ORACLE

Migration Strategies

Tips and Tricks and Insights and Secrets

Mike Dietrich
Distinguished Product Manager
Database Upgrade and Migrations

Daniel Overby Hansen
Senior Principal Product Manager
Cloud Migration

Photo by Samuel Ferrara on Unsplash

Mike Dietrich

Distinguished Product Manager

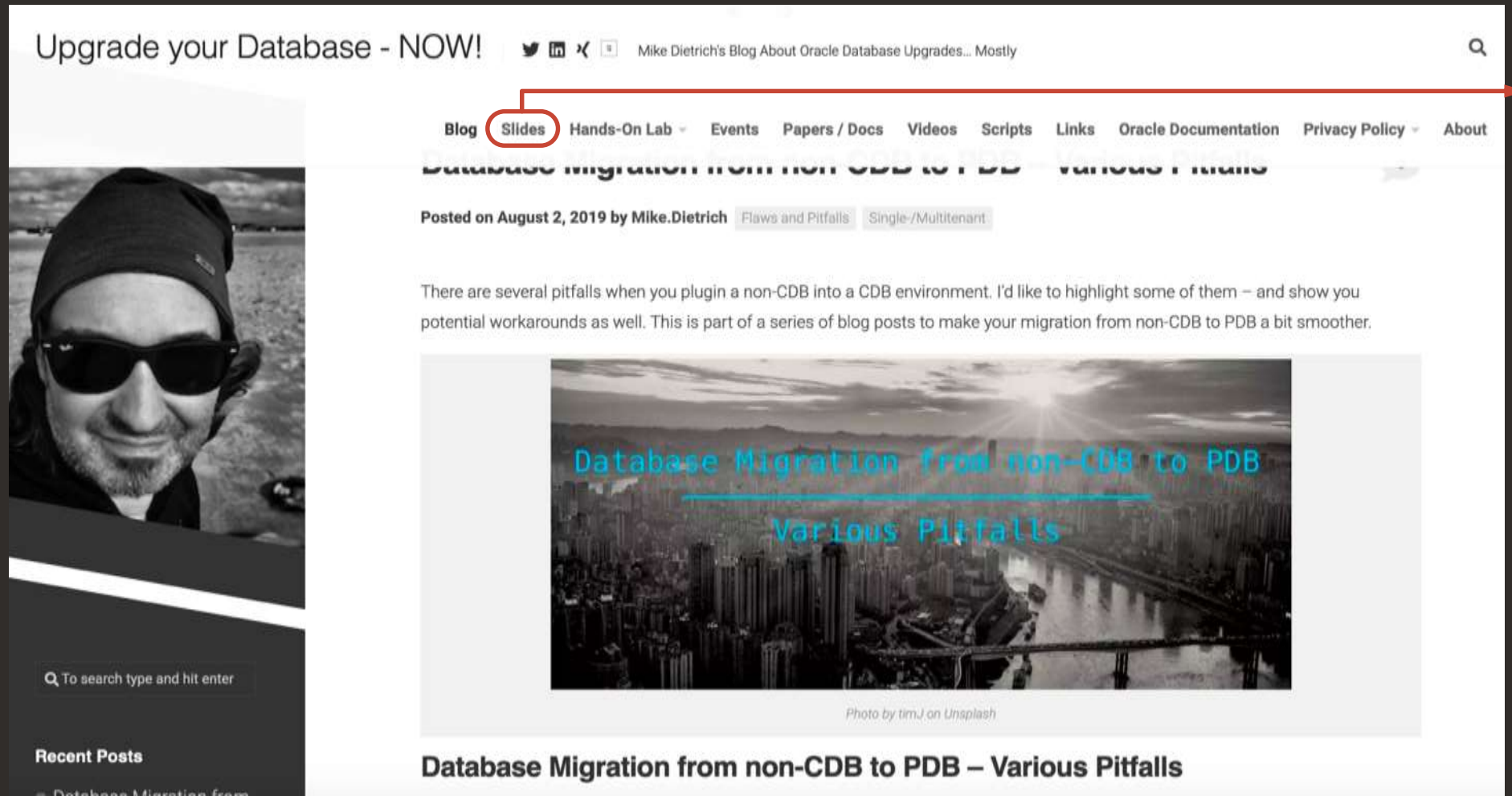
Database Upgrade
and Migrations

https://MikeDietrichDE.com

@MikeDietrichDE

mikedietrich

Slides | https://MikeDietrichDE.com



Daniel Overby Hansen

Senior Principal Product Manager
Cloud Migration

https://dohdatabase.com

@dohdatabase

dohdatabase

Cloud | https://dohdatabase.com/

Blog Categories About

Upgrading in the cloud – VM DB Systems – 11.2.0.4 to 19c (minimal downtime)

Follow Blog via Email

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

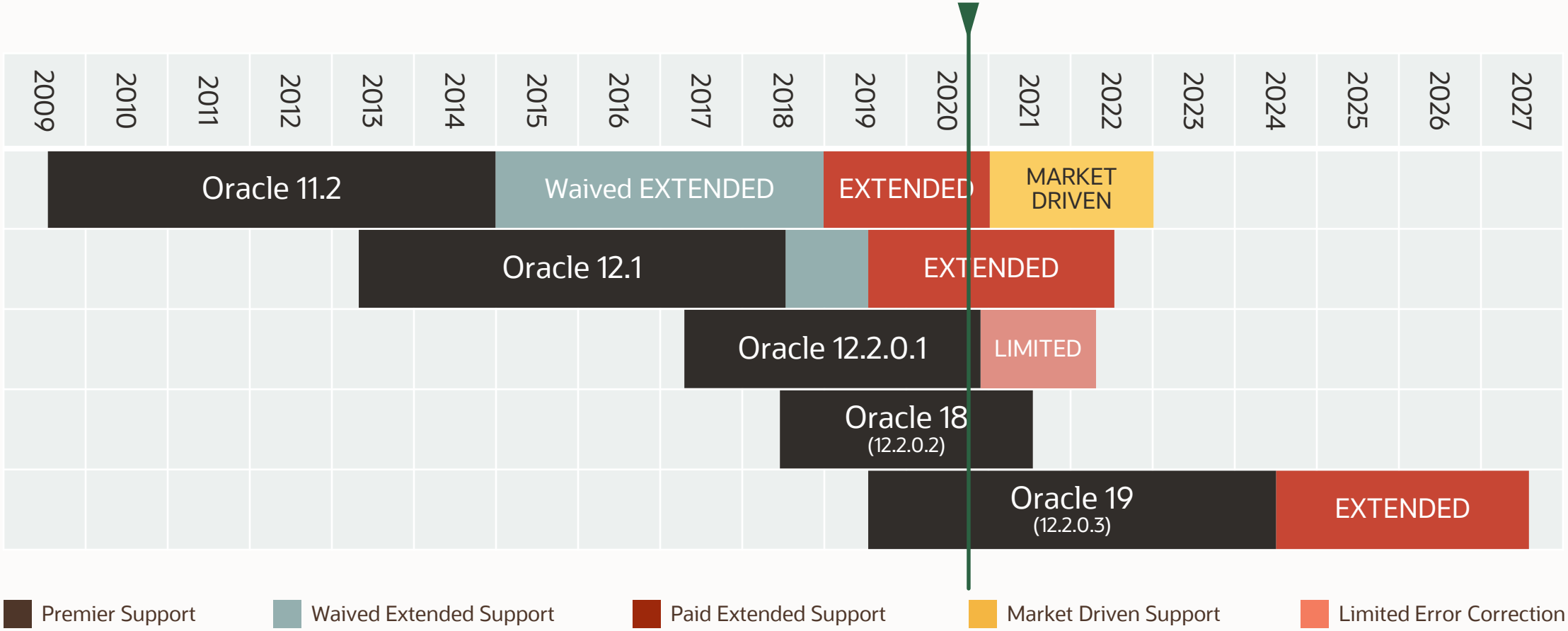
Follow

Photo by SpaceX on Unsplash

Database Migration

Why you "want" to migrate

Lifetime Support Policy



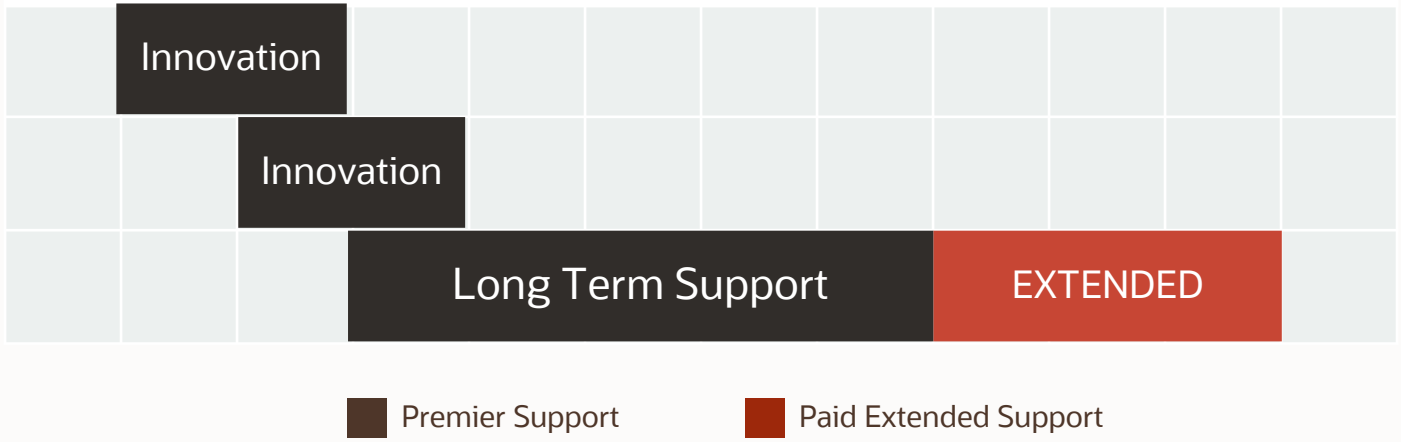
Release Types | Long Term Support vs Innovation Releases

Long Term Support Release

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

Innovation Release

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

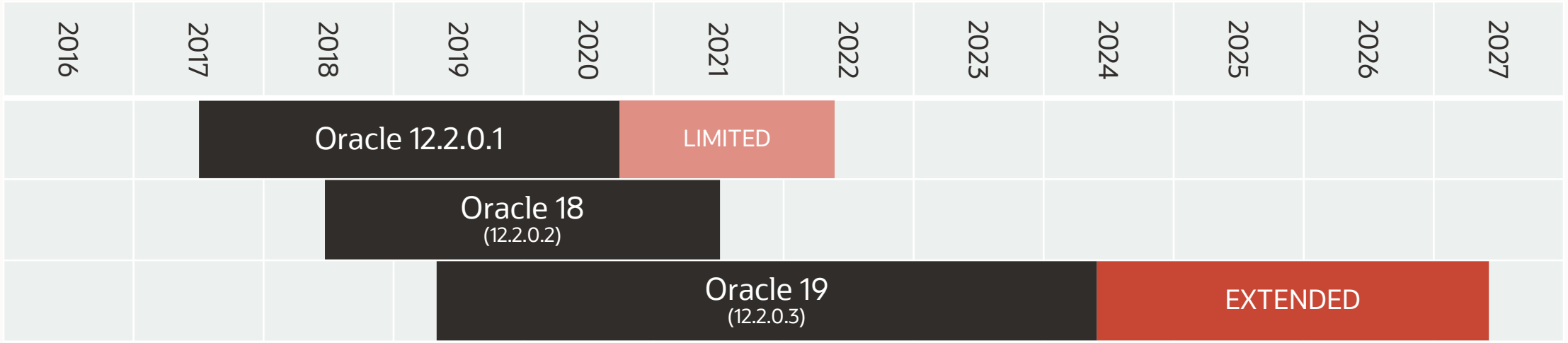


- MOS Note: 742060.1 - Release Schedule of Current Database Releases

Oracle 12.2 Release Family

Includes:

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



- MOS Note:742060.1 - The Single Source of Truth
- MOS Note:161818.1 - Releases Support Status Summary

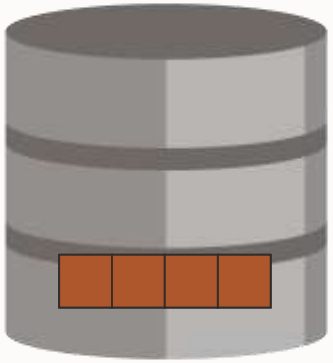
Photo by Jackman Chiu on Unsplash

Migration Strategies

Various Techniques

What do we call a Migration?

Migration | Move data from A to B



Which one is the best technique?

Migration | Techniques

Techniques include:

- Data Pump
- Transportable Tablespaces
- Full Transportable Export/Import
- Data Guard
- Incremental Backups
- Oracle GoldenGate

We will give you a detailed overview!

Migration Strategies

Data Pump

Data Pump

Advantages

- Ease of use
- Universal
- Change structures, character set, and much more
- Platform independent
- Architecture independent
- Works across versions
- Backwards compatible

Consideration

- Duration for large amounts of data and complex structures

Documentation

- Oracle Database 19c Utilities Guide

Data Pump

Setup

Data Pump | Setup Tasks

Directory

```
$ mkdir /home/oracle/dp

SQL> create directory DP_DIR as '/home/oracle/dp';

SQL> select OWNER, DIRECTORY_NAME, DIRECTORY_PATH from DBA_DIRECTORIES;

OWNER           DIRECTORY_NAME              DIRECTORY_PATH
--------------- --------------------------- ----------------------------------------
SYS             DP_DIR                      /home/oracle/dp
SYS             DATA_PUMP_DIR               /u01/app/oracle/admin/UPGR/dpdump/
```

Permissions

```
SQL> grant read, write on directory DP_DIR to SYSTEM;
```


Data Pump | Setup Tasks

Set `STREAMS_POOL_SIZE` to a reasonable value

- 64MB ⇠⇢ 256MB

```
SQL> alter system set STREAMS_POOL_SIZE=128M scope=both;
```


Data Pump

General Best Practices

Data Pump | Dump File

Data Pump | Parameter File

Always use a par file

- Sounds odd but many issues result from people typing complex long Data Pump commands on the command line
- Example:

```
DIRECTORY=DP_DIR
DUMPFILE=dumpfile.dmp
LOGFILE=logfile.log
SCHEMAS=TPCC
EXCLUDE=STATISTICS
LOGTIME=ALL
METRICS=YES
FLASHBACK_TIME=SYSTIMESTAMP
```

```
expdp parfile=your_parfile.par
```


Data Pump | Consistency

Consistent Data Pump export

- `FLASHBACK_SCN=<scn>`
- `FLASHBACK_TIME=SYSTIMESTAMP`
  - Since Oracle 11.2 - Legacy Interface:
    - `CONSISTENT=Y`
  - This will increase UNDO requirements for the duration of the export

Data Pump | Dictionary Statistics

Current dictionary statistics are important

Speeds up large exports up to 60%

```
DBMS_STATS.GATHER_SCHEMA_STATS('SYS')
DBMS_STATS.GATHER_SCHEMA_STATS('SYSTEM')
```


Data Pump | Standard Parameters and Statistics

Always use during export:

- `EXCLUDE=STATISTICS`
  - Either recreate fresh statistics or use a DBMS_STATS staging table for stats migration
- `LOGTIME=ALL`
  - Since Oracle Database 12.1
- `METRICS=YES`

Always use during import:

- `LOGTIME=ALL`
  - Since Oracle Database 12.1
- `METRICS=YES`

```
DBMS_STATS.CREATE_STAT_TABLE
DBMS_STATS.EXPORT_DATABASE_STATS
expdp … TABLES=my_stats
impdp … TABLES=my_stats
DBMS_STATS.IMPORT_DATABASE_STATS
```


Data Pump | Monitoring - METRICS=YES

Without Monitoring

```
Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
Master table "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_SCHEMA_01":  system/******** parfile=imp.par
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "TPCC"."ORDER_LINE"                         100.2 MB 1556919 rows
. . imported "TPCC"."STOCK"                              88.64 MB  300000 rows
. . imported "TPCC"."CUSTOMER"                           47.97 MB   90000 rows
. . imported "TPCC"."HISTORY"                            7.444 MB  155395 rows
. . imported "TPCC"."ITEM"                               7.231 MB  100000 rows
. . imported "TPCC"."ORDERS"                             5.075 MB  155584 rows
. . imported "TPCC"."NEW_ORDER"                          397.4 KB   28669 rows
. . imported "TPCC"."DISTRICT"                           11.92 KB      30 rows
. . imported "TPCC"."WAREHOUSE"                          8.578 KB       3 rows
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/POST_SCHEMA/PROCACT_SCHEMA
Job "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully completed at Mon Sep 28 17:48:28 2020 elapsed 0 00:00:09
```

With METRICS=YES

```
Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
W-1 Startup took 0 seconds
W-1 Master table "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_SCHEMA_01":  system/******** parfile=imp.par
W-1 Processing object type SCHEMA_EXPORT/USER
W-1      Completed 1 USER objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
W-1      Completed 1 SYSTEM_GRANT objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/ROLE_GRANT
W-1      Completed 2 ROLE_GRANT objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
W-1      Completed 1 DEFAULT_ROLE objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
W-1      Completed 1 TABLESPACE_QUOTA objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
W-1      Completed 1 PROCACT_SCHEMA objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/TABLE/TABLE
W-1      Completed 9 TABLE objects in 1 seconds
W-1 Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
W-1 . . imported "TPCC"."ORDER_LINE"                         100.2 MB 1556919 rows in 4 seconds using direct_path
W-1 . . imported "TPCC"."STOCK"                              88.64 MB  300000 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."CUSTOMER"                           47.97 MB   90000 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."HISTORY"                            7.444 MB  155395 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."ITEM"                               7.231 MB  100000 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."ORDERS"                             5.075 MB  155584 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."NEW_ORDER"                          397.4 KB   28669 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."DISTRICT"                           11.92 KB      30 rows in 0 seconds using direct_path
W-1 . . imported "TPCC"."WAREHOUSE"                          8.578 KB       3 rows in 0 seconds using direct_path
W-1 Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
W-1      Completed 5 PROCEDURE objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
W-1      Completed 5 ALTER_PROCEDURE objects in 0 seconds
W-1 Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
W-1      Completed 8 INDEX objects in 1 seconds
W-1 Processing object type SCHEMA_EXPORT/POST_SCHEMA/PROCACT_SCHEMA
W-1      Completed 1 PROCACT_SCHEMA objects in 0 seconds
W-1      Completed 9 SCHEMA_EXPORT/TABLE/TABLE_DATA objects in 4 seconds
Job "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully completed at Mon Sep 28 17:49:43 2020 elapsed 0 00:00:09
```

Data Pump | Diagnostics

Only with LOGTIME=ALL

```
Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
28-SEP-20 17:59:53.714: Master table "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded
28-SEP-20 17:59:53.934: Starting "SYSTEM"."SYS_IMPORT_SCHEMA_01":  system/******** parfile=imp.par
28-SEP-20 17:59:53.971: Processing object type SCHEMA_EXPORT/USER
28-SEP-20 17:59:54.142: Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
28-SEP-20 17:59:54.220: Processing object type SCHEMA_EXPORT/ROLE_GRANT
28-SEP-20 17:59:54.303: Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
28-SEP-20 17:59:54.380: Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
28-SEP-20 17:59:54.455: Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
28-SEP-20 17:59:54.643: Processing object type SCHEMA_EXPORT/TABLE/TABLE
28-SEP-20 17:59:55.554: Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
28-SEP-20 17:59:58.643: . . imported "TPCC"."ORDER_LINE"                  100.2 MB 1556919 rows
28-SEP-20 17:59:58.819: . . imported "TPCC"."STOCK"                      88.64 MB  300000 rows
28-SEP-20 17:59:58.965: . . imported "TPCC"."CUSTOMER"                   47.97 MB   90000 rows
28-SEP-20 17:59:59.022: . . imported "TPCC"."HISTORY"                    7.444 MB  155395 rows
28-SEP-20 17:59:59.085: . . imported "TPCC"."ITEM"                       7.231 MB  100000 rows
28-SEP-20 17:59:59.145: . . imported "TPCC"."ORDERS"                     5.075 MB  155584 rows
28-SEP-20 17:59:59.183: . . imported "TPCC"."NEW_ORDER"                  397.4 KB   28669 rows
28-SEP-20 17:59:59.203: . . imported "TPCC"."DISTRICT"                   11.92 KB      30 rows
28-SEP-20 17:59:59.220: . . imported "TPCC"."WAREHOUSE"                  8.578 KB       3 rows
28-SEP-20 17:59:59.251: Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
28-SEP-20 17:59:59.366: Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
28-SEP-20 17:59:59.643: Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
28-SEP-20 18:00:00.775: Processing object type SCHEMA_EXPORT/POST_SCHEMA/PROCACT_SCHEMA
28-SEP-20 18:00:00.957: Job "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully completed at Mon Sep 28 18:00:00
 2020 elapsed 0 00:00:00
```

