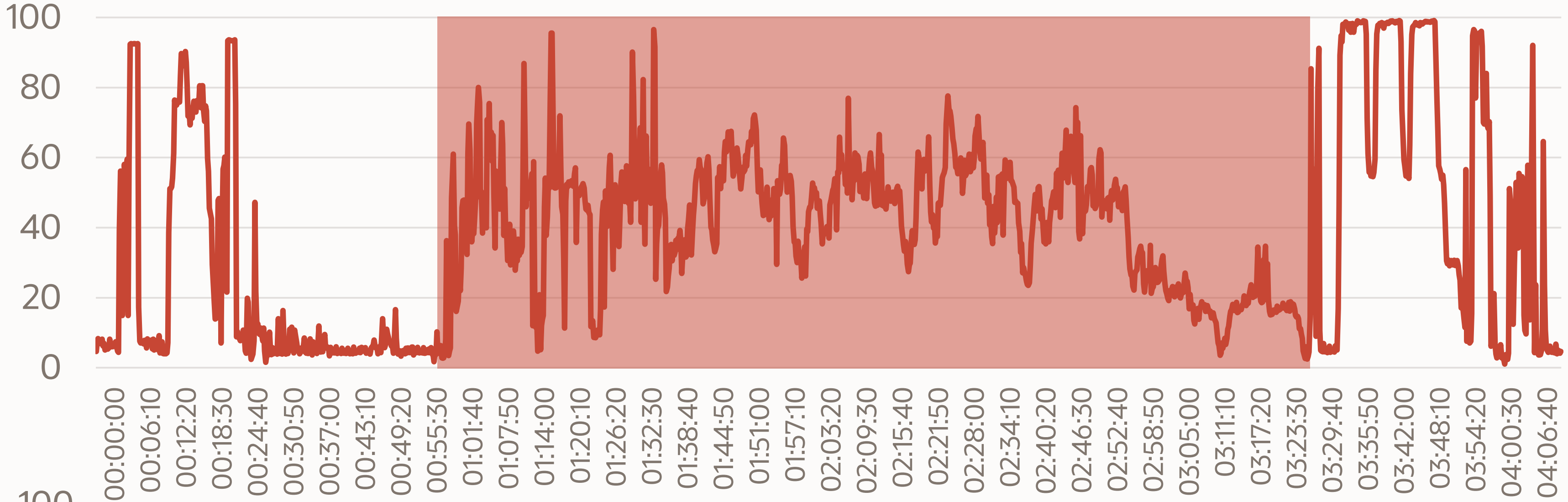
Catctl Options | Faster Upgrades



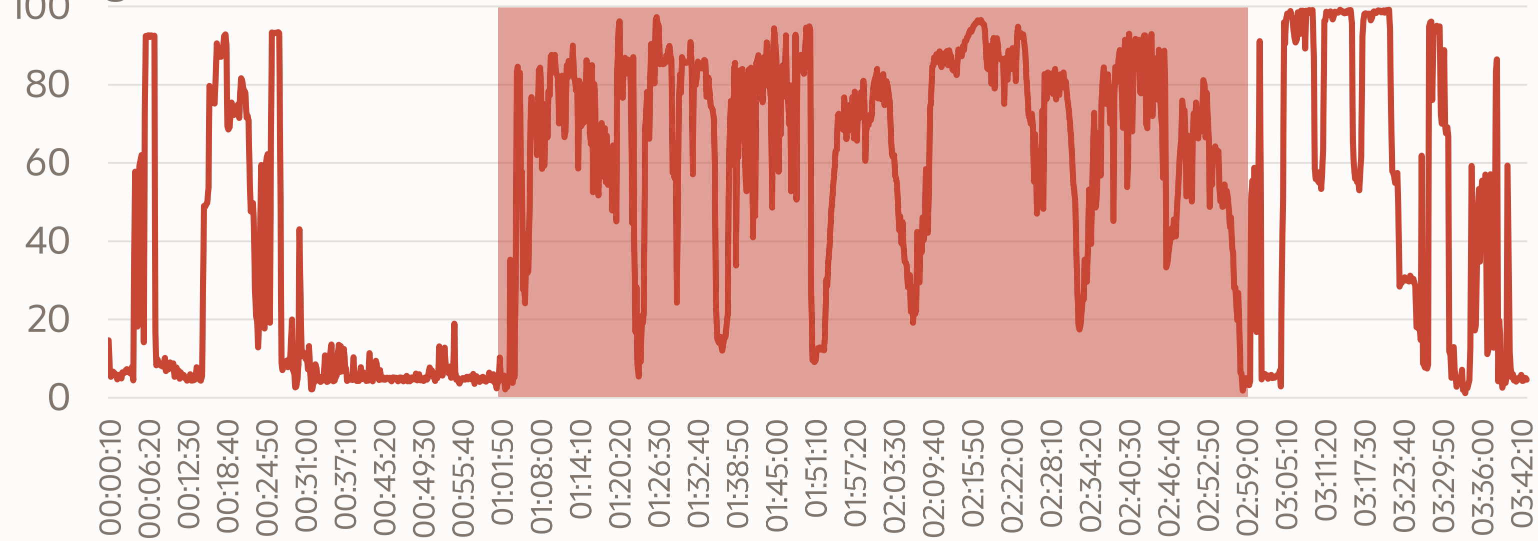
Upgrade PDB$SEED and user PDBs

- Add more PDBs (`catctl -n`)
- Keep parallel processes per PDB at default (2)
- Remove components from PDBs