With METRICS=YES and LOGTIME=ALL

```
Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
28-SEP-20 17:50:48.328: W-1      Startup took 0 seconds
28-SEP-20 17:50:48.932: W-1 Master table "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded
28-SEP-20 17:50:49.145: Starting "SYSTEM"."SYS_IMPORT_SCHEMA_01":  system/******** parfile=imp.par
28-SEP-20 17:50:49.181: W-1 Processing object type SCHEMA_EXPORT/USER
28-SEP-20 17:50:49.359: W-1      Completed 1 USER objects in 0 seconds
28-SEP-20 17:50:49.359: W-1 Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
28-SEP-20 17:50:49.449: W-1      Completed 1 SYSTEM_GRANT objects in 0 seconds
28-SEP-20 17:50:49.449: W-1 Processing object type SCHEMA_EXPORT/ROLE_GRANT
28-SEP-20 17:50:49.547: W-1      Completed 2 ROLE_GRANT objects in 0 seconds
28-SEP-20 17:50:49.547: W-1 Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
28-SEP-20 17:50:49.629: W-1      Completed 1 DEFAULT_ROLE objects in 0 seconds
28-SEP-20 17:50:49.629: W-1 Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
28-SEP-20 17:50:49.705: W-1      Completed 1 TABLESPACE_QUOTA objects in 0 seconds
28-SEP-20 17:50:49.705: W-1 Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
28-SEP-20 17:50:49.890: W-1      Completed 1 PROCACT_SCHEMA objects in 0 seconds
28-SEP-20 17:50:49.890: W-1 Processing object type SCHEMA_EXPORT/TABLE/TABLE
28-SEP-20 17:50:50.865: W-1      Completed 9 TABLE objects in 1 seconds
28-SEP-20 17:50:50.887: W-1 Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
28-SEP-20 17:50:54.222: W-1      . imported "TPCC"."ORDER_LINE"                  100.2 MB 1556919 rows in 4 seconds using direct_path
28-SEP-20 17:50:54.407: W-1      . imported "TPCC"."STOCK"                       88.64 MB  300000 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.544: W-1      . imported "TPCC"."CUSTOMER"                    47.97 MB   90000 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.628: W-1      . imported "TPCC"."HISTORY"                     7.444 MB  155395 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.692: W-1      . imported "TPCC"."ITEM"                        7.231 MB  100000 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.761: W-1      . imported "TPCC"."ORDERS"                      5.075 MB  155584 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.807: W-1      . imported "TPCC"."NEW_ORDER"                   397.4 KB   28669 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.829: W-1      . imported "TPCC"."DISTRICT"                    11.92 KB      30 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.851: W-1      . imported "TPCC"."WAREHOUSE"                   8.578 KB       3 rows in 0 seconds using direct_path
28-SEP-20 17:50:54.888: W-1 Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
28-SEP-20 17:50:55.007: W-1      Completed 5 PROCEDURE objects in 0 seconds
28-SEP-20 17:50:55.007: W-1 Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
28-SEP-20 17:50:55.239: W-1      Completed 5 ALTER_PROCEDURE objects in 0 seconds
28-SEP-20 17:50:55.307: W-1 Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
28-SEP-20 17:50:56.544: W-1      Completed 8 INDEX objects in 1 seconds
28-SEP-20 17:50:56.544: W-1 Processing object type SCHEMA_EXPORT/POST_SCHEMA/PROCACT_SCHEMA
28-SEP-20 17:50:56.635: W-1      Completed 1 PROCACT_SCHEMA objects in 0 seconds
28-SEP-20 17:50:56.678: W-1      Completed 9 SCHEMA_EXPORT/TABLE/TABLE_DATA objects in 4 seconds
28-SEP-20 17:50:56.719: Job "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully completed at Mon Sep 28 17:50:56 2020 elapsed 0 00:00:09
```

Data Pump | Parallelism

Parallelism

- You must set `PARALLEL=<n>` manually
  - Typically *n = 2x <number of CPU cores>*
  - `PARALLEL` defines also how many indexes get created in parallel
  - If you don't set it, only 1 worker (W-1) will do all the work

- New feature since Oracle 12.2:
  Parallel Export/Import of Metadata
  - But not with Transportable Tablespaces, Full Transportable Export/Import and over NETWORK_LINK

Data Pump | Parallelism – Only for imports to 11.2 and 12.1

Parallelism for import into 11.2.0.4 / 12.1.0.2

- Apply patch for bug 22273229 to enable parallel import of constraints/indexes

Data Pump | LOBs

BasicFile (old) LOBs are always slow

SecureFile LOBs can be fast

- Especially in conjunction with partitioning
- Use DBMS_REDEFINITION or ONLINE TABLE MOVE (18c) to convert BasicFile into SecureFile LOBs
- `LOB_STORAGE=SECUREFILE` during `impdp` to convert old LOBs into SecureFile LOBs

Data Pump

Useful Features

Data Pump | Generate SQL Statements from Dumpfile with SQLFILE

Parfile example

```
DIRECTORY=DP_DIR
DUMPFILE=dumpfile.dmp
LOGFILE=logfile.log
SCHEMAS=TPCC
SQLFILE=all_statements.sql
```

Impdp call

```
impdp system/oracle parfile=impsql.par
```

```
--CONNECT SYSTEM
ALTER SESSION SET EVENTS '10150 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10904 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '25475 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10407 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10851 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '22830 TRACE NAME CONTEXT FOREVER, LEVEL 192 ';
-- new object type path: SCHEMA_EXPORT/USER
 CREATE USER "TPCC" IDENTIFIED BY VALUES
'S:F9E9DD2D0A8D0AEA2ACB9000FD1EDE144005661F7A9AE2BD6951DE396931;BB4954843B02D85D'
      DEFAULT TABLESPACE "TPCCTAB"
      TEMPORARY TABLESPACE "TEMP";
-- new object type path: SCHEMA_EXPORT/SYSTEM_GRANT
GRANT UNLIMITED TABLESPACE TO "TPCC";
-- new object type path: SCHEMA_EXPORT/ROLE_GRANT
 GRANT "CONNECT" TO "TPCC";
 GRANT "RESOURCE" TO "TPCC";
-- new object type path: SCHEMA_EXPORT/DEFAULT_ROLE
 ALTER USER "TPCC" DEFAULT ROLE ALL;
-- new object type path: SCHEMA_EXPORT/TABLESPACE_QUOTA
DECLARE
  TEMP_COUNT NUMBER;
  SQLSTR VARCHAR2(200);
BEGIN
  SQLSTR := 'ALTER USER "TPCC" QUOTA UNLIMITED ON "TPCCTAB"';
  EXECUTE IMMEDIATE SQLSTR;
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE = -30041 THEN
      SQLSTR := 'SELECT COUNT(*) FROM USER_TABLESPACES
              WHERE TABLESPACE_NAME = ''TPCCTAB'' AND CONTENTS = ''TEMPORARY''';
      EXECUTE IMMEDIATE SQLSTR INTO TEMP_COUNT;
      IF TEMP_COUNT = 1 THEN RETURN;
      ELSE RAISE;
      END IF;
    ELSE
      RAISE;
    END IF;
END;
/
```

Data Pump | Generate DBMS_DATAPUMP with Event 10046

Set in a test database

```
alter system set event='10046 trace name context forever, level 4';
```

Run `expdp` or `impdp`

Run the trace file through `tkprof`

```
$ORACLE_BASE/diag/rdbms/sid/SID/trace/SID_ora_12345.trc
```

```
tkprof SID_ora_12345.trc out.txt
```

Examine `DBMS_DATAPUMP` calls, e.g.

```
BEGIN
  SYS.DBMS_DATAPUMP.SET_PARALLEL(handle => :JOBHNDL, degree => :DEGREE);
END;
```

Data Pump | Network Mode

Data Pump | Network Link

NETWORK_LINK Parameter

- impdp over a database link
- expdp on source side implicitly used
- No dumpfile generated
  - Helpful when you have no access to file system
- Does not work for downgrades

Limitations:

- No parallel metadata support yet
- No LONG and RAW data
  - This works when source DB is Oracle 12.2 or newer
  - ACCESS_METHOD=DIRECT_PATH
- Data parallelism is restricted to multiple partitions or tables
  - There is no PQ parallelism within a large, unpartitioned table over a dblink

Data Pump | Network Link - Example

Create database link

```
SQL> create database link V11204 connect to
SYSTEM identified by pwd using 'UPGR';
```

Parfile

```
DIRECTORY=DP_DIR
LOGFILE=logfile.log
SCHEMAS=TPCC
NETWORK_LINK=V11204
```

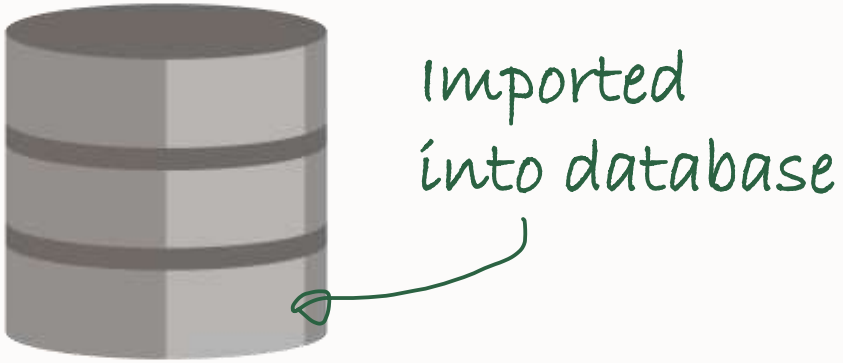
Data Pump | VERSION Parameter

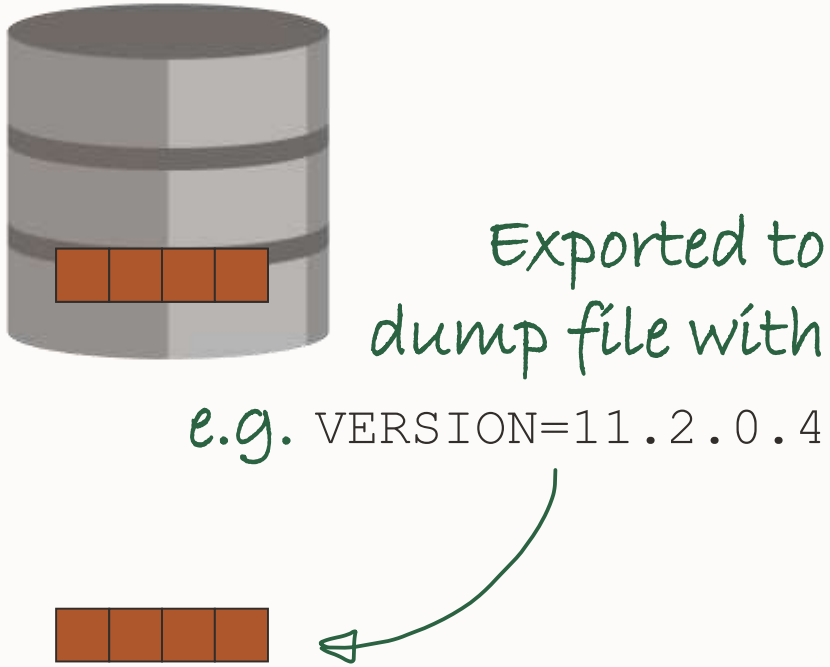
MOS Note: 553337.1
Export/Import DataPump Parameter VERSION - Compatibility of Data Pump Between Different Oracle Versions

```
VERSION = [ COMPATIBLE | LATEST | version_string ]
```

Most useful only in cases where you migrate data to a lower database version

Fallback | Data Pump with dumpfile





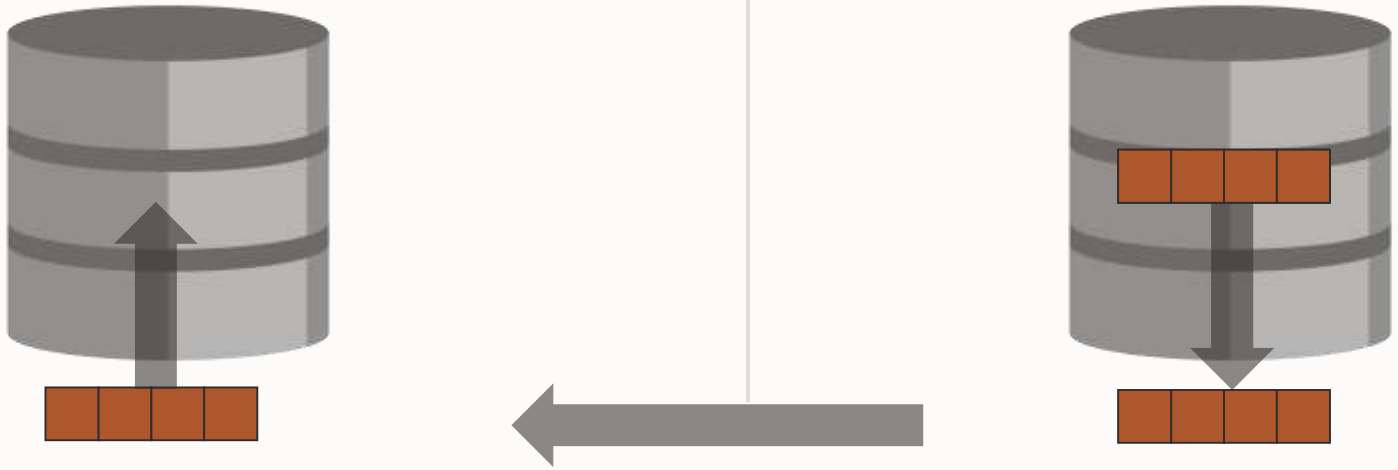
Copied over
the network

Data Pump | VERSION Parameter for Fallbacks

Create dumpfile in format of <VERSION>

Parfile example

```
DIRECTORY=DP_DIR
DUMPFILE=downgrade.dmp
LOGFILE=downgrade.log
FULL=YES
EXCLUDE=STATISTICS
VERSION=11.2.0.4
```



Data Pump | Test Mode for Transportable Tablespaces

NEW IN 19c

Test Transportable Tablespaces export or FTEX without "read only"

- `TTS_CLOSURE_CHECK = ON | OFF | FULL | TEST_MODE`

Preview a TTS/FTEX export

- Additionally: Specify the degree of closure checking
- `ON` : Self-containment closure check
- `OFF`: No closure check
- `FULL`: Bi-directional closure check (`TTS_FULL_CHECK` & `TRANSPORT_FULL_CHECK` parameters are still supported)
- `TEST_MODE`: Tablespaces are not required to be read-only

Data Pump | Test Mode for Transportable Tablespaces

Works from Oracle 19c on

Export parfile

```
DIRECTORY=DP_DIR
DUMPFILE=tts.dmp
LOGFILE=logfile.log
TTS_CLOSURE_CHECK=TEST_MODE
TRANSPORT_TABLESPACES=(TTS)
```

There is no TTS import possible

- **Dump file set is unusable. TEST_MODE requested**

```
$ expdp system/oracle@pdb1 parfile=tts.par

Export: Release 19.0.0.0.0 - Production on Wed Sep 30 21:59:23 2020
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.


Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********@pdb1
parfile=tts.par
 Startup took 0 seconds
 Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
      Completed 3 TABLE_STATISTICS objects in 0 seconds
 Processing object type TRANSPORTABLE_EXPORT/STATISTICS/MARKER
      Completed 1 MARKER objects in 4 seconds
 Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
      Completed 1 PLUGTS_BLK objects in 0 seconds
 Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
      Completed 1 PLUGTS_BLK objects in 0 seconds
 Processing object type TRANSPORTABLE_EXPORT/TABLE
      Completed 3 TABLE objects in 2 seconds
 Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully
loaded/unloaded
 ******************************************************************************
 Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
   /u01/app/oracle/admin/CDB2/dpdump/B08E2264E5651243E055000000000001/tts.dmp
 Dump file set is unusable. TEST_MODE requested.
 ******************************************************************************
 Datafiles required for transportable tablespace TTS:
   /u02/oradata/CDB2/pdb1/tts01.dbf
 Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Wed Sep
30 21:59:46 2020 elapsed 0 00:00:20
```

Data Pump | Keep Tablespaces Read-Only for TTS

NEW IN 19c

Allow "read only" tablespaces for TTS import

- `TRANSPORTABLE=NEVER|ALWAYS|KEEP_READ_ONLY|NO_BITMAP_REBUILD`

By default, a tablespace gets switched to read/write once transport has completed

Usage:

- Repeat the TTS operation without need to restore files again
- Share tablespaces between several databases

Data Pump | Keep Tablespaces Read-Only for TTS

Works from Oracle 19c on

Import parfile

```
DIRECTORY=DP_DIR
DUMPFILE=tts.dmp
LOGFILE=logfile.log
TRANSPORT_DATAFILES='/CDB2/pdb1/tts.dbf'
TRANSPORTABLE=KEEP_READ_ONLY
```

Tablespaces kept read-only

```
TABLESPACE_NAME    STATUS
------------------ ----------
SYSTEM             ONLINE
SYSAUX             ONLINE
UNDOTBS1           ONLINE
TEMP               ONLINE
TTS                READ ONLY
```

```
$ impdp parfile=imptts.par

Import: Release 19.0.0.0.0 - Production on Wed Sep 30 23:21:36 2020
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully
loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********@pdb1
parfile=imptts.par
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/MARKER
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Wed Sep 30
23:21:57 2020 elapsed 0 00:00:19
```

Data Pump | Remove Column Encryption

NEW IN 19c

Remove column encryption during import

- `TRANSFORM=OMIT_ENCRYPTION_CLAUSE:[Y|N]`
- Usage:
  - Migrate to Oracle Cloud when the source DB has encrypted columns
- Example:
  - `impdp hr DIRECTORY=dpump_dir1 DUMPFILE=hr.dmp SCHEMAS=hr TRANSFORM=OMIT_ENCRYPTION_CLAUSE:Y`
- Details:
  - Valid for TABLE object types
  - `Y`: Encrypted columns in source won't be encrypted in imported tables
  - `N`: column encryption clauses created as in source

Data Pump

Advanced Features

Data Pump | Control Parallel Executions for PDBs

`MAX_DATAPUMP_JOBS_PER_PDB` = { integer | AUTO }

- Default: 100 - `AUTO`: 50% of SESSIONS
- Per PDB: Maximum number of concurrent DP jobs

```
SQL> alter system set MAX_DATAPUMP_JOBS_PER_PDB=2 container=all;
```

  - Starting a 3rd Data Pump job then results in: **ORA-39391: maximum number of Data Pump jobs (2) exceeded**

`MAX_DATAPUMP_PARALLEL_PER_JOB` = { integer | AUTO }

NEWIN 19c

- Default: 50 - `AUTO`: 50% of SESSIONS
- Per PDB: Maximum number of parallel processes per DP job

```
SQL> alter system set MAX_DATAPUMP_PARALLEL_PER_JOB=1 container=all;
```

  - Only 1 worker (W-1) will be started and visible

Data Pump | Gain Speed with Compression

Speed up large exports drastically with COMPRESSION_ALGORITHM

- You will need an Advanced Compression Option license

COMPRESSION_ALGORITHM

- Defines the compression algorithm when compressing dump files
  - BASIC The same algorithm used in previous versions. Good compression, without severely impacting on performance
  - LOW : For use when reduced CPU utilization is a priority over compression ratio
  - MEDIUM: Recommended option. Similar characteristics to BASIC, but uses a different algorithm
  - HIGH: Maximum available compression, but more CPU intensive

- Performance:
  - Compression ratio
  - CPU usage

```
$ expdp scott/tiger tables=emp directory=mydir
dumpfile=emp.dmp logfile=expdp_emp.log
compression=ALL compression_algorithm=MEDIUM
```

Data Pump | Compression Algorithm Example

Customer evaluation

- `BASIC`
  at 3.5 TB/hour

- `MEDIUM`
  at 7.0 TB/hour

Disk-Group-I/O

Name	Disks	AvgBusy	Read|Write-KB/s	TotalMB/s	xfers/s	BlockSizeKB
slot02	6	9.3%	123120.4|0.0	120.2	241.1	510.7
slot03	6	6.7%	103354.8|0.0	100.9	202.2	511.1
slot05	6	9.0%	130420.9|7.0	127.4	262.0	497.8
slot06	6	10.5%	158841.9|175.3	155.3	329.3	
slot08	6	8.4%	130835.3|0.0	127.8	256.0	511.0
slot09	6	10.1%	136525.9|0.0	133.3	267.0	511.3
slot10	6	6.6%	140383.4|0.0	137.1	275.0	510.6
slot11	6	6.8%	112600.0|2.0	110.0	220.7	510.3
Groups= 8 TOTALS	48	1.4%	1036082.5|184.3	1012.0	2053.3	

Disk-Group-I/O

Name	Disks	AvgBusy	Read|Write-KB/s	TotalMB/s	xfers/s	BlockSizeKB
slot02	6	14.5%	255770.4|0.0	249.8	500.9	510.7
slot03	6	16.0%	273037.4|11.5	266.6	535.1	510.3
slot05	6	15.4%	264851.1|17.5	258.7	519.0	510.3
slot06	6	13.2%	222160.7|425.5	217.4	502.4	443.1
slot08	6	15.0%	267156.6|1.5	260.9	523.3	510.5
slot09	6	14.8%	263140.4|6.5	257.0	515.3	510.6
slot10	6	14.6%	259603.7|2.5	253.5	508.5	510.5
slot11	6	14.9%	258113.0|5.4	252.1	505.8	510.4
Groups= 8 TOTALS	48	2.5%	2063833.5|470.4	2015.9	4110.285	

2x

Data Pump | Import into existing objects

Slowness when objects exist already

Data Pump uses additional checks

Precreate partitions and import in parallel

- `DATA_OPTION = TRUST_EXISTING_TABLE_PARTITIONS`
- Data Pump will load partition data in parallel into existing tables
  - Useful typically when you have a high number of partitions

Data Pump | Legacy Mode

Introduced in Oracle 11.2

Useful when you have old `exp/imp` parameter files

4 Oracle Data Pump Legacy Mode

With Oracle Data Pump legacy mode, you can use original Export and Import parameters on the Oracle Data Pump Export and Data Pump Import command lines.