Catctl Options | Faster Upgrades



32 parallel processes



54 parallel processes

`upg1.catctl_options=-n 54`

26 minutes faster

Pro tip: Remember to increase `PROCESSES` dramatically

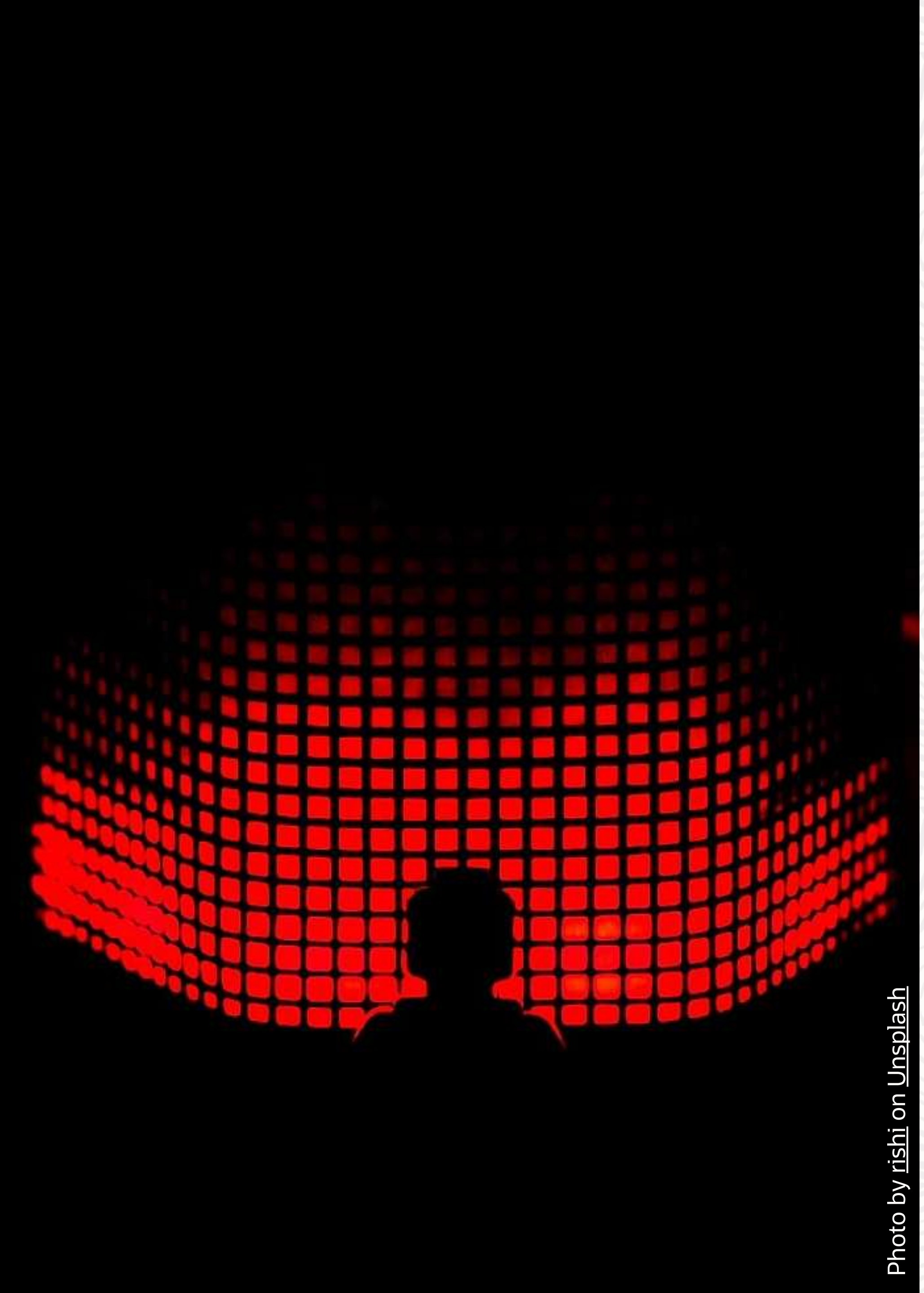


Photo by rishi on Unsplash

Zip Options

Zip Option | Overview

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
```

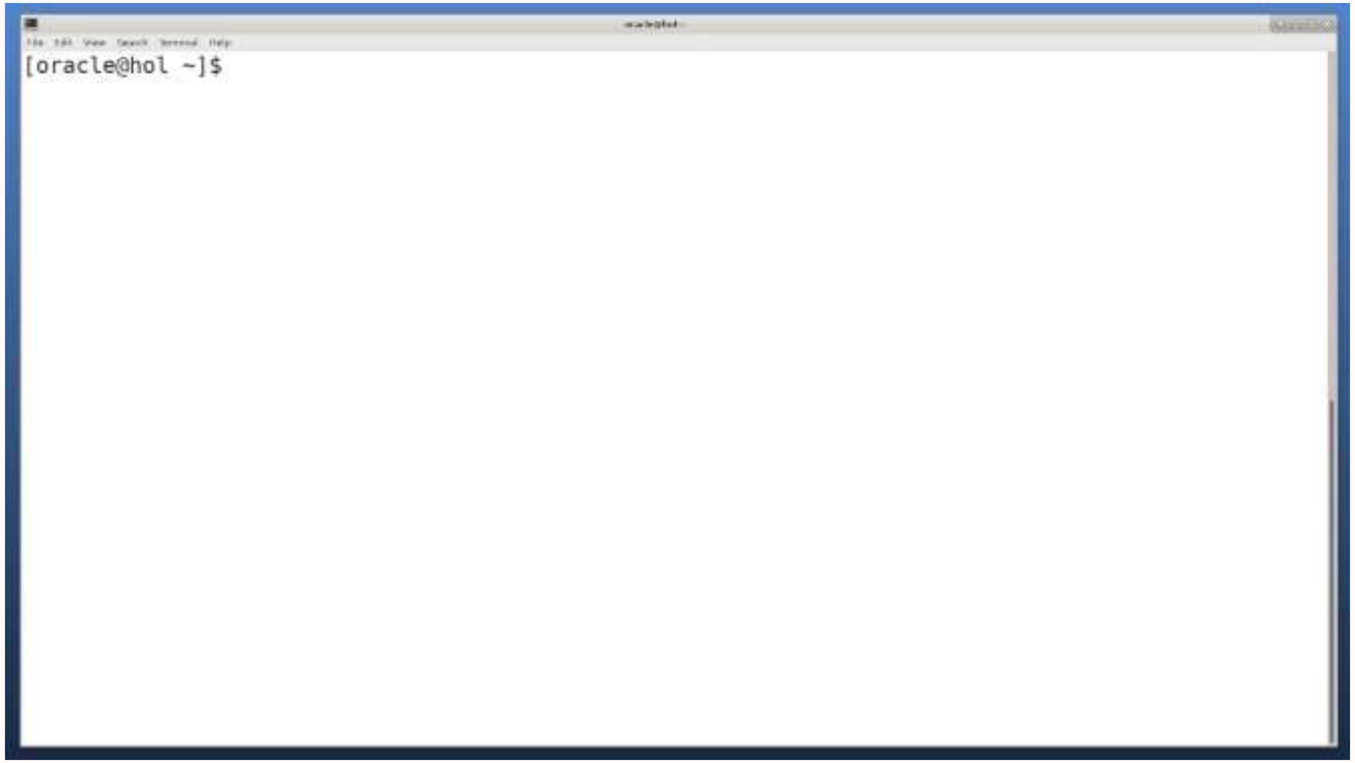
3. Upload it to My Oracle Support

My Oracle Support

Welcome

Search our Knowledge base and engage Communities to learn about products, services, and to find help resolving issues

Zip Option | Demo