Oracle Data Pump Legacy Mode Use Cases
Oracle Data Pump enters legacy mode when it encounters legacy export or import parameters, so that you can continue using existing scripts.

Parameter Mappings
You can use original Export and Import parameters when they map to Oracle Data Pump Export and Import parameters that supply similar functionality.

Management of File Locations in Data Pump Legacy Mode
Original Export and Import and Data Pump Export and Import differ on where dump files and log files can be written to and read from because the original version is client-based and Data Pump is server-based.

Adjusting Existing Scripts for Data Pump Log Files and Errors
Describes how to adjust existing scripts for Data Pump log files and errors.

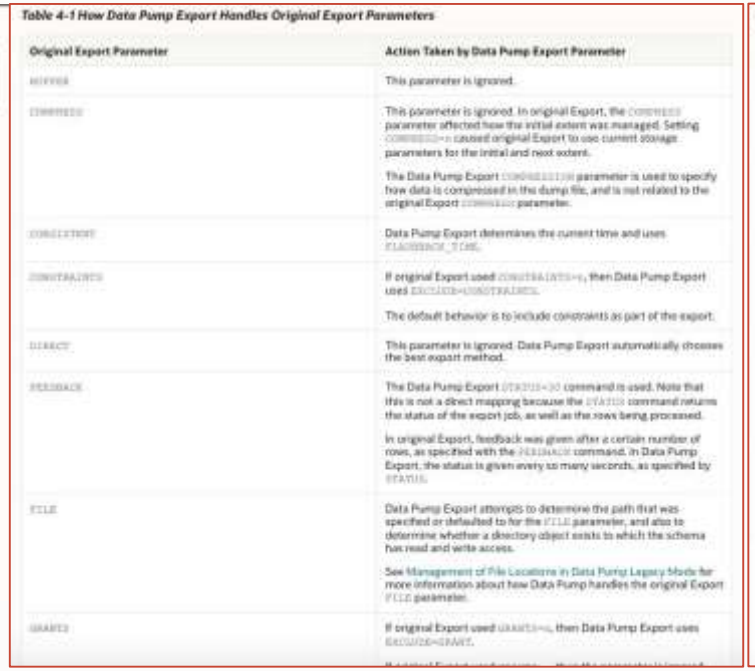


Table 4-1 How Data Pump Export Handles Original Export Parameters

Original Export Parameter	Action Taken by Data Pump Export Parameter
BUFFER	This parameter is ignored.
COMPRESS	This parameter is ignored. In original Export, the COMPRESS parameter affected how the initial extent was managed. Setting COMPRESS=Y caused original Export to use current storage parameters for the initial and next extent. The Data Pump Export COMPRESSION parameter is used to specify how data is compressed in the dump file, and is not related to the original Export COMPRESS parameter.
CONSISTENT	Data Pump Export determines the current time and uses FLASHBACK_TIME.
CONSTRAINTS	If original Export used CONSTRAINTS=n, then Data Pump Export uses EXCLUDE=CONSTRAINTS. The default behavior is to include constraints as part of the export.
DIRECT	This parameter is ignored. Data Pump Export automatically chooses the best export method.
FEEDBACK	The Data Pump Export STATUS=30 command is used. Note that this is not a direct mapping because the STATUS command returns the status of the export job, as well as the rows being processed. In original Export, feedback was given after a certain number of rows, as specified with the FEEDBACK command. In Data Pump Export, the status is given every so many seconds, as specified by STATUS.
FILE	Data Pump Export attempts to determine the path that was specified or defaulted to for the FILE parameter, and also to determine whether a directory object exists to which the schema has read and write access. See Management of File Locations in Data Pump Legacy Mode for more information about how Data Pump handles the original Export FILE parameter.
GRANTS	If original Export used GRANTS=n, then Data Pump Export uses EXCLUDE=GRANT.

Table 4-2 How Data Pump Import Handles Original Import Parameters

Original Import Parameter	Action Taken by Data Pump Import Parameter
BUFFER	This parameter is ignored.
CHARSET	This parameter was desupported several releases ago and should no longer be used. It will cause the Data Pump Import operation to abort.
COMMIT	This parameter is ignored. Data Pump Import automatically performs a commit after each table is processed.
COMPILE	This parameter is ignored. Data Pump Import compiles procedures after they are created. A recompile can be executed if necessary for dependency reasons.
CONSTRAINTS	If original Import used CONSTRAINTS=n, then Data Pump Import uses the EXCLUDE=CONSTRAINT parameter. If original Import used CONSTRAINTS=y, then the parameter is ignored and does not need to be remapped because that is the Data Pump Import default behavior.
DATAFILES	The Data Pump Import TRANSPORT_DATAFILES parameter is used.
DESTROY	If original Import used DESTROY=y, then Data Pump Import uses the REUSE_DATAFILES=y parameter. If original Import used DESTROY=n, then the parameter is ignored and does not need to be remapped because that is the Data Pump Import default behavior.
FEEDBACK	The Data Pump Import STATUS=30 command is used. Note that this is not a direct mapping because the STATUS command returns the status of the import job, as well as the rows being processed. In original Import, feedback was given after a certain number of rows, as specified with the FEEDBACK command. In Data Pump Import, the status is given every so many seconds, as specified by STATUS.
FILE	Data Pump Import attempts to determine the path that was specified or defaulted to for the FILE parameter, and also to determine whether a directory object exists to which the schema has read and write access. See Management of File Locations in Data Pump Legacy Mode for more information about how Data Pump handles the original Import FILE parameter.

[Oracle Database 19c Utilities Guide – Legacy Mode]

Data Pump | Further Information

MOS Note:1264715.1
Master Note for Data Pump

MOS Note:5533337.1
For Compatibility and version changes

MOS Note: 2457955.1
19c Data Pump New Features

The Second Exadata in Europe

Payback GmbH Germany
HP to Exadata Migration Project 2009

https://www.payback.de/ratgeber/besser-leben/payback-tricks

Customer Case | Payback

- Customer
- Project 2009
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Payback GmbH

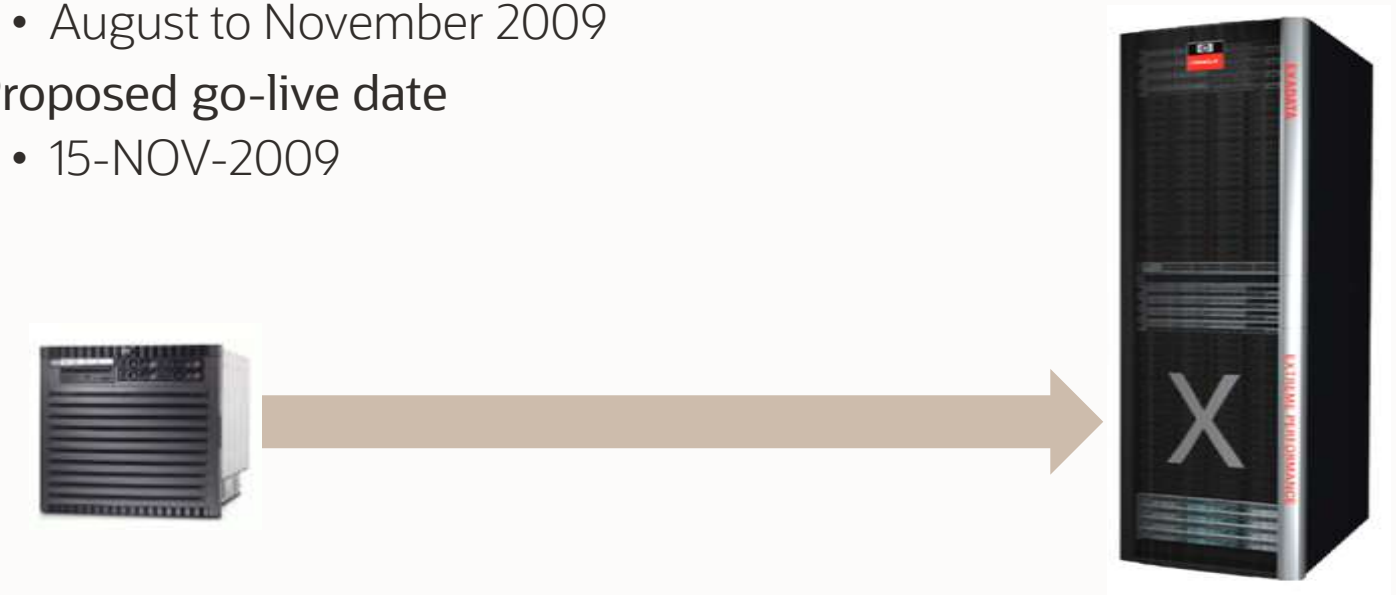
- Belongs to American Express
- HQ in Munich, Germany
- Develops and operates professional customer loyalty programs based on customized IT solutions

Customer Case | Payback

- Customer
- Project 2009
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Migrate 7TB / 1.5TB from HP-UX to Exadata V1

- Cross platform, cross Endianness, cross version
  - Oracle 9.2.0.7 on HP-UX ⇨ Oracle 11.1.0.7 on OL
- 4 months planning and migration phase
  - August to November 2009
- Proposed go-live date
  - 15-NOV-2009



Customer Case | Payback

- Customer
- Project 2009
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Move everything in less than 24 hrs

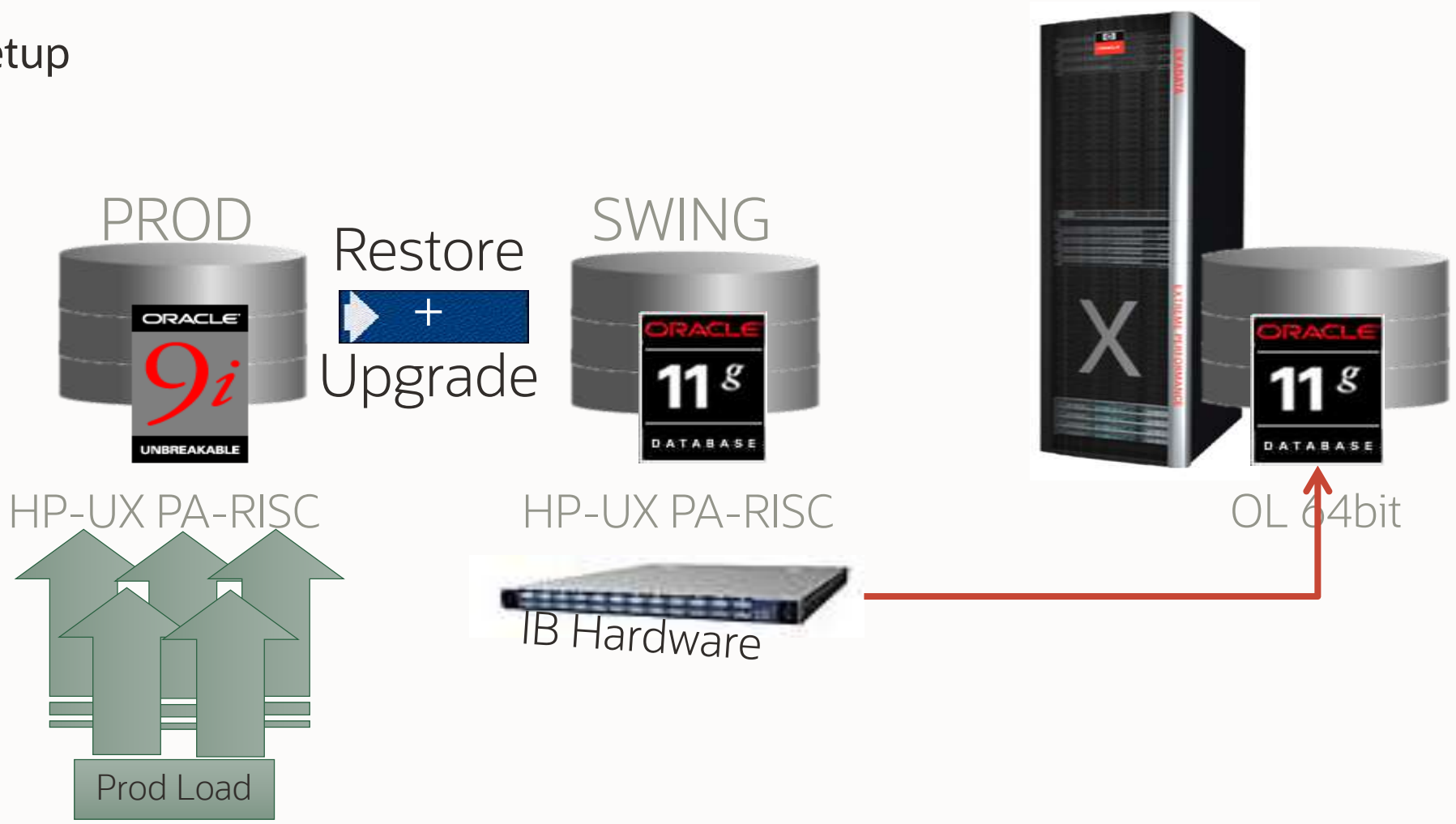
Network bottleneck

- Remedy:
  Install extra InfiniBand hardware into HP box
  ⇨ ~ 3GB/sec throughput!