```
[oracle@hol ~]$
```

Watch on YouTube

Photo by Diyana Qua on Unsplash

Error Codes

Error Codes | Overview

What does an error mean?

```
$ java -jar autoupgrade.jar -error_code UPG-1400

ERROR1400.ERROR  = UPG-1400
ERROR1400.CAUSE  = Database upgrade failed with errors
```

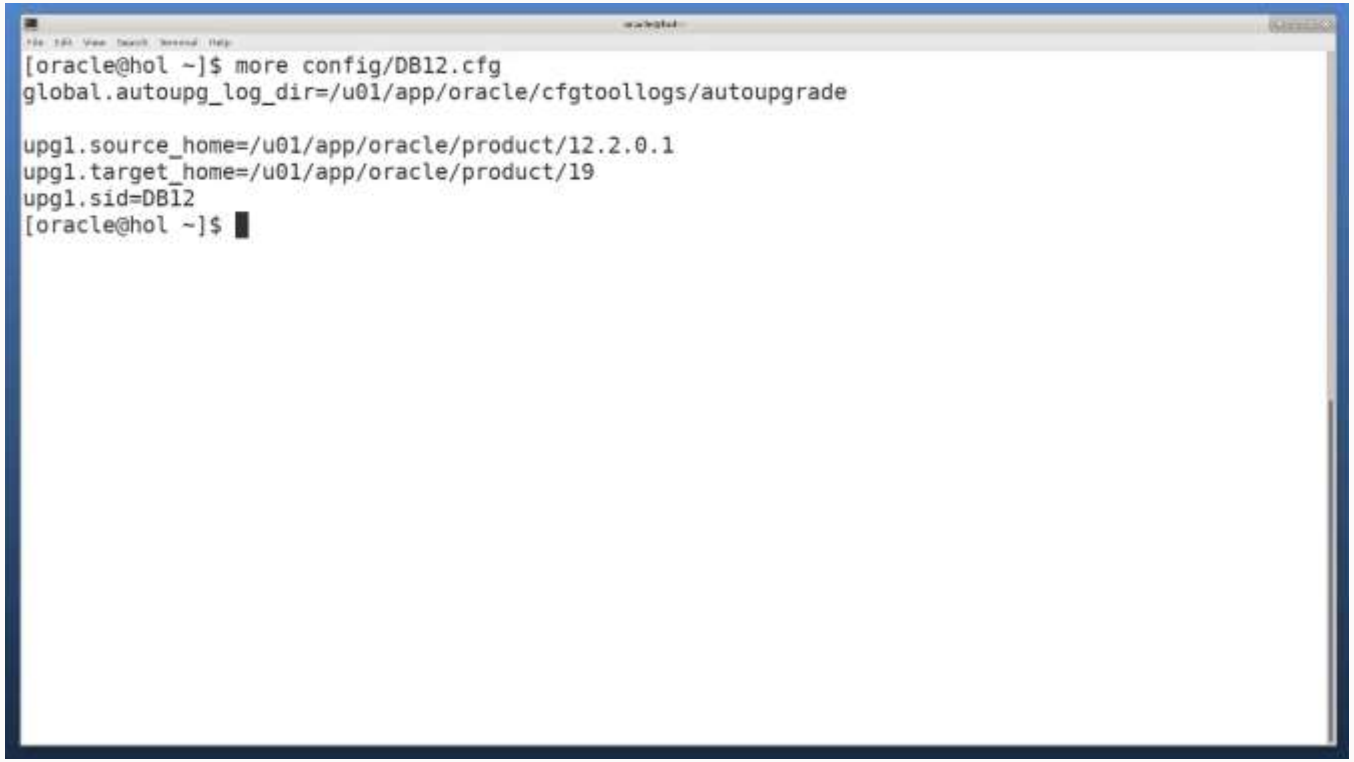
Omit the error code and get a list of all error codes