Customer Case | Payback

- Customer
- Project 2009
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

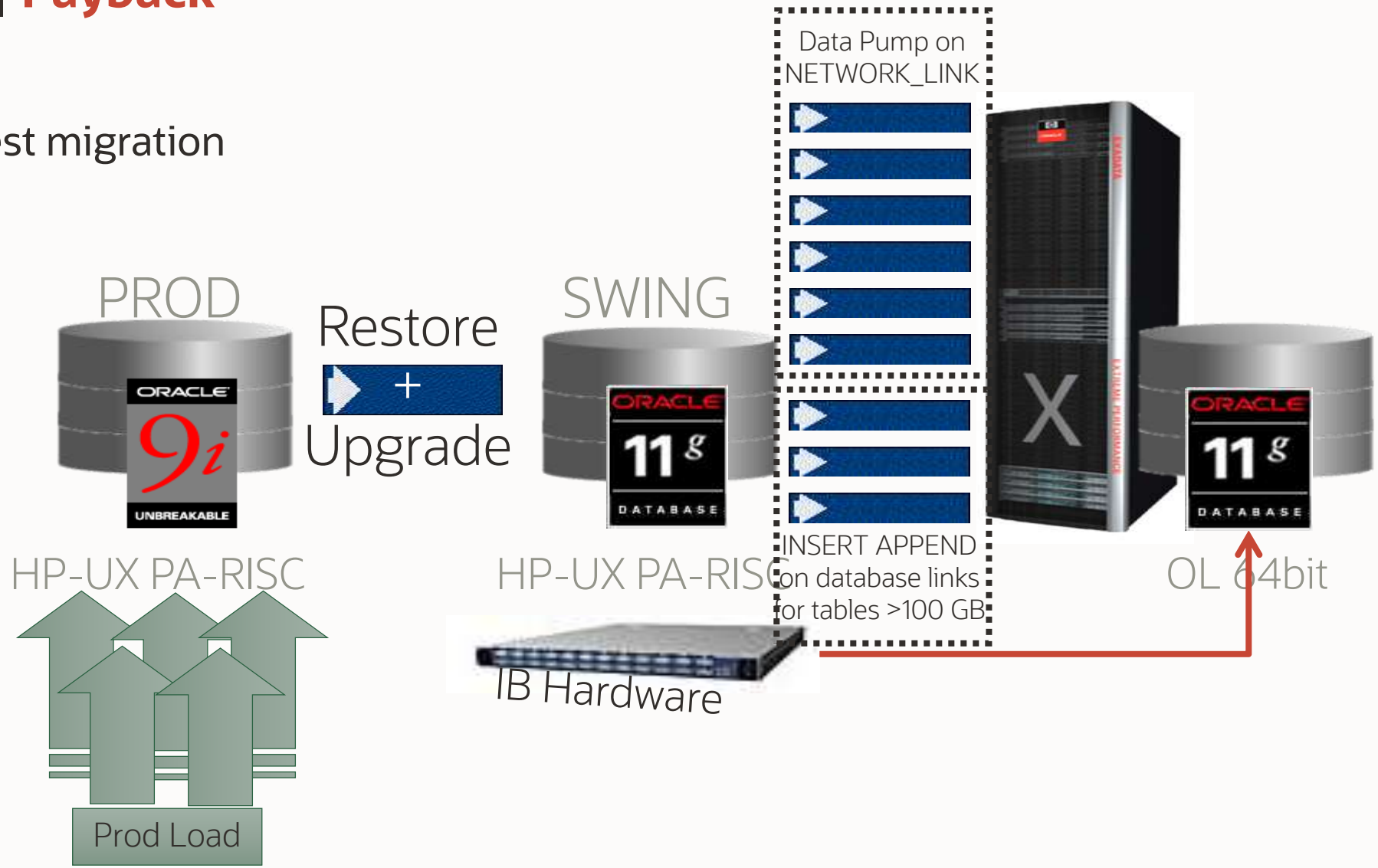
Setup



Customer Case | Payback

- Customer
- Project 2009
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

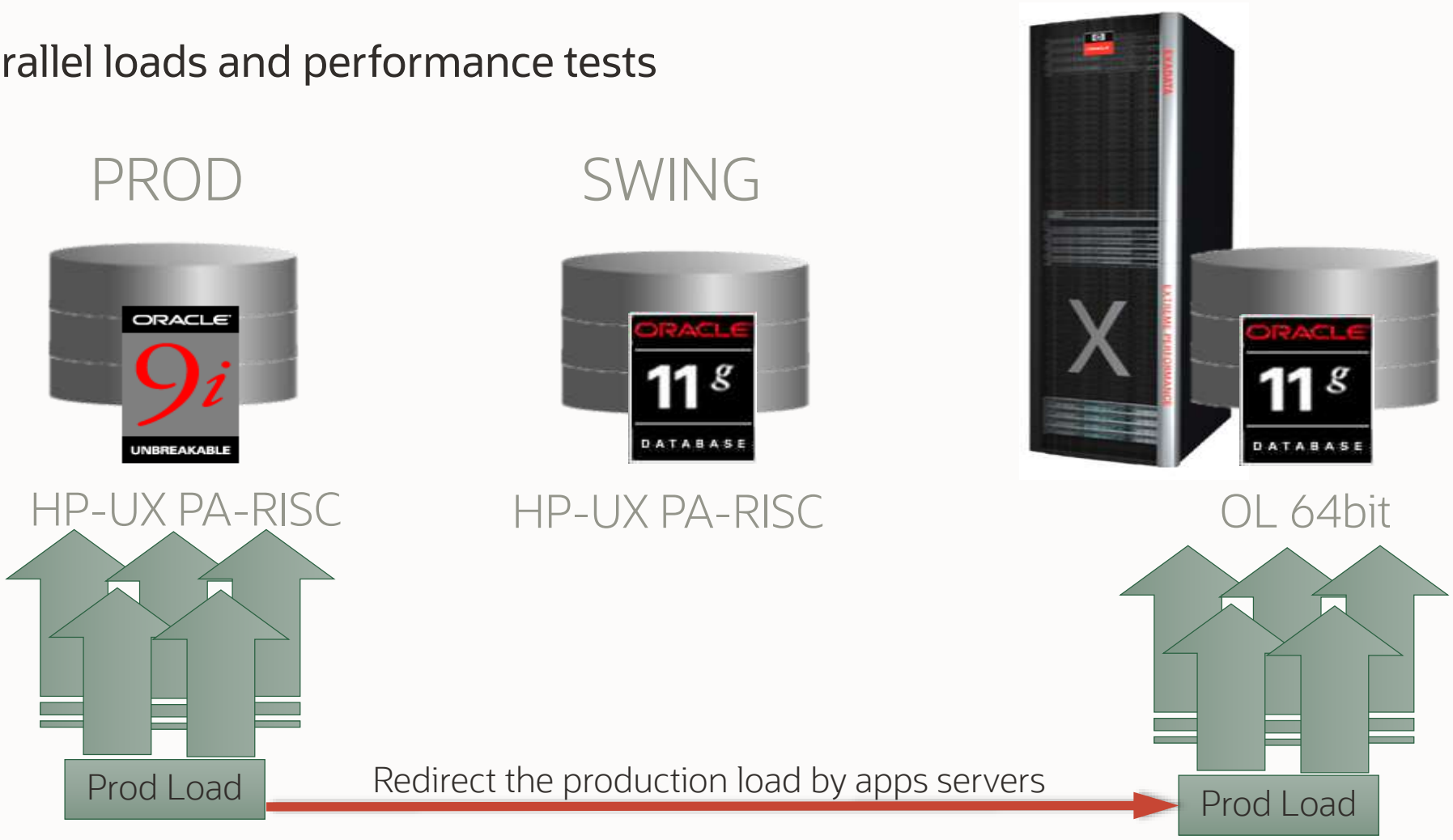
Test migration



Customer Case | Payback

- Customer
- Project 2009
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

Parallel loads and performance tests



Customer Case | Payback

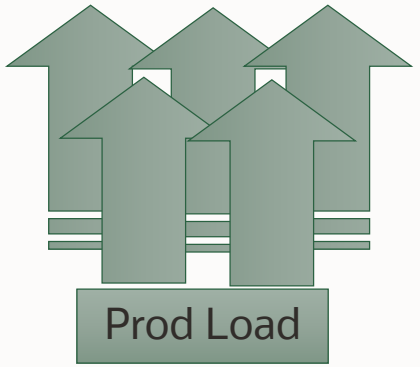
- Customer
- Project 2009
- Constraints
- Preparation
- **Migration**
- Success?
- Remarks

Last test came live migration

HP-UX PA-RISC

PROD

OL 64bit



Customer Case | Payback

- Customer
- Project 2009
- Constraints
- Preparation
- Migration
- **Success?**
- Remarks

Live? And alive?

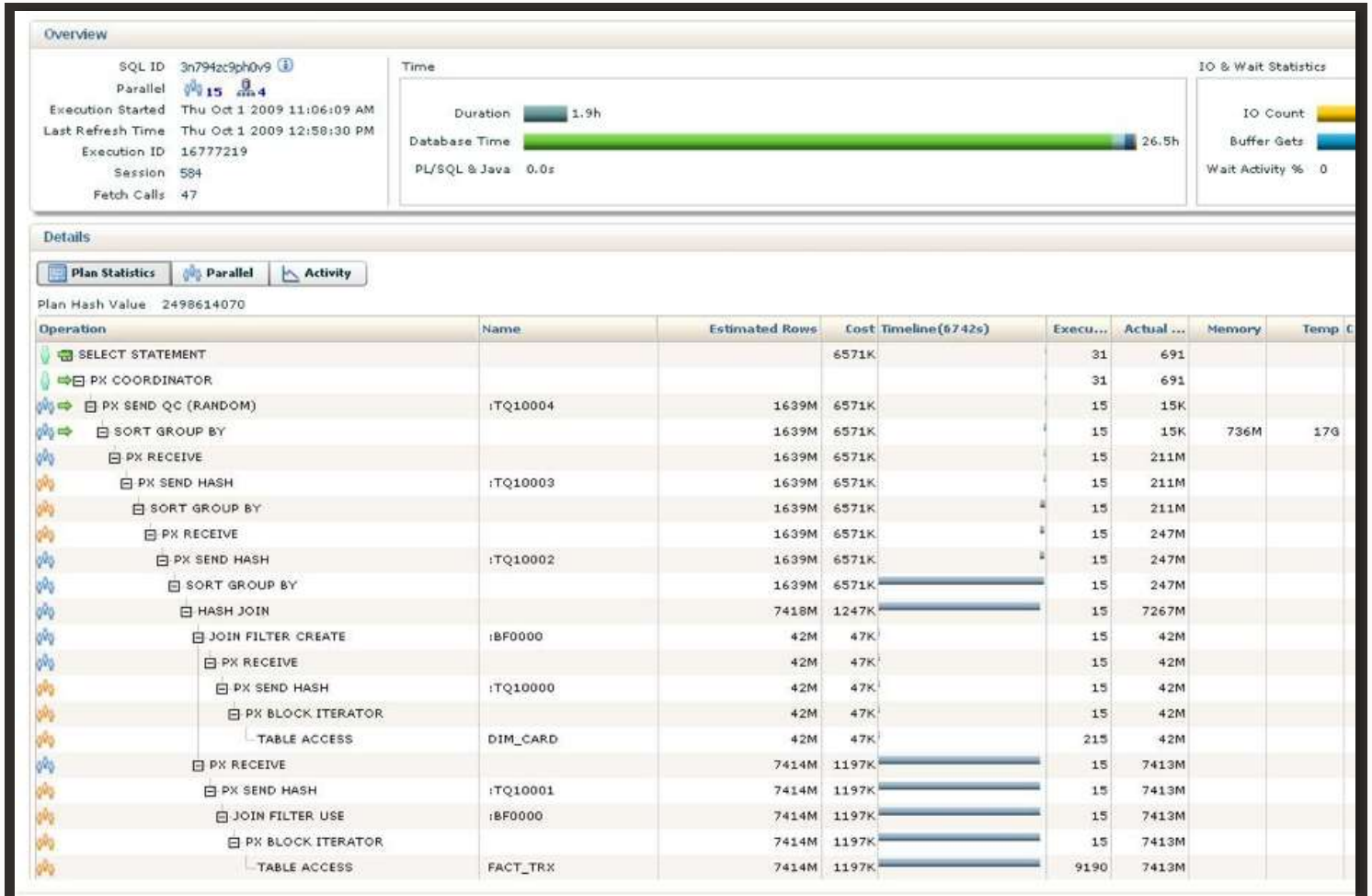
- Yes! Go-live in early November 2009
  - Two weeks earlier than proposed
- Total upgrade and migration time: ~20 hours
  - ~ 8 hours: Restore and recovery
  - ~ 1 hour: Database upgrade to Oracle 11.1.0.7
  - ~10 hours: Data migration to Exadata V1
  - ~ 1 hour: Smoke testing and final verification
- Dramatic performance improvements
  - Job runtimes decreased by 80%
  - User complaints about too fast performance … really!!

Customer Case | Payback

- Customer
- Project 2009
- Constraints
- Preparation
- Migration
- Success?
- **Remarks**

Not a single piece of SQL got changed!!!

- Most critical job: runtime from 30hrs to < 2hrs



Operation	Name	Estimated Rows	Cost	Timeline(6742s)	Execu...	Actual ...	Memory	Temp
SELECT STATEMENT			6571K		31	691		
PX COORDINATOR					31	691		
PX SEND QC (RANDOM)	:TQ10004	1639M	6571K		15	15K		
SORT GROUP BY		1639M	6571K		15	15K	736M	17G
PX RECEIVE		1639M	6571K		15	211M		
PX SEND HASH	:TQ10003	1639M	6571K		15	211M		
SORT GROUP BY		1639M	6571K		15	211M		
PX RECEIVE		1639M	6571K		15	247M		
PX SEND HASH	:TQ10002	1639M	6571K		15	247M		
SORT GROUP BY		1639M	6571K		15	247M		
HASH JOIN		7418M	1247K		15	7267M		
JOIN FILTER CREATE	:BF0000	42M	47K		15	42M		
PX RECEIVE		42M	47K		15	42M		
PX SEND HASH	:TQ10000	42M	47K		15	42M		
PX BLOCK ITERATOR		42M	47K		15	42M		
TABLE ACCESS	DIM_CARD	42M	47K		215	42M		
PX RECEIVE		7414M	1197K		15	7413M		
PX SEND HASH	:TQ10001	7414M	1197K		15	7413M		
JOIN FILTER USE	:BF0000	7414M	1197K		15	7413M		
PX BLOCK ITERATOR		7414M	1197K		15	7413M		
TABLE ACCESS	FACT_TRX	7414M	1197K		9190	7413M		

Photo by Adam Muise on Unsplash

Migration Strategies

Data Guard

Data Guard

Advantages

- No.1 choice for hardware refresh/exchange
- Fast and simple
- Avoid copy downtime
- Test it multiple times
- Independent of file system, raw devices and ASM
- Cross-platform

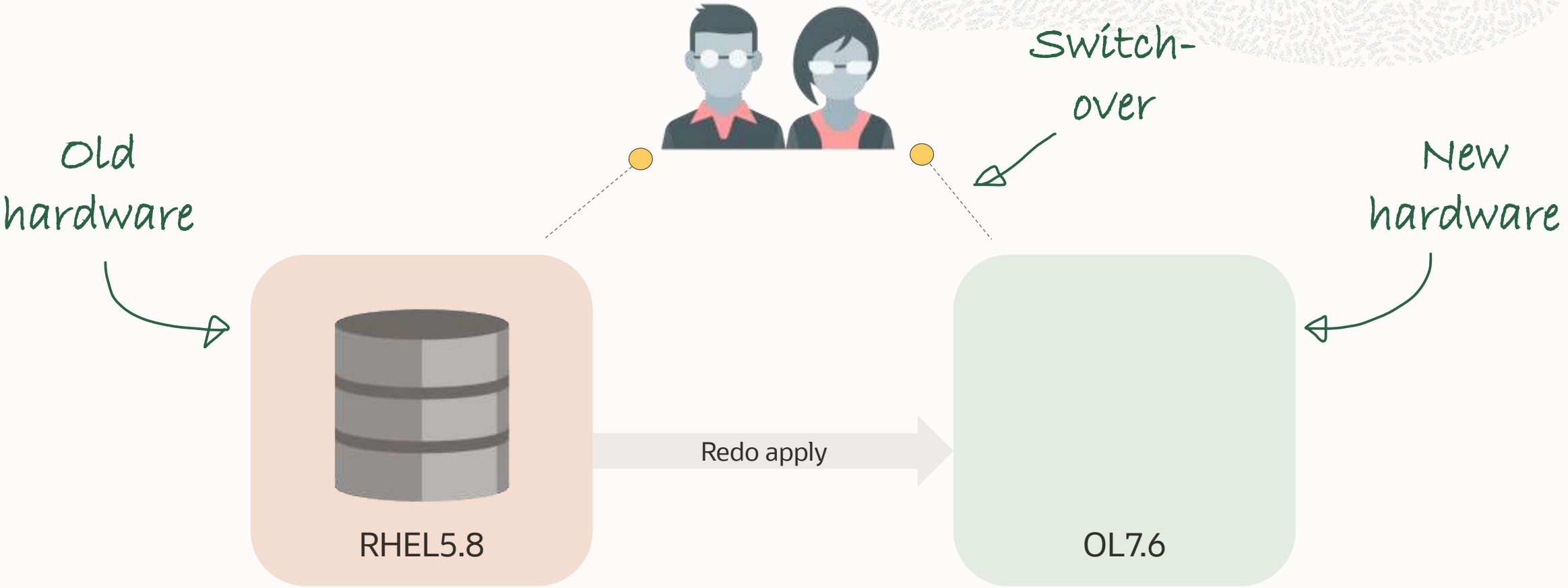
Consideration

- Requires source software on target hardware since Oracle 12c
- Does not work cross-Endianness

Documentation

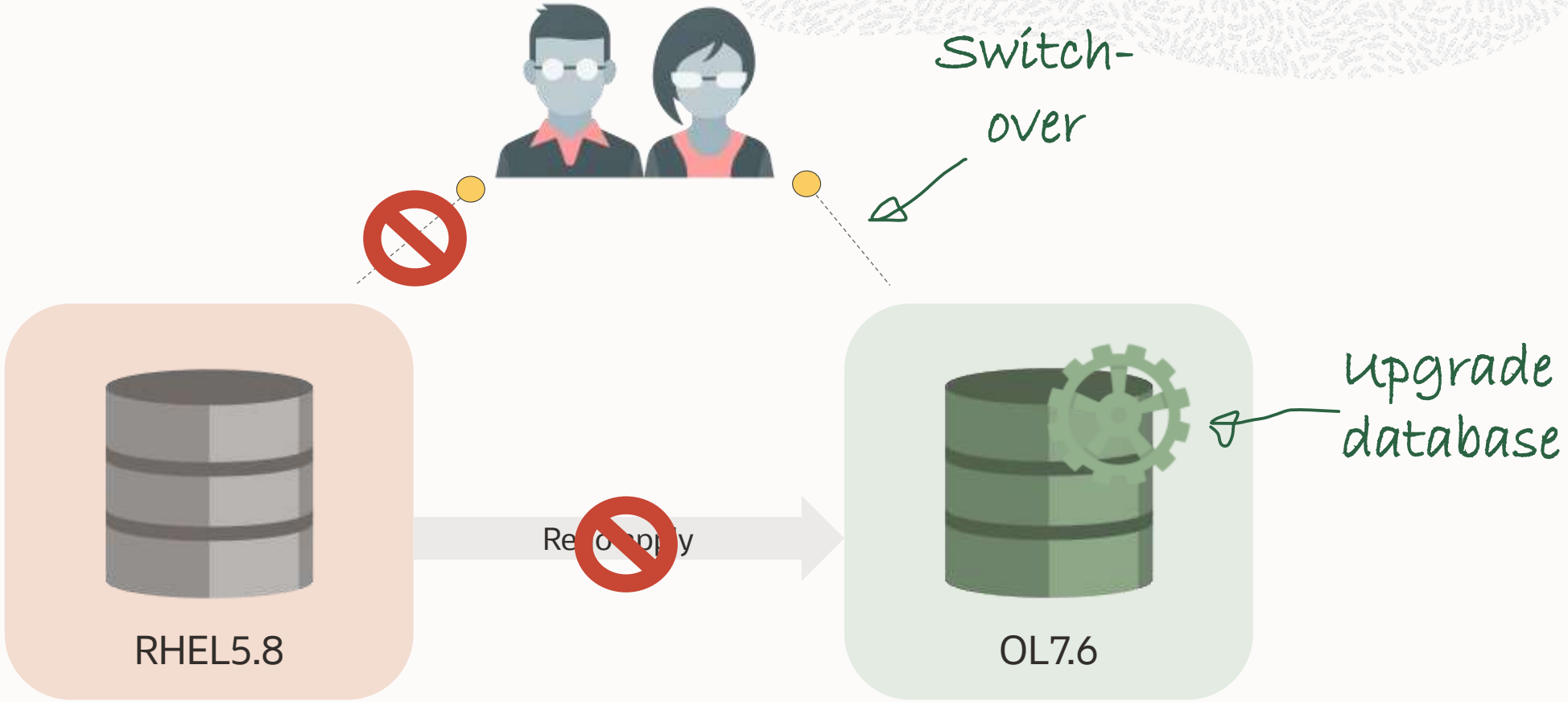
- Oracle 19c Data Guard Concepts and Administration

Data Guard | Migrate

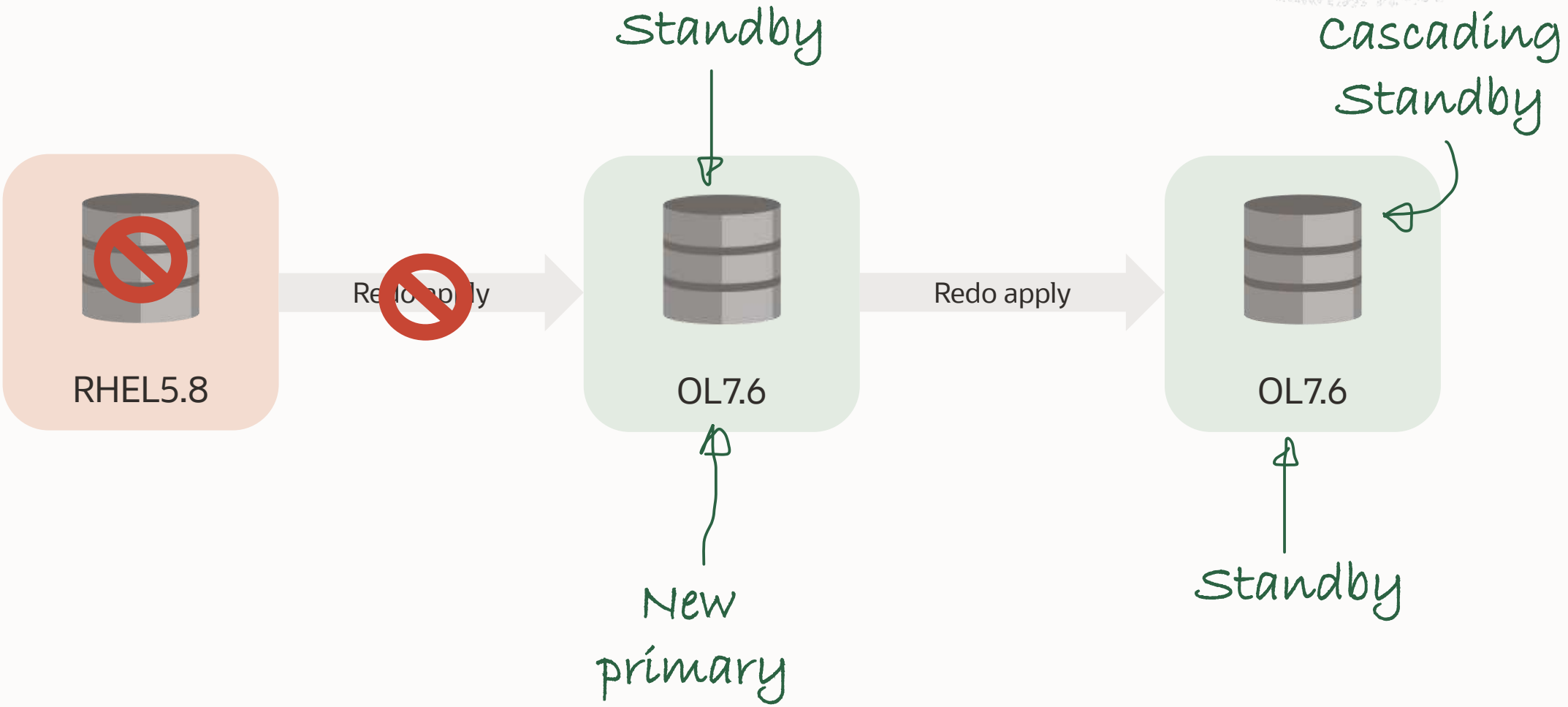



```
RMAN> DUPLICATE TARGET DATABASE
      FOR STANDBY
      FROM ACTIVE DATABASE ...
```

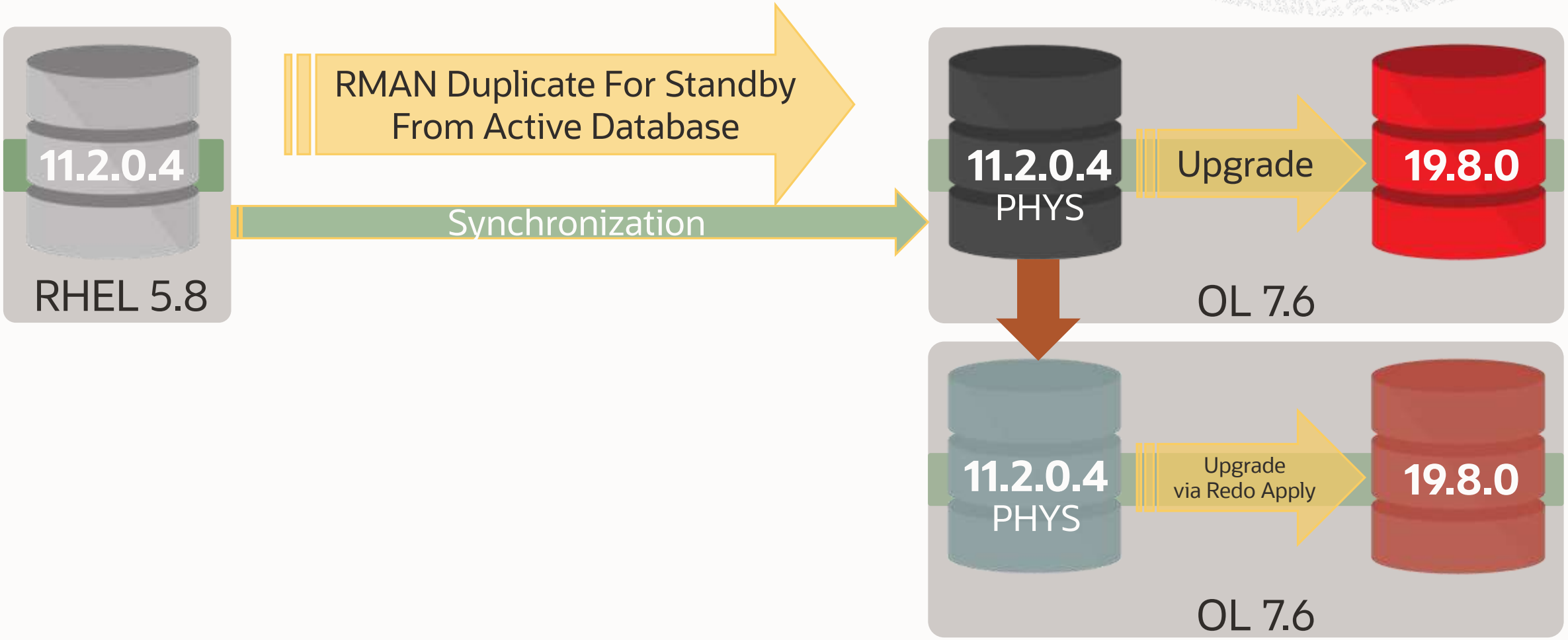
Data Guard | Migrate and Upgrade



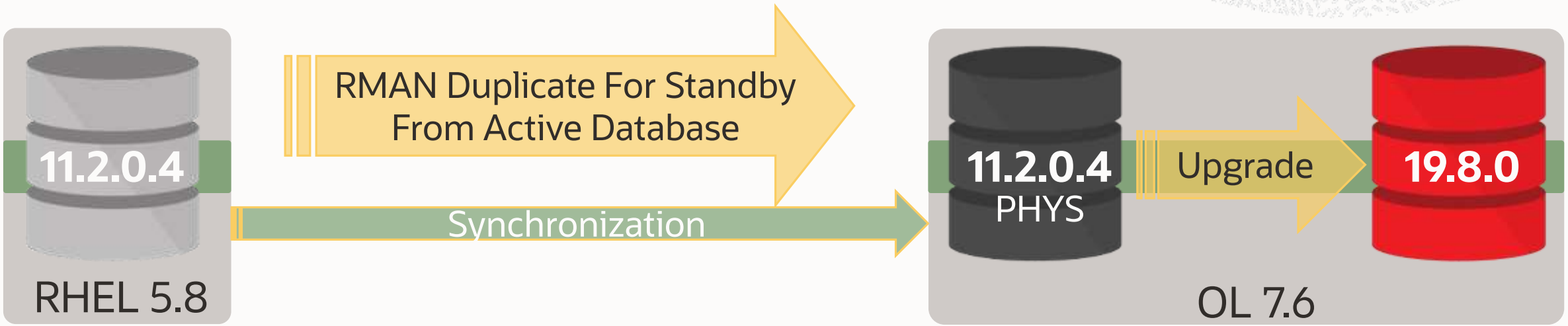
Data Guard | Migrate with Cascading Standby



Data Guard as a Migration Vehicle with Cascaded Standbys



Data Guard as a Migration Vehicle




```
rman target sys/xy@UPGR auxiliary sys/xy@UPGR2

RUN
{
   DUPLICATE TARGET DATABASE TO UPGR2
   FROM ACTIVE DATABASE;
}
```

Data Guard | Important Notes

MOS Note: 273015.1
Migrating to RAC using Data Guard

MOS Note: 413484.1
DG Support for Heterogeneous Primary and Physical Standbys in Same DG Configuration

MOS Note: 1079563.1
RMAN DUPLICATE/RESTORE/RECOVER Mixed Platform Support

MOS Note: 2439602.1
Implement Standby in OCI (Bare Metal) and how to gather diagnosti information for any failure

MOS Note: 881421.1
Using Active Database Duplication to Create Cross Platform Data Guard Setup (Windows/Linux)

MOS Note: 1617946.1
Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC)

MOS Note: 1055938.1
Migrating from HP Oracle Database Machine to Sun Oracle Database Machine 11.2 using Data Guard

Using Data Guard as migration vehicle

Payback GmbH Germany
Exadata Migration Project 2012

Source: obs / Payback GmbH / Bild: Payback

Customer Case | Payback

- Customer
- Project 2012
- Constraints
- Preparation
- Upgrade
- Success?
- Remarks
- 2020?

Payback GmbH

- Belongs to American Express
- HQ in Munich, Germany
- Develops and operates professional customer loyalty programs based on customized IT solutions

Customer Case | Payback

- Customer
- **Project 2012**
- Constraints
- Preparation
- Migration
- Success?
- Remarks
- 2020?

Migrate 14TB from Exadata V1 to Exadata X2-2

Project timeline: 2 months including all tests

How to?

- MOS Note: 1055938.1
  Hardware and Oracle Migration using Data Guard (Case 2)

1. Use RMAN duplicate create a physical standby on the 11.2 DBM
2. Manually copy archive logs to the 11.2 DBM Recover archive logs to bring standby forward When the standby is caught up except for the current logs, shutdown the application, restart the database in exclusive mode, archive log current, copy the remaining the logs and apply. Depending this step, your downtime will vary. Upgrade and recompilation time vary per application.
3. Activate the standby, open the database, and perform the upgrade.

Customer Case | Payback

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- Success?
- Remarks
- 2020?

Oracle 11.1.0.7 software must not be installed on Exadata X2-2

- Upgrading source Exadata to 11.2.0.3 not an option

Database 14TB

Downtime: less than 8hrs

Network "bottleneck"

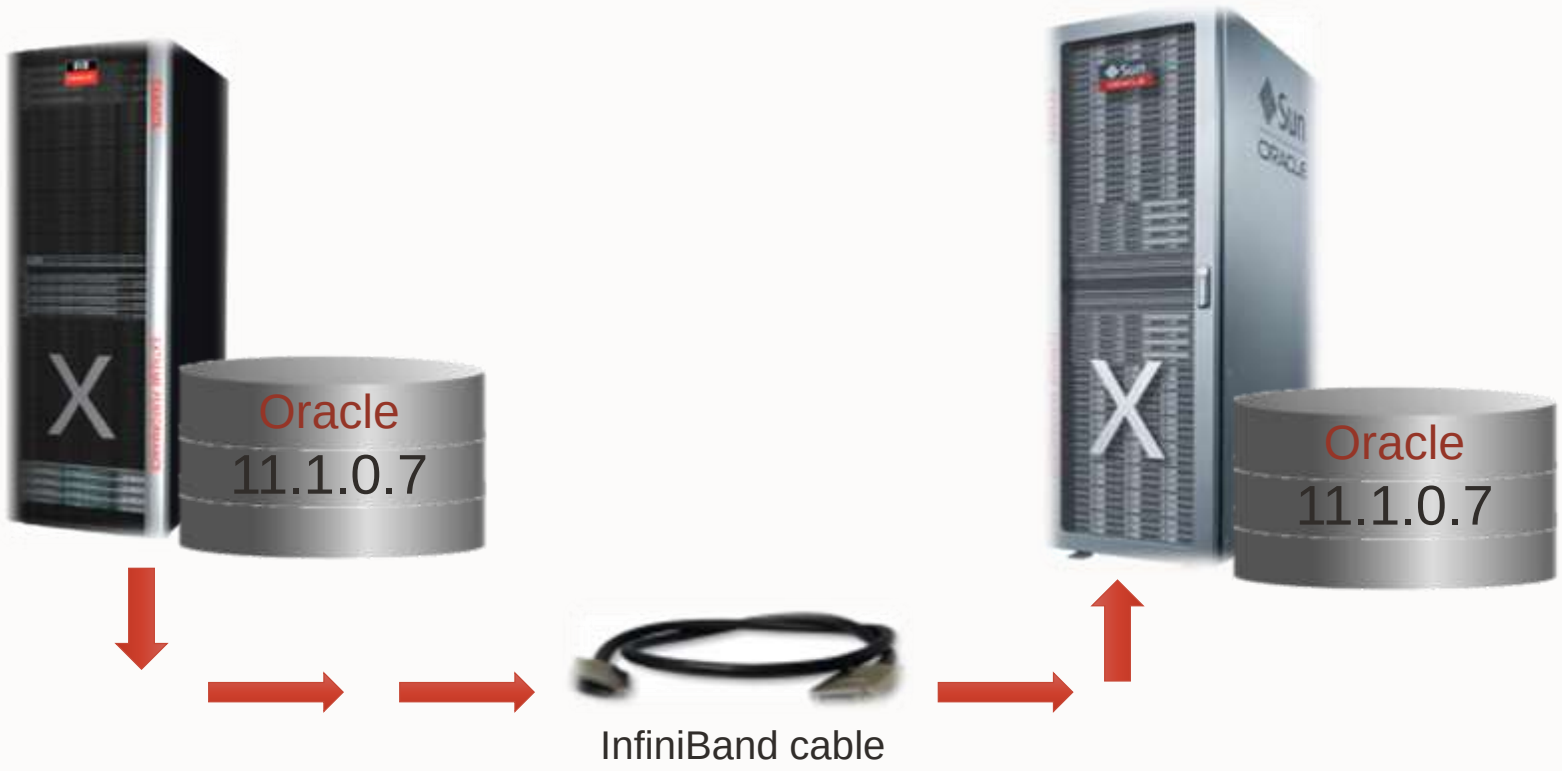
- Remedy: Special IB cabled connection from V1 to X2-2

Customer Case | Payback

- Customer
- Project 2012
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks
- 2020?

Restoring 14TB with RMAN

- `DUPLICATE FOR STANDBY FROM ACTIVE DATABASE`

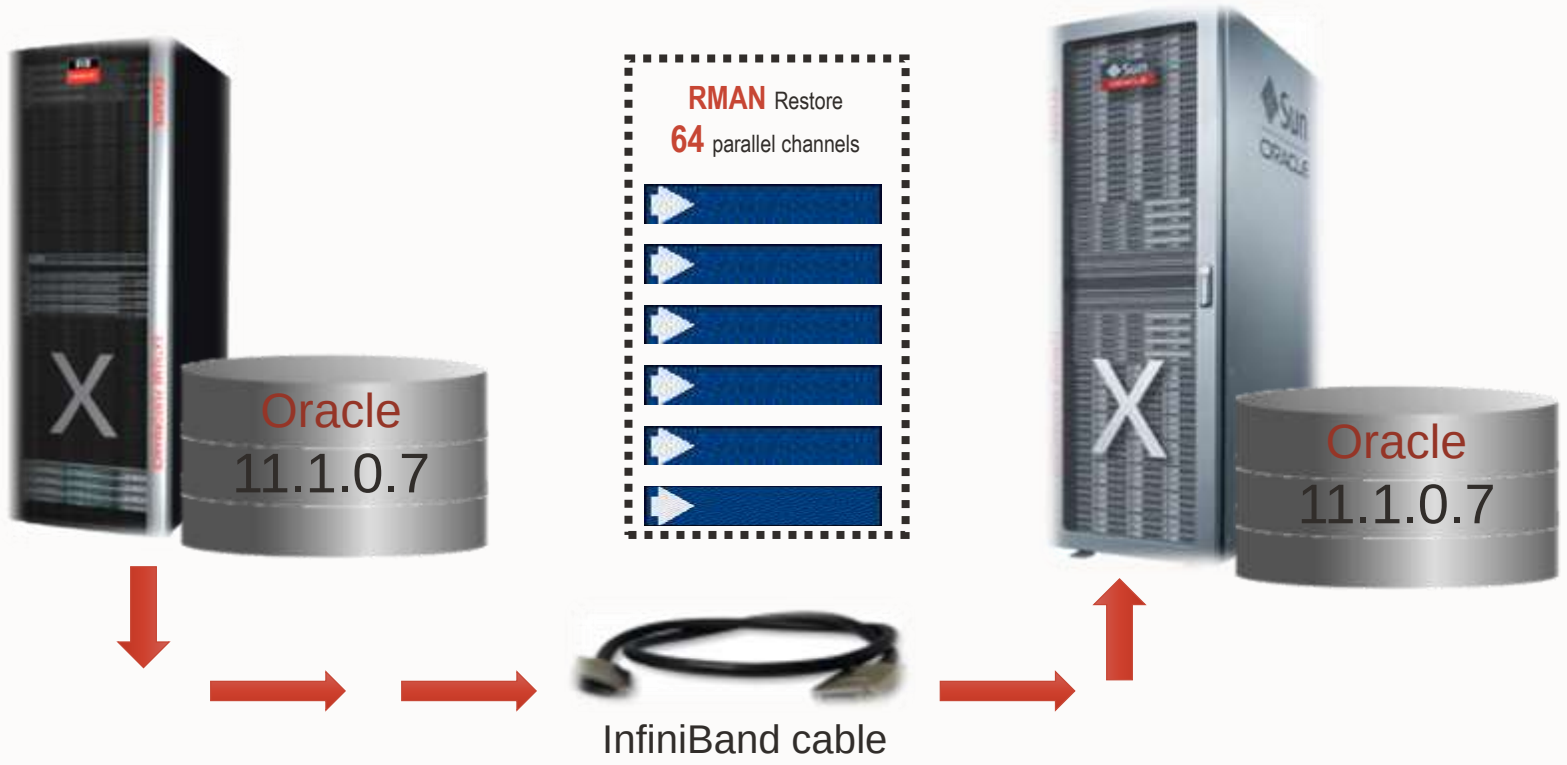


Customer Case | Payback

- Customer
- Project 2012
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks
- 2020?

Live upgrade/migration

- RMAN Restore and Recovery: <3 hours
  - 64 parallel RMAN channels allocated: >4TB/hour



Customer Case | Payback

- Customer
- Project 2012
- Constraints
- Preparation
- **Migration**
- Success?
- Remarks
- 2020?

Database upgrade 11.1.0.7 ⇨ 11.2.0.3

- Used the **new PARALLEL UPGRADE** tool `catctl.pl` **as Beta customer**
  - Total database upgrade time including recompilation: 20 mins

Customer Case | Payback

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- **Success?**
- Remarks
- 2020?

Live? And alive?

- Yes! Go-live on 3-JUL-2012
  - Almost three weeks earlier than proposed
- Total migration and upgrade time officially: ~4 hours
  - < 3 hours: Restore for Standby and recovery
  - < 20 mins: Database upgrade to Oracle 11.2.0.3
  - ~ 40 mins: Extra tasks (crsctl etc.)

Customer Case | Payback

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- Success?
- **Remarks**
- 2020?

A few plans did change – but we were prepared ☺

- AWR and SQL Plan Management

Physical standby as migration vehicle was the key technique

- Allows several test runs
- Copy time does not account for downtime

Customer Case | Payback

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- Success?
- Remarks
- 2020?