```
$ java -jar autoupgrade.jar -error_code

ERROR1000.ERROR  = UPG-1000
ERROR1000.CAUSE  = It was not possible to create the data file where the jobsTable is being written or there was a
problem during the writing, it might be thrown due to a permission error or a busy resource scenario

ERROR1001.ERROR  = UPG-1001
ERROR1001.CAUSE  = There was a problem reading the state file perhaps there was corruption writing the file and in
the next write it might be fixed
.
.
```

Error Codes | Demo



```
[oracle@hol ~]$ more config/DB12.cfg
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
[oracle@hol ~]$ 
```

Watch on YouTube

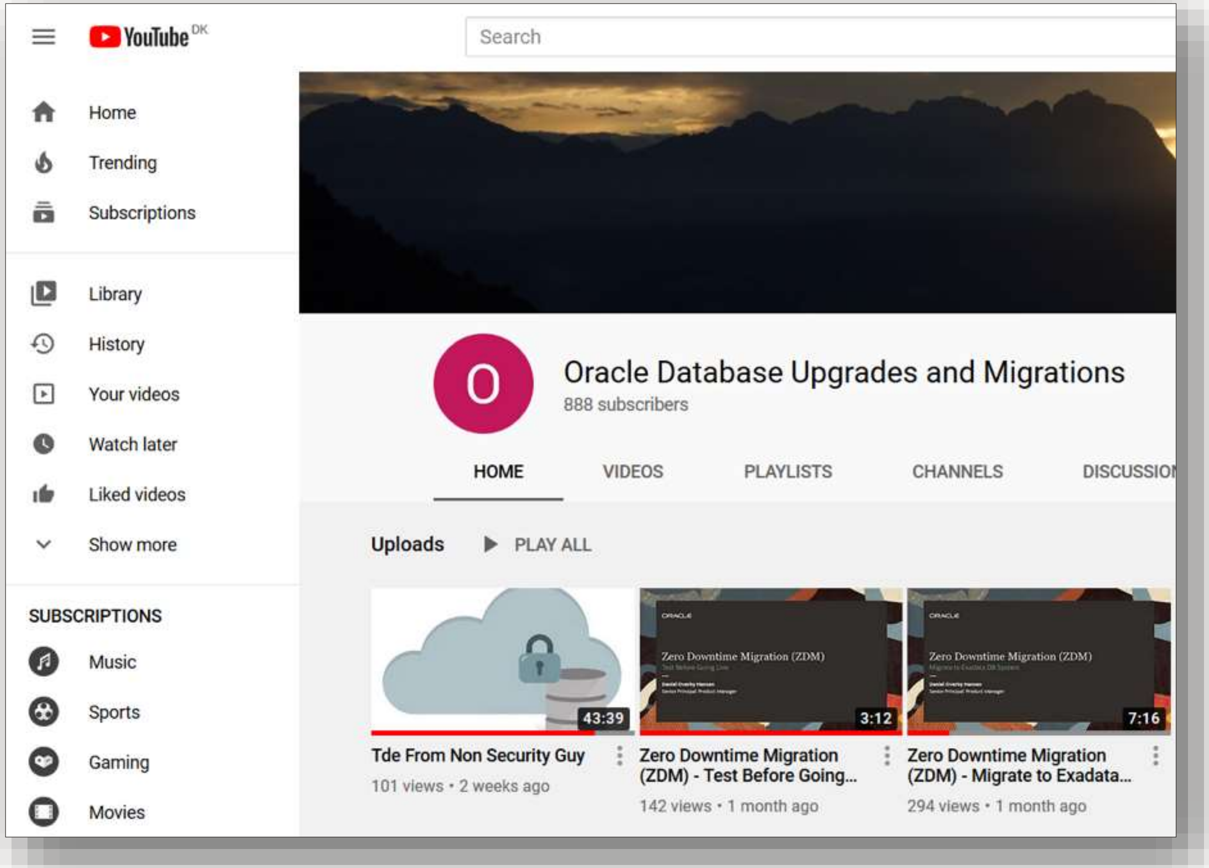
Photo by Justin Lim on Unsplash

Enjoy 18 hours of high-quality tech

https://dohdatabase.com/webinars/

- Release and Patching Strategy
- AutoUpgrade to Oracle Database 19c
- Performance Stability, Tips and Tricks and Underscores
- Migration to Oracle Multitenant
- Migration Strategies – Insights, Tips and Secrets
- Move to the Cloud – Not only for techies
- Cool Features – Not only for DBAs
- Database Upgrade Internals – and so much more
- Performance Stability for Cloud Migration

YouTube | Oracle Database Upgrades and Migrations



Oracle Database Upgrades and Migrations

888 subscribers

Tde From Non Security Guy