Today, Payback has many production databases on Oracle 19.8.0

Photo by Alternate Skate on Unsplash

Migration Strategies

Rolling Upgrades with Transient Logical Standby

Transient Logical Standby

Advantages

- Leverage your physical standby for upgrades
- Well proven approach
- Less than 5 minutes of downtime

Considerations

- Can't be done on same hardware
- Log miner performance (SQL Apply)

- Unsupported data types in Oracle 19c
  - ROWID, UROWID
  - Nested tables
  - Objects with nested tables
  - Identity columns
- Unsupported partitioning types in 19c
  - System
  - Reference

Documentation

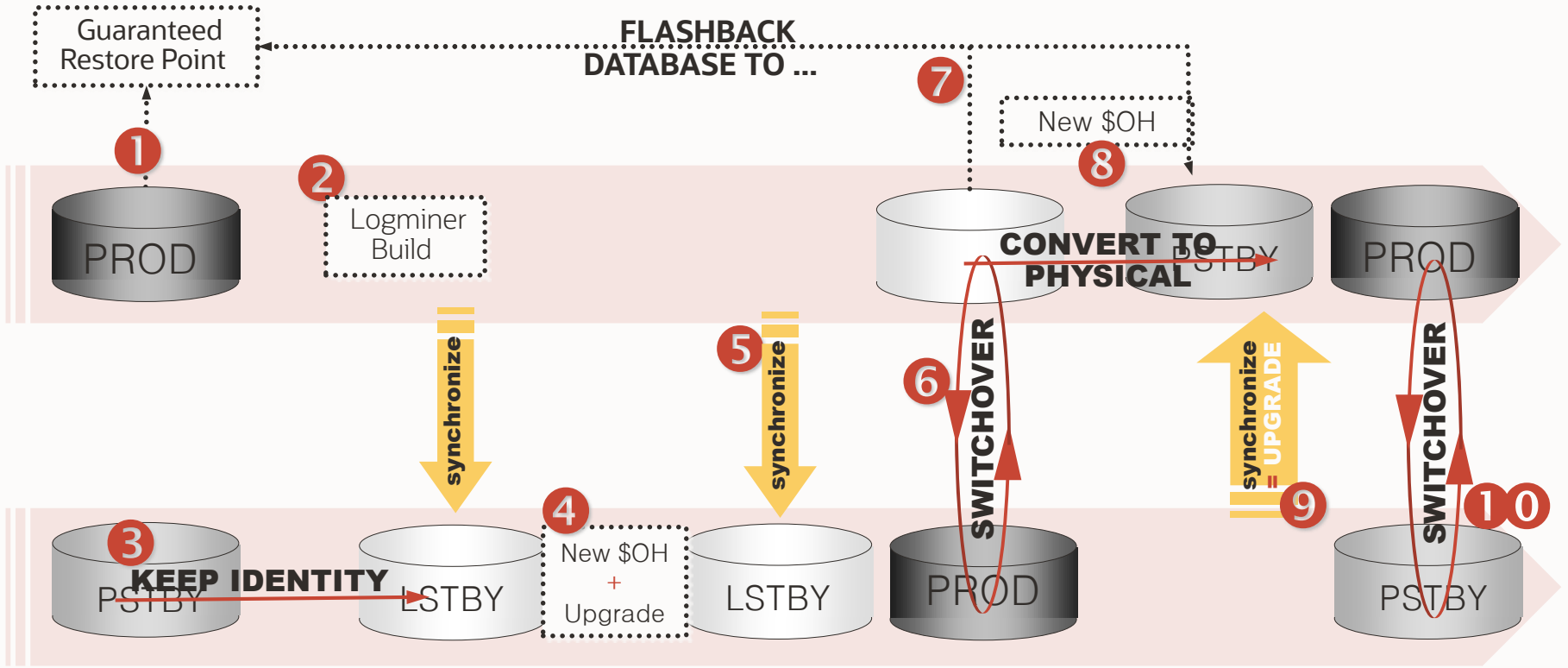
- Oracle 19c Data Guard Concepts and Administration

Transient Logical Standby | Concept

Rolling upgrade using a physical standby
Converting it temporarily into a logical standby
Converting it back into a physical standby

1. Start with Physical Standby database
2. Set a guaranteed restore point
3. Convert Physical Standby into Logical Standby
4. Upgrade Logical Standby
5. Switchover – Standby will become upgraded production database now
6. Then: Flashback the former production database
7. Convert it into a Physical Standby
8. Upgrade happens implicitly by log apply
9. Switchover to the original setup

Transient Logical Standby | Classical



[Oracle Database Rolling Upgrades Using a Physical Standby Database]

[MOS Note: 2350945.1] - Using Transient Logical Rolling Upgrade for Database Migration

Transient Logical Standby | Differentiation

Classical

- Execute all steps on the command line
- Works since Oracle 11.1.0.7
- Requires no extra license

Advanced

- Use shell scripts provided by Oracle
- Works since Oracle 11.2.0.3
- Requires no extra license

DBMS_ROLLING

- Use `DBMS_ROLLING` package
- Works since Oracle 12.1.0.2
- Requires Active Data Guard license

Transient Logical Standby | Advanced

MOS Note: 949322.1

Oracle11g Data Guard: Database Rolling Upgrade Shell Script

- Potentially not adjusted for Oracle 12c and newer

Transient Logical Standby | DBMS_ROLLING

MOS Note: 2086512.1

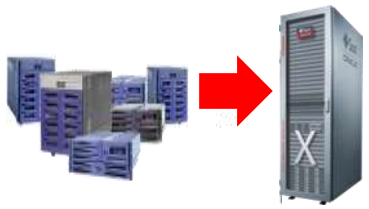
Rolling upgrade using DBMS_ROLLING - Complete Reference Package DBMS_ROLLING

- Semi-automation of Transient Logical Standby Rolling Upgrade
- Supports Data Guard Broker
  - INIT_PLAN
  - DESTROY_PLAN
  - BUILD_PLAN
  - SET_PARAMETER
  - START_PLAN
  - FINISH_PLAN
  - ROLLBACK_PLAN
  - SWITCHOVER
- Usable for upgrades from 12.1.0.2 and later releases
  - *DBMS_ROLLING usage will require a license for Active Data Guard*

Transient Logical Standby | Nippon Steel & Sumitomo Metal

Benefits

"Consolidating 4 Databases including Steel factory systems onto Exadata providing High performance and reliability, Enabling making use of High Quality of infrastructure."

Consolidation of	Minimize planned	High performance
	5 minutes 	

Business Objectives

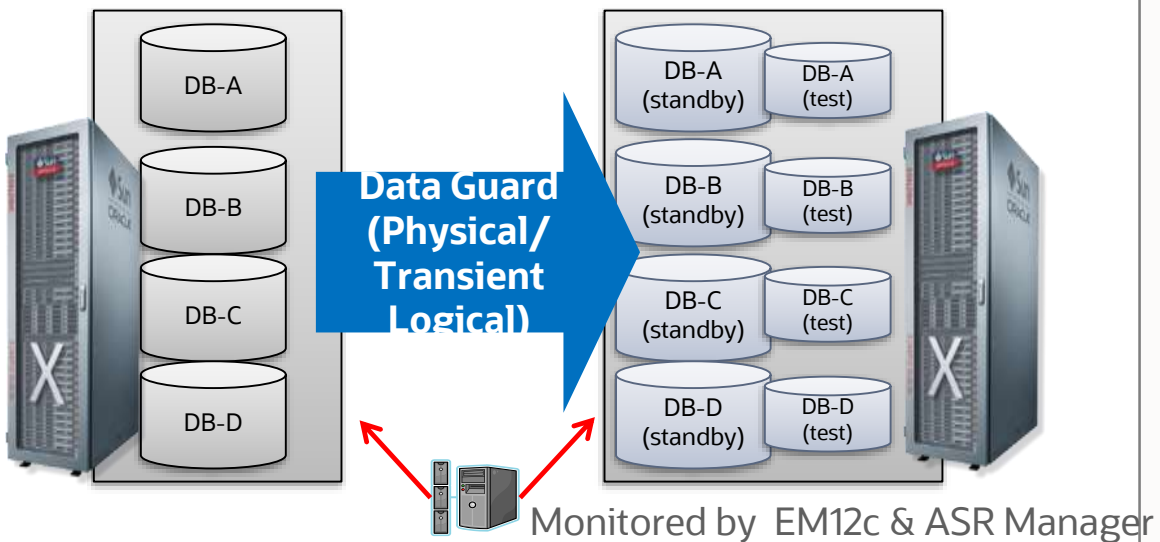
- High availability
- DB Infra consolidation

Solution

- Adopting Exadata providing high performance and high availability
- Rolling upgrade using Data Guard minimizes planned downtime

- Rolling upgrade using Transient Logical Standby realized minimizing downtime of upgrading DB (11.2.0.3→11.2.0.4)
- 5 minutes downtime x 2times (switchover) per 1 DB

- Mainframe migration
- Consolidating 4 DBs including Steel factory system & DWH onto Exadata



Migration Strategies

Transportable Tablespaces

Full Transportable Exp/Imp

Transportable Tablespaces

Advantages

- No.1 choice for VLDB Endianness migrations
- Independent of file system, raw devices and ASM
- Cross-Endianness since Oracle 10g
- Works to same and higher version
- Does work to SE2

Considerations

- Tablespaces need to be in read-only mode for live transport
- No structural changes possible
- Does not work to lower release
- Does not work from SE2 to EE

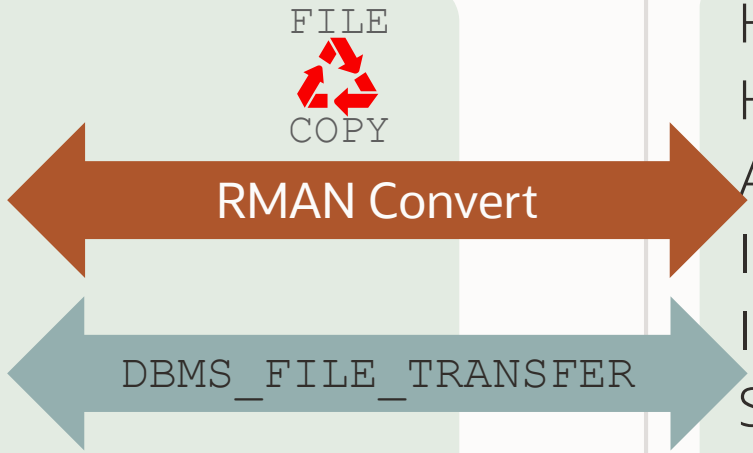
Documentation

- Oracle 19c Database Administrator's Guide – Transporting Data

TTS | Cross Platform Support

Little Endian platforms

- HP IA Open VMS
- HP Open VMS
- HP Tru64 UNIX
- Linux IA (32-bit)
- Linux IA (64-bit)
- Linux x86 64-bit
- Microsoft Windows IA (64-bit)
- Microsoft Windows x86 64-bit
- Microsoft Windows IA (32-bit)
- Solaris Operating System (x86)
- Solaris Operating System (x86-64)

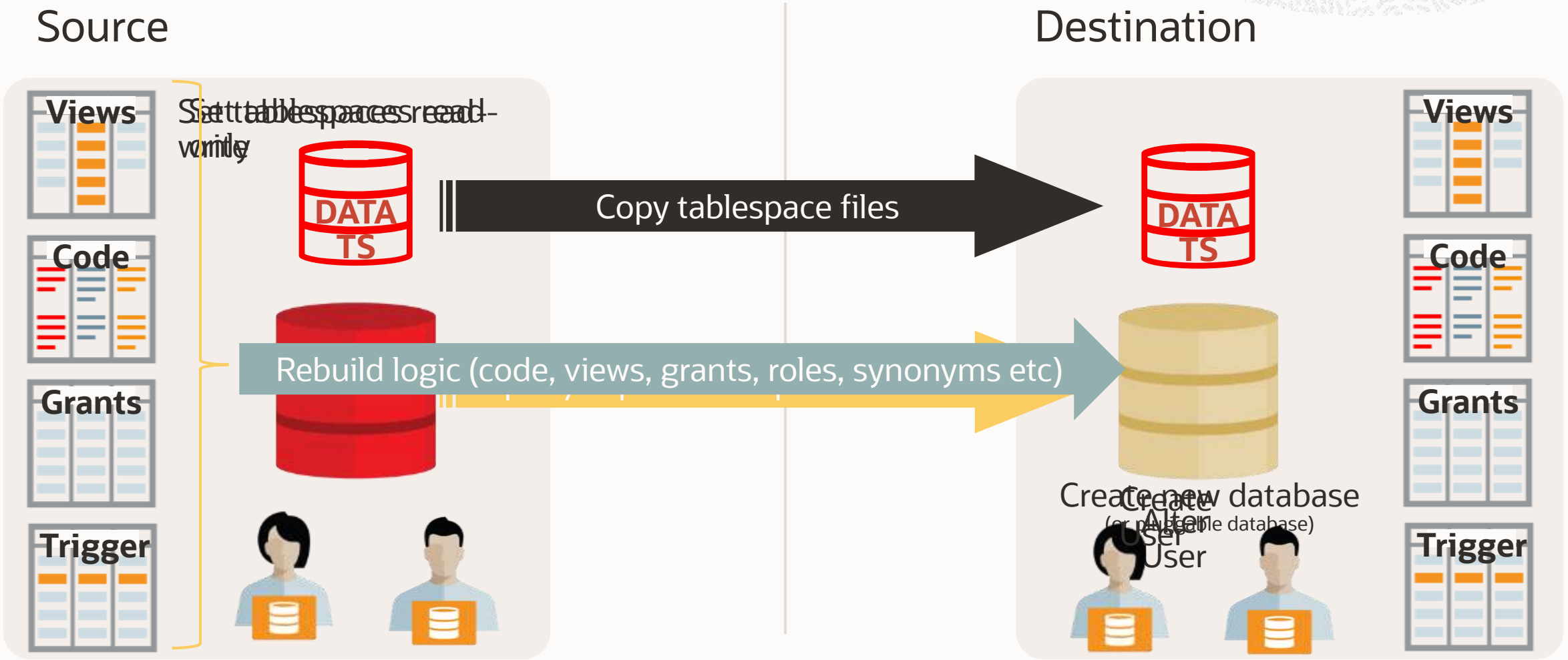


Big Endian platforms

- HP-UX (64-bit)
- HP-UX IA (64-bit)
- AIX-Based Systems (64-bit)
- IBM zSeries Based Linux
- IBM Power Based Linux
- Solaris[tm] OE (32-bit)
- Solaris[tm] OE (64-bit)

Check `V$TRANSPORTABLE_PLATFORM` for exact platform names as mentioned in the slide

TTS | Database Migration



TTS | Further Information

Transportable Tablespaces – Information

- MOS Note:1166564.1
  Master Note for Transportable Tablespaces - Common Questions and Issues
- MOS Note:1454872.1
  Transportable Tablespace Restrictions and Limitations: Details, Reference, and Version Where Applicable
- For TTS Technical Briefs see the MAA webpage
- Database Upgrades using TTS
- Platform Migration using Transportable Database (RMAN)
- Customer example: Amadeus Customer Case

Resources for EBS

- MOS Note:1581549.1 Best Practices for Minimizing Oracle E-Business Suite Release 12 Upgrade Downtime
- Oracle recommends that you upgrade to the latest Database version certified for your EBS release
  - MOS ⇨ Certifications ⇨ E-Business Suite ⇨ <version> ⇨ <platform>
    Then select the latest certified database release

The biggest pain points of TTS?

Downtime for large databases and complexity

Photo by Photos Hobby on Unsplash

TTS | Using Incremental Backups

Size

- RMAN Incremental Backups
  - MOS Note: 2471245.1
    V4 PERL Scripts to reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup

- Source: 10.2.0.3 or newer
- Target: 11.2.0.4 or newer

Complexity

- Full Transportable Export/Import
  - One-Command Migration with Data Pump

- Source: 11.2.0.3 or newer
- Target: 12.1.0.1 or newer

Can be combined

TTS | PERL Scripts

PERL scripts V4 – NEW and IMPROVED

- MOS Note: 2471245.1 – V4 Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup

PERL scripts for the ZDLRA

- MOS Note: 2460552.1 – Cross Platform Database Migration using ZDLRA

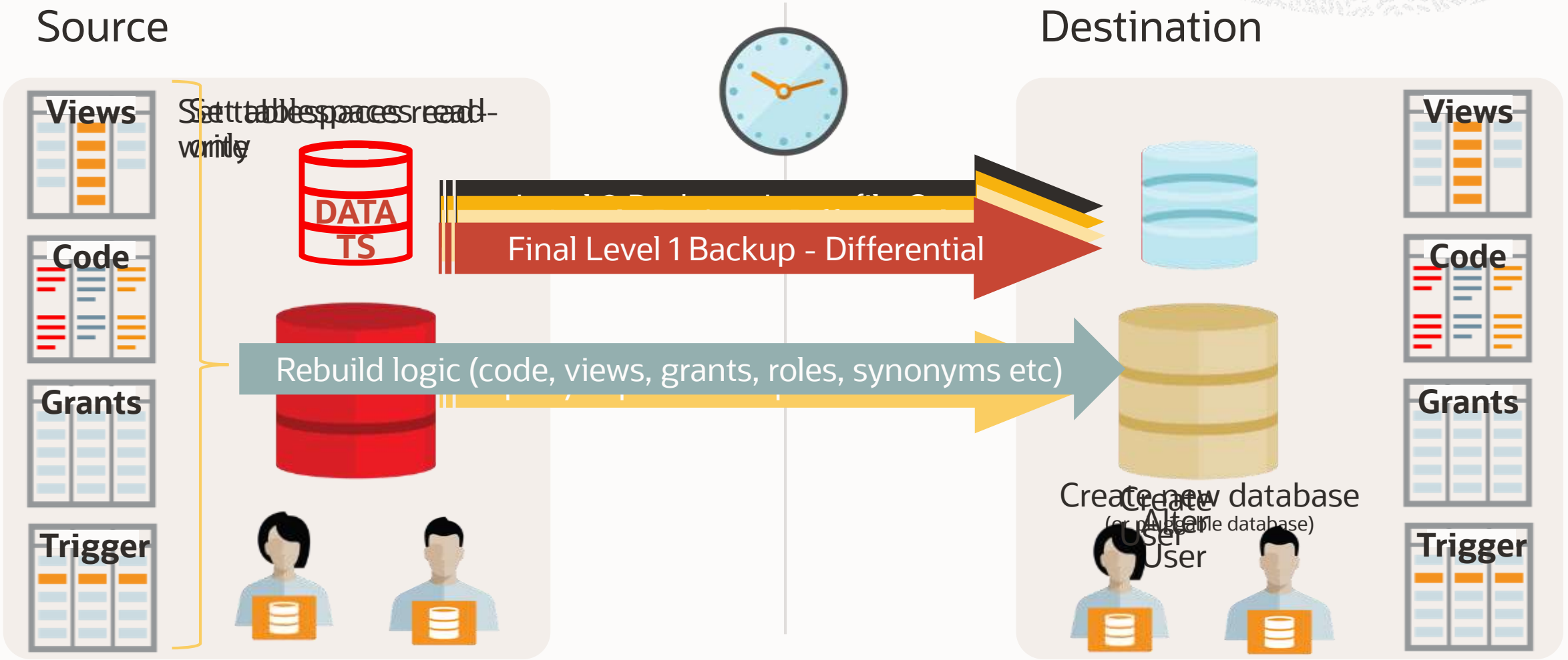
PERL script for Oracle 11g

- MOS Note:1389592.1 – 11G – Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup

PERL scripts for Oracle 12c:

- MOS Note: 2005729.1 – 12C – Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup

TTS | Using Incremental Backups



Full Transportable Export/Import | Intro

Data Pump does the manual work involved with TTS

- Tablespace contents export and import
- Meta data rebuild for views, synonyms, packages, trigger etc. including passwords

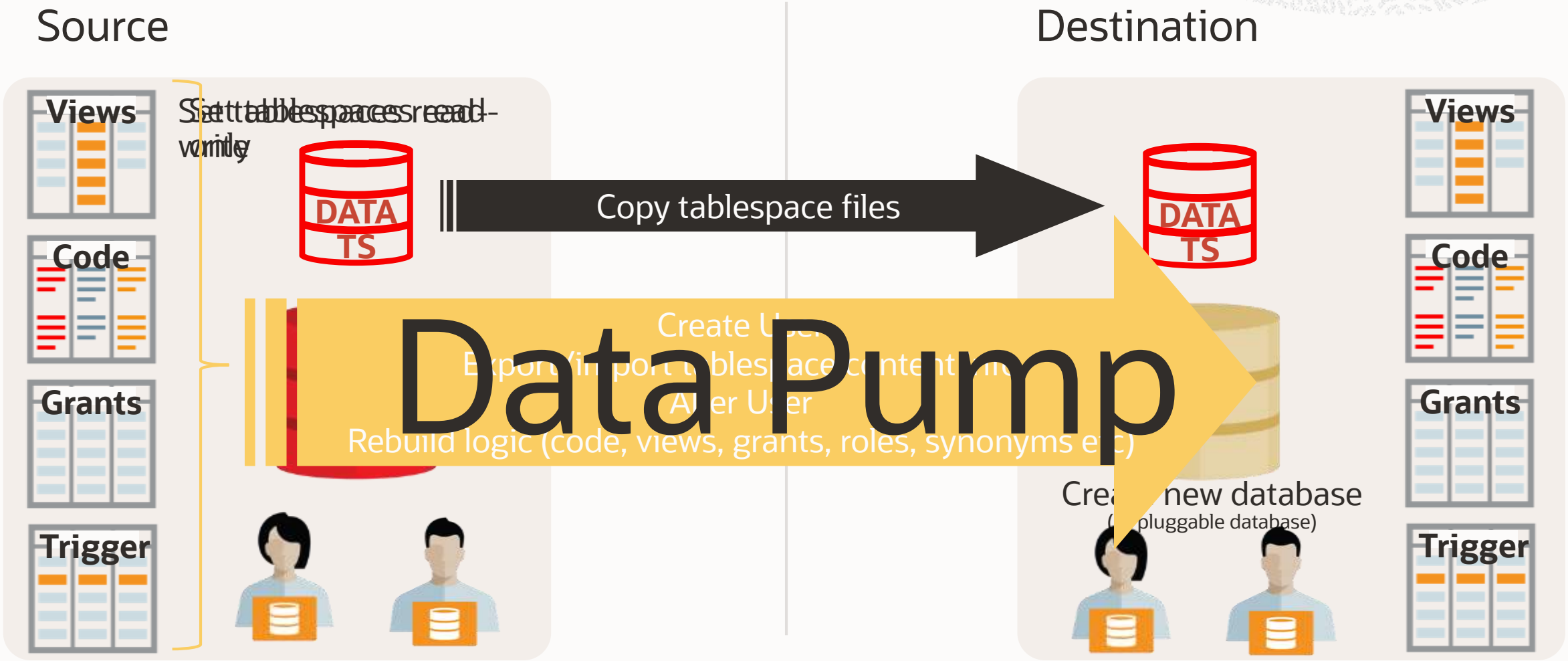
Requirements:

- Source: Oracle 11.2.0.3 or newer
- Destination: Oracle 12.1.0.1 or newer

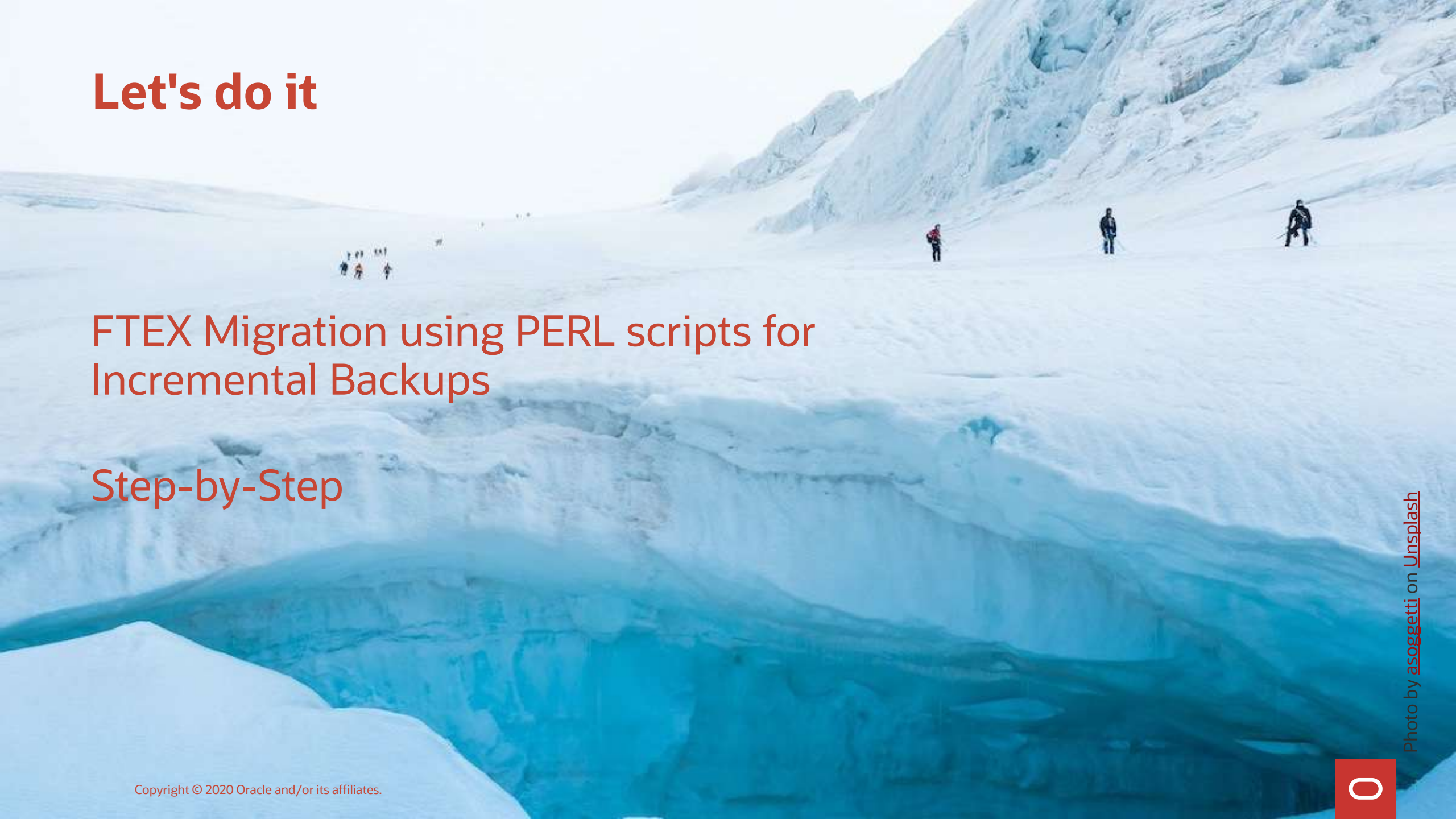
Example:

```
$ impdp mike/passwd@V19C  [NETWORK_LINK=LNK_V112]
          FULL=Y TRANSPORTABLE=ALWAYS [VERSION=12]
          METRICS=Y  EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
          LOGTIME=ALL  LOGFILE=ftex_dir:v112fullimp.log
          TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts1.dbf'
          TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts2.dbf'
```

FTEX | Database Migration



Let's do it



FTEX Migration using PERL scripts for Incremental Backups

Step-by-Step

Photo by asoggetti on Unsplash

Transport - Incremental Backups | 6 Phases

Phase 1 - Initial Setup phase
Phase 2 - Prepare phase
Phase 3 - Roll Forward phase
Phase 4 - Final Incremental Backup
Phase 5 - Transport Phase: Import all Metadata
Phase 6 - Validate and Cleanup

Phase 1 | Setup - Database Creation

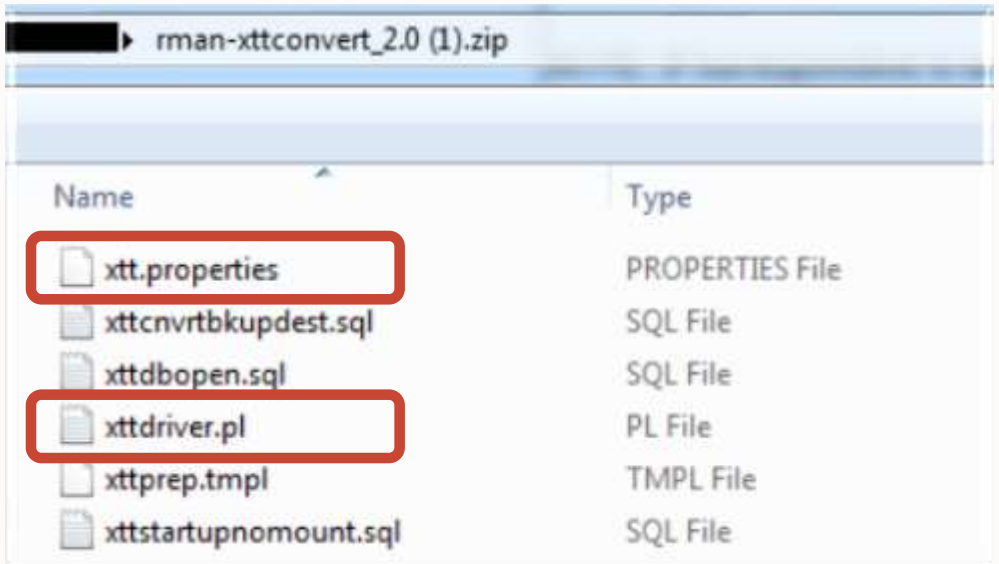
Create a destination database

- COMPATIBLE equal or higher
- Identical
  - Database character sets
  - National character sets
  - Time zone versions https://mikedietrichde.com/2016/12/08/create-a-database-with-non-default-time-zone/
  - Database time zone setting

Phase 1 | Setup - PERL Scripts

Download PERL scripts

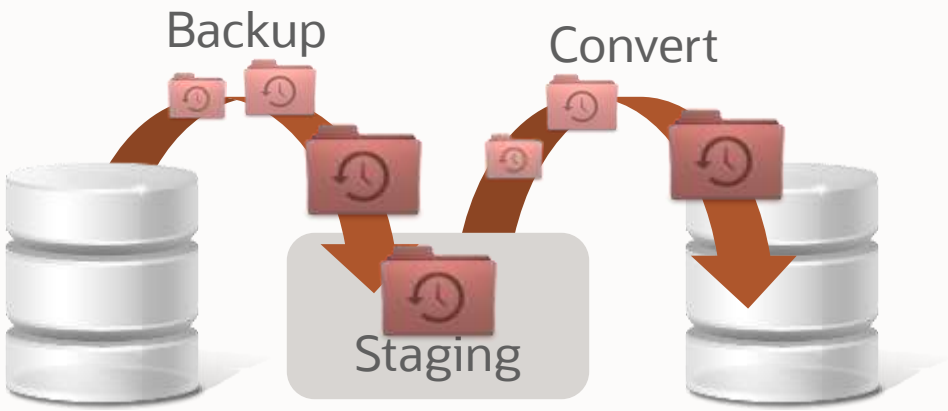
- MOS Note: 2471245.1 – V4 Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup



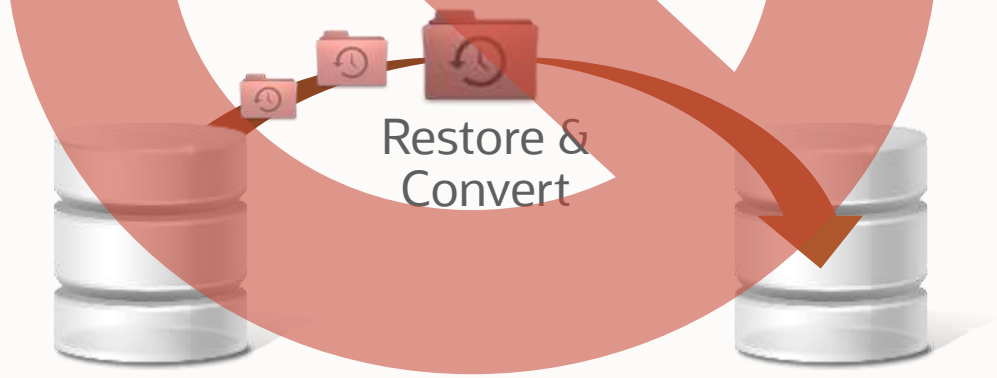
Phase 1 | Methods Choice

Choose the best method

- RMAN backup / convert
  - Requires staging space for CONVERT
  - `xttdriver.pl -p` and `-c`



- DBMS_FILE_TRANSFER *
  - 2TB limitation per file
  - No staging space
  - CONVERT implicitly
  - `xttdriver.pl -S` and `-G`



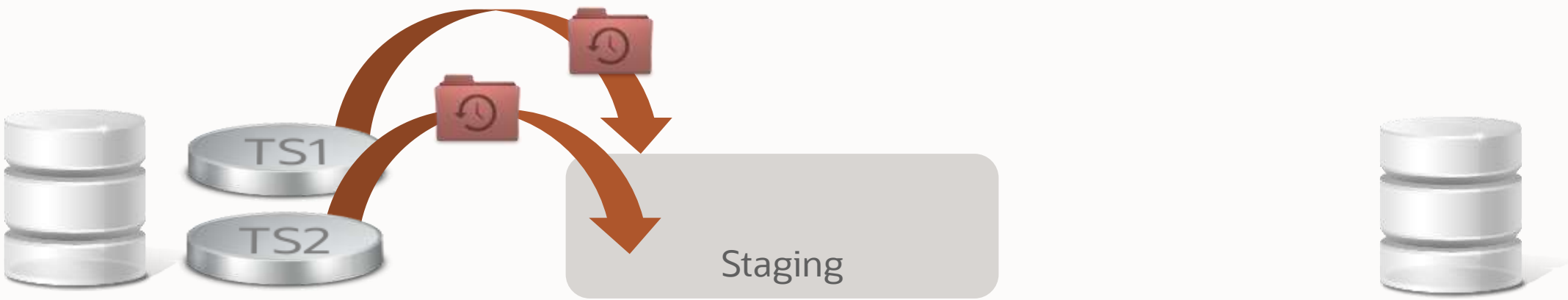
* The V4 scripts don't support DBMS_FILE_TRANSFER anymore

Phase 1 | xtt.properties Configuration

xtt.properties

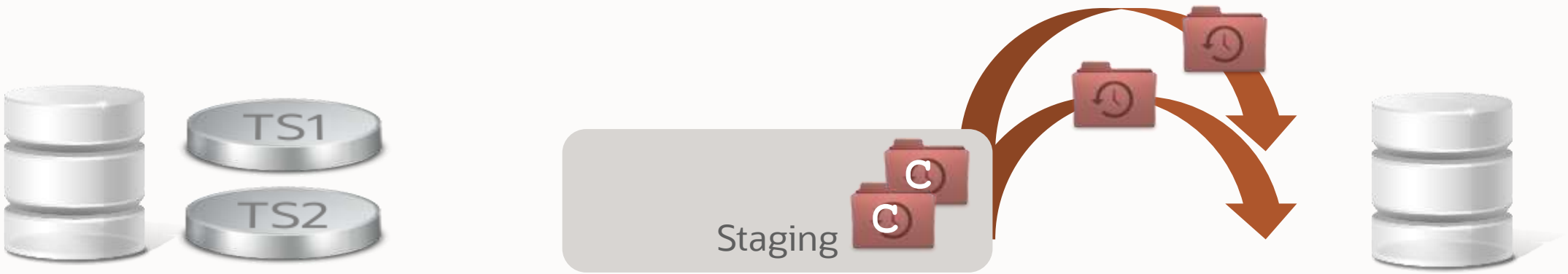
```
## Tablespaces to transport
## ========================
tablespaces=TS1,TS2

## Source database platform ID
## ===========================
platformid=13

## Source system file locations
## ============================
## Location where datafile copies are created
## during the "-p prepare" step.
dfcopydir=/oracle/DQ1/rman_stage

## backupformat
## ------------
## Location where incremental backups are created.
backupformat=/oracle/DQ1/rman_stage

## Destination system file locations
## =================================
## Location where datafile copies are placed by the user
## when they are transferred manually from souce system.
stageondest=/oracle/DQ1/rman_stage

## storageondest
## -------------
## Location where the converted datafile copies will be
## written during the "-c conversion of datafiles" step.
## This is the final location of the datafiles
## where they will be used by the destination database.
storageondest=/oracle/DQ1/sapdata50

## backupondest
## ------------
## Location where converted incremental backups
## on the destination system will be written during
## the "-r roll forward datafiles" step.
backupondest=/oracle/DQ1/rman_stage_incr
```

Phase 2 | Level-0 Backup



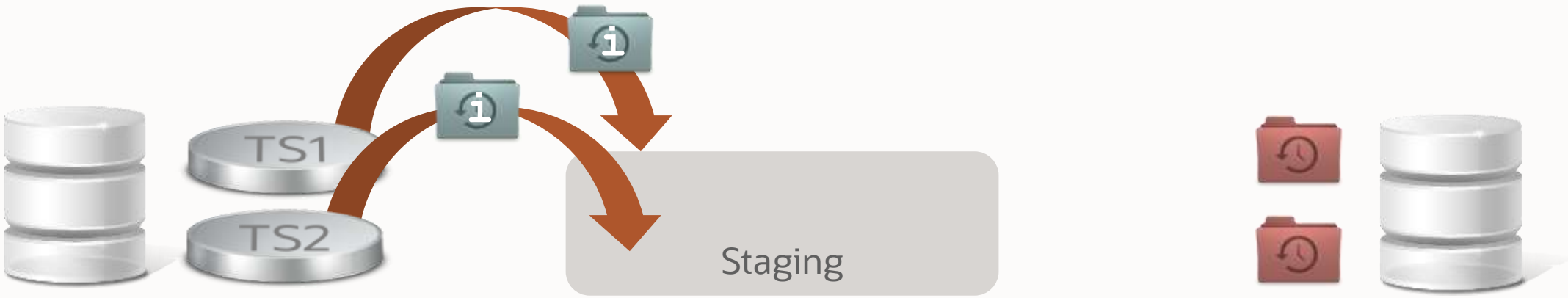

```
xttdriver.pl -p
```

Phase 2 | Conversion of level-0 backup




```
xttdriver.pl -c
```

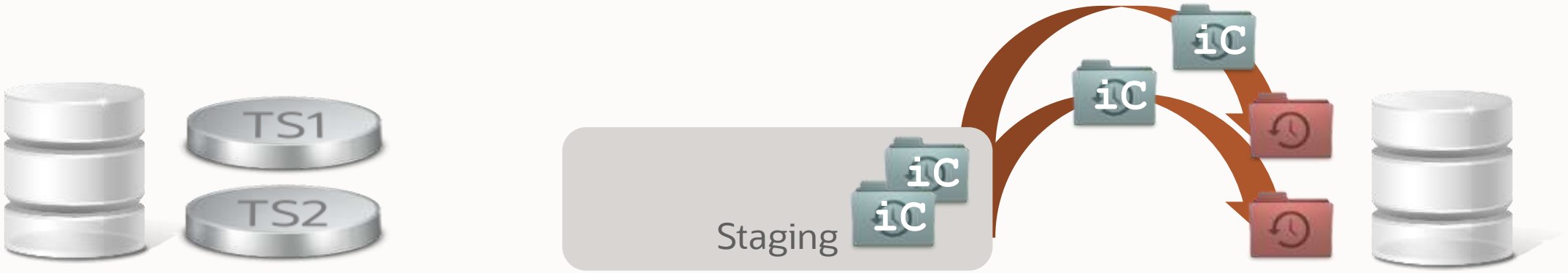
Phase 3 | Inc level-1 backup and SCN marker




```
xttdriver.pl -i
```