101 views • 2 weeks ago

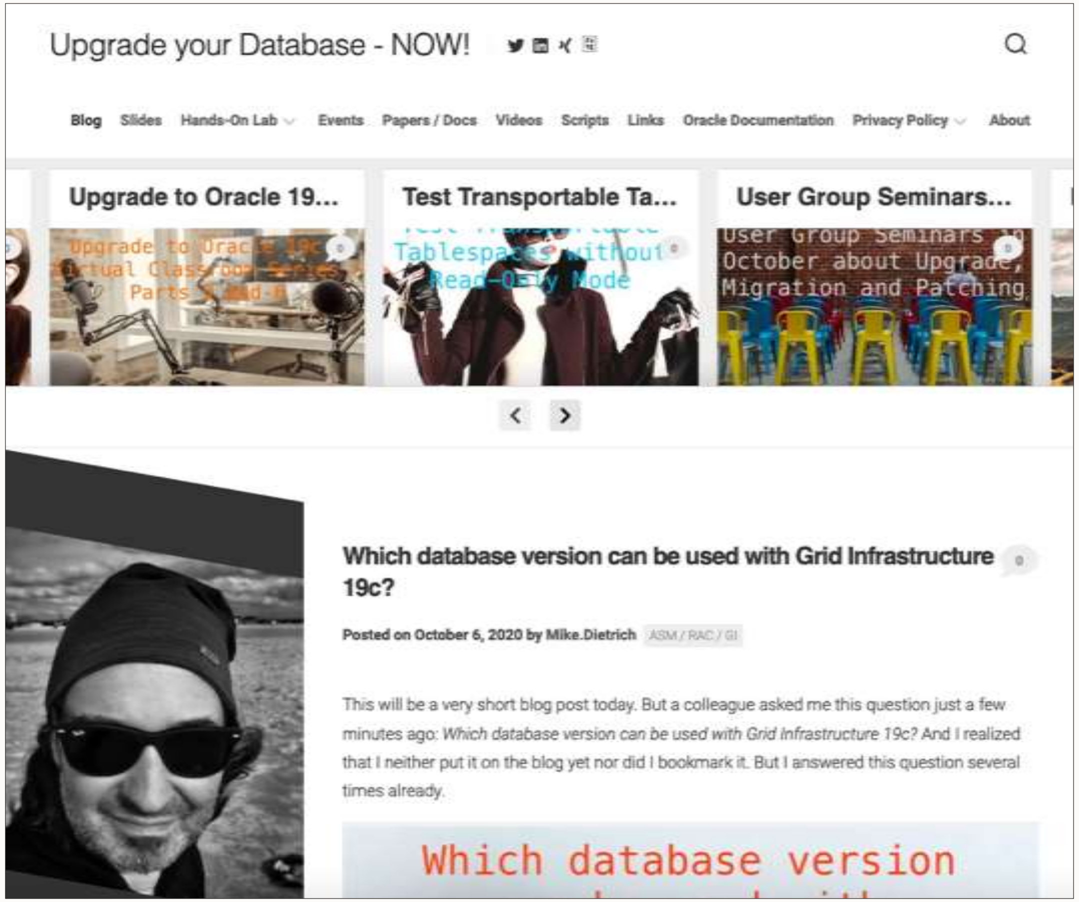
Zero Downtime Migration (ZDM) - Test Before Going...
142 views • 1 month ago

Zero Downtime Migration (ZDM) - Migrate to Exadata...
294 views • 1 month ago

YouTube Channel

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...

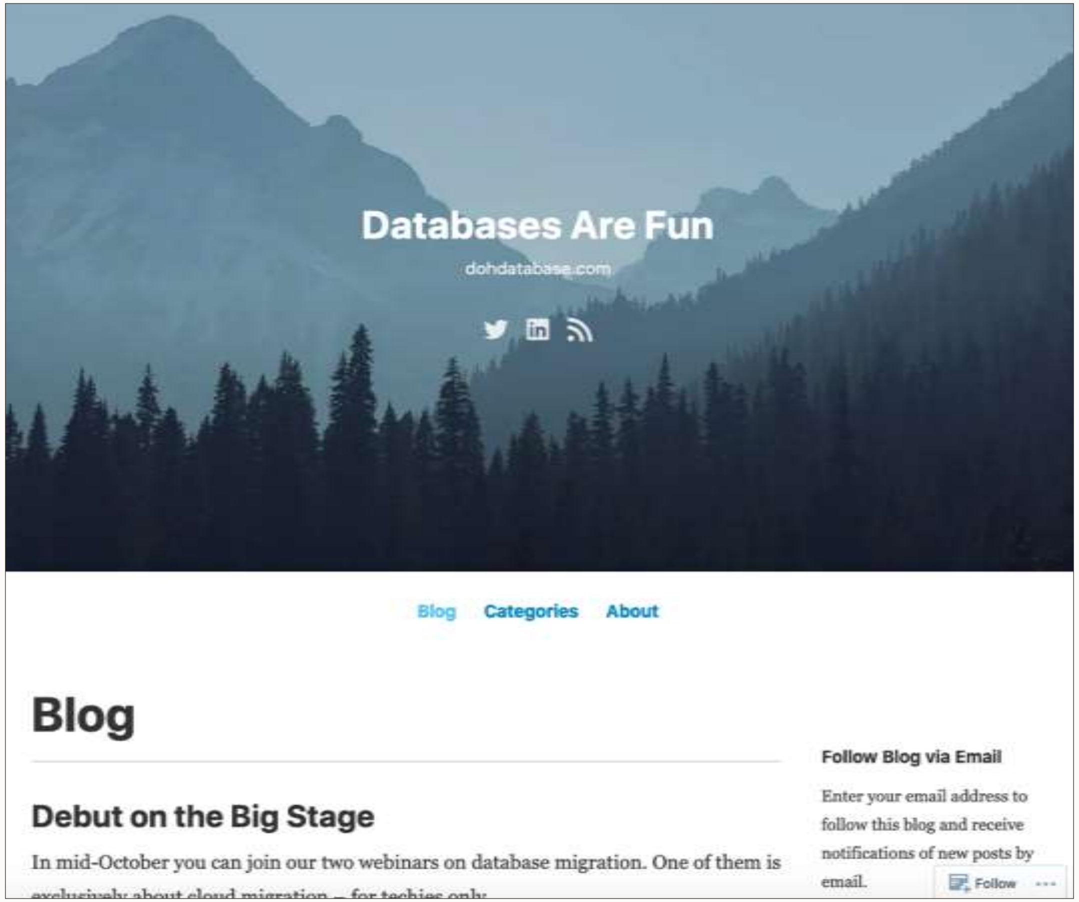
Upgrade to Oracle 19c Virtual Classroom Series Parts 5 and 6
Test Transportable Tablespaces without Read-Only Mode
User Group Seminars in October about Upgrade, Migration and Patching

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

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.

Thank you!