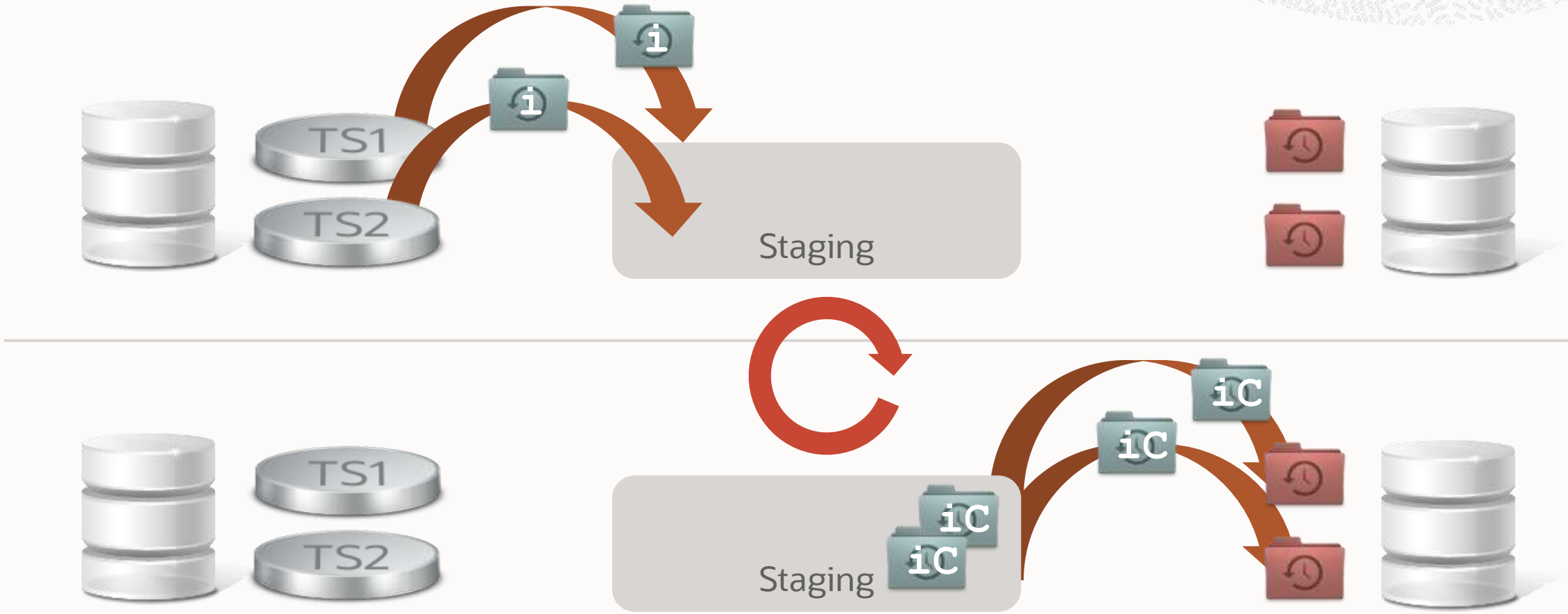

```
xttdriver.pl -s
```

Phase 3 | Inc level-1 backup: Convert / Merge




```
xttdriver.pl -r
```

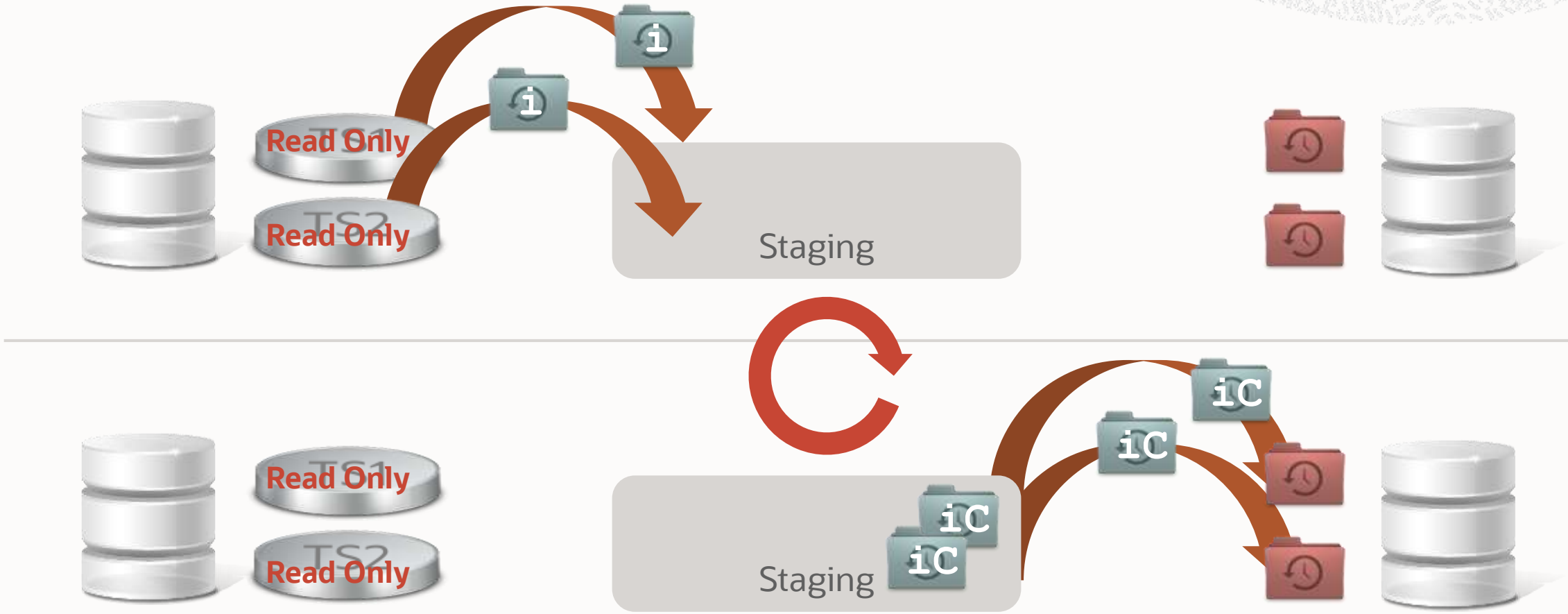
Phase 3 | Repeat Level-1: Backup/Convert/Merge



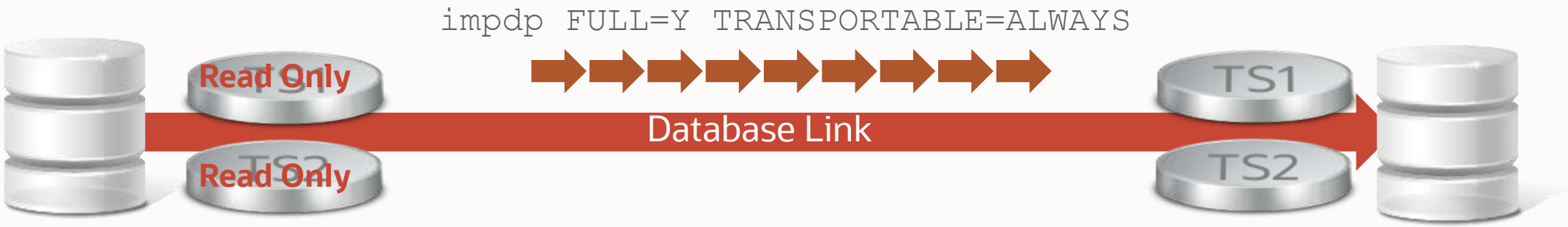
Phase 4 | Downtime - Read Only

Staging

Phase 4 | Final Inc: Backup/Convert/Merge

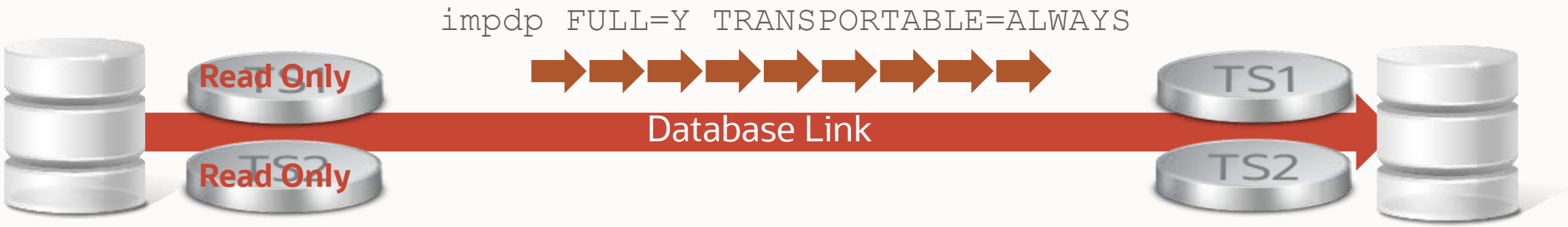


Phase 5 | Full Transportable Export/Import




```
impdp mike/passwd@v19c  NETWORK_LINK=v112
                  FULL=Y TRANSPORTABLE=ALWAYS [VERSION=12]
                  METRICS=Y EXCLUDE=STATISTICS
                  LOGTIME=ALL  LOGFILE=ftex_dir:v112fullimp.log
                  TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts1.dbf'
                  TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts2.dbf'
```

Phase 5 | Full Transportable Export/Import




```
[oracle@dest]$ impdp mike/passwd@V19C
                  FULL=Y TRANSPORTABLE=ALWAYS
                  METRICS=Y    EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
                  LOGTIME=ALL  LOGFILE=ftex_dir:v112fullimp.log
                  TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts1.dbf'
                  TRANSPORT_DATAFILES='/oracle/DQ1/sapdata50/ts2.dbf'
```

Phase 6 | Validation and Cleanup

230 Terabyte in less than 24 hours

PERL scripts in action at a top health care insurance

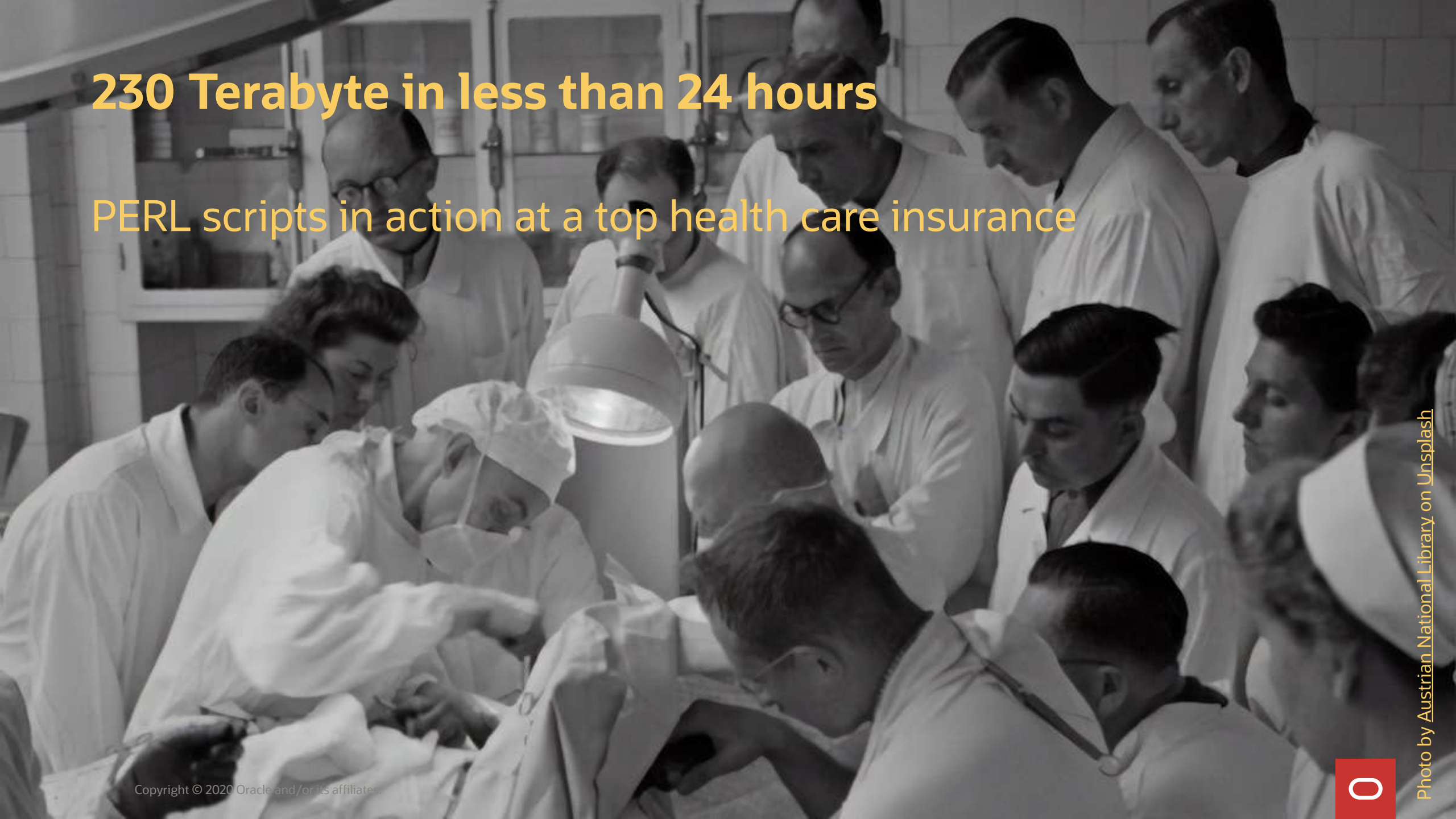


Photo by Austrian National Library on Unsplash

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- Preparation
- Migration
- Success?
- Remarks

The Client

- One of the top healthcare insurance providers in the United States
  - Over 50,000 employees, over $50 BILLION annual revenue

The Platinum Partner

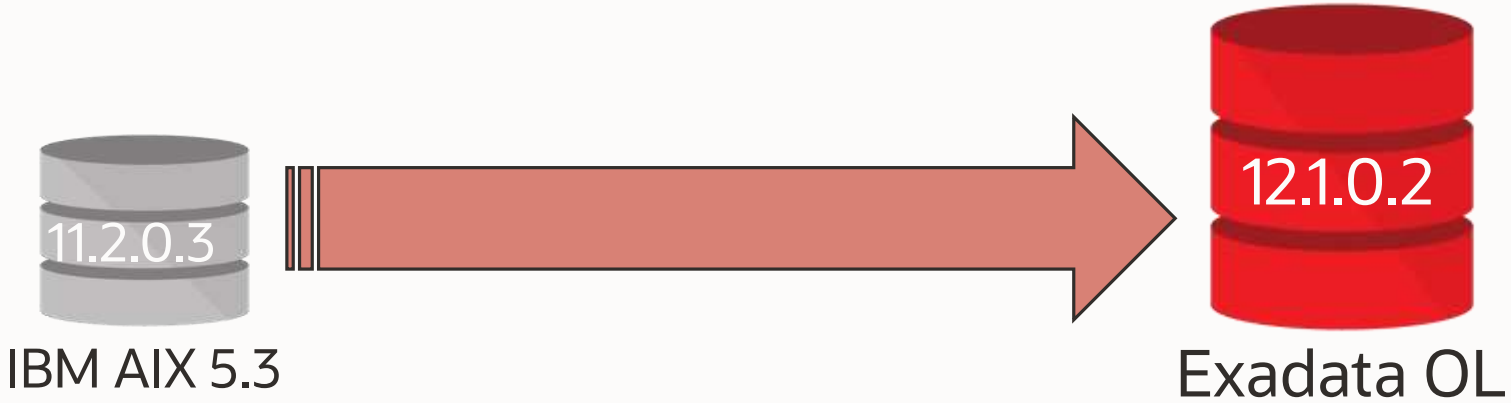
- Centric Consulting
- A management and technology consulting company

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- Preparation
- Migration
- Success?
- Remarks

The Database

- Source: AIX 5.3, Oracle Database 11.2.0.3, SI DB on filesystem
- Target: Exadata running Oracle Linux, Database 12.1.0.2, RAC/ASM
- Enterprise data warehouse & operational data store
  - Critical for day-to-day operations
  - Minimizing downtime is critical
  - Data Guard in place for disaster recovery



Customer Case | Health Care

- Customer
- Project 2017
- **Constraints**
- Preparation
- Migration
- Success?
- Remarks

Huge, active database

- 230+ TB (and growing!)
- Generates ~1.2TB redo *per hour*

Initial attempts using Oracle GoldenGate were unsuccessful

- Could not keep up with massive redo generation

v.2 of Oracle's PERL migrations scripts limitations

- Did not handle addition of tablespaces during migration
- Single-threaded file transfer

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

Single-threaded file transfer

- v.2 `xttdriver.pl` script reads tablespaces from the `xtt.properties` file
  - Default: Only processes one tablespace at a time
  - Remedy: Configure up to 8 data files to be processed concurrently with PARALLEL
- Transfer was too slow for our efforts
  - 100 MB/sec throughput
  - For 230 TB: almost 27 days (!) just for the Prepare phase

Workaround

- Reduced prepare phase from 27 days to 6 days
  - Created forty (40) identical directories
    - Each held a complete XTTS utility installation
  - Broke up 530+ tablespaces into 40 equal tablespace groups
  - Migrated 40 jobs concurrently with PARALLEL=2, or 80 files at a time
  - Result: ~800 MB/sec throughput

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

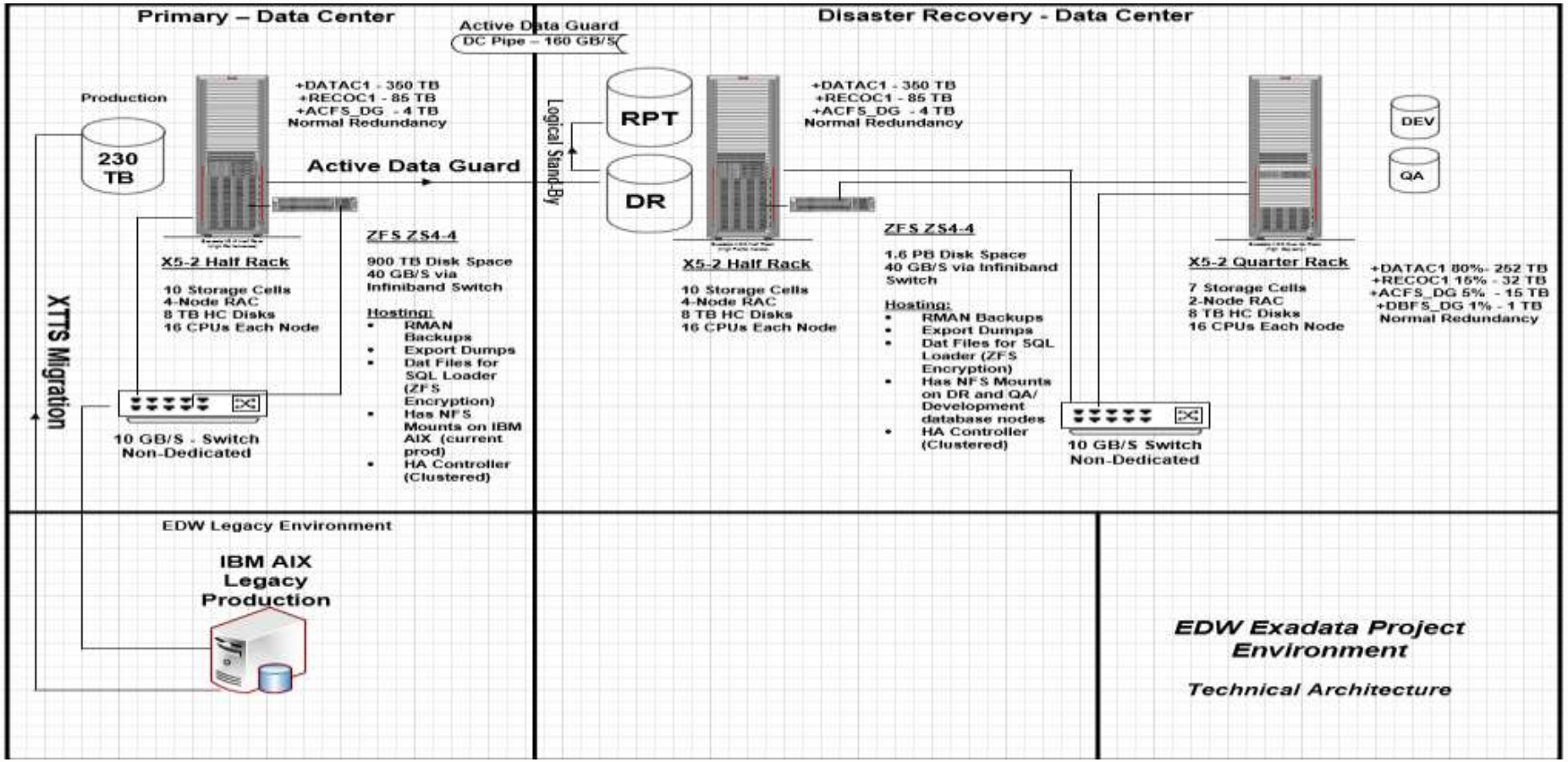
Further customizations

- Cross-check scripts to ensure all tablespaces were being migrated
- Custom scripts
  - Automate 40 parallel script executions
  - Data Pump import par file for the Plug-In Step
- Load balanced RMAN CONVERT
  - Distributed conversion phase to all 4 Exadata nodes equally
  - Convert speed up to over 230 TBs in under 10 hours.

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Environment



Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- Preparation
- **Migration**
- Success?
- Remarks

Migration and upgrade completed in one phase

- AIX ⇢ Linux
- 230+ TB
- Database 11.2.0.3 ⇢ 12.1.0.2
- Single Instance ⇢ RAC
- File system ⇢ ASM

Everything done in an 18-hour READ ONLY window!

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- Preparation
- Migration
- **Success?**
- Remarks

Migration timeline

	Phase	Location	Duration
0	Prepare Phase (RMAN Level 0)	Source	6 Days
1	Convert Image Copies	Target	10 Hrs
2	Incremental Backup (Pass #1)	Source	24 Hrs
3	Convert / Roll-Forward	Target	12 Hrs
4	Increment SCN	Source	1 Min
5	Incremental Backup (Pass #2)	Source	12 Hrs
6	Convert / Roll-Forward	Target	6 Hrs
7	Increment SCN	Source	1 Min
8	Incremental Backup (Pass #3)	Source	6 Hrs
9	Convert / Roll-Forward	Target	3 Hrs
10	Increment SCN	Source	1 Min
	*** READ-ONLY Window		18 Hrs
11	Place Tablespaces in READ-ONLY Mode	Source	30 Mins
12	Incremental Backup (Final)	Source	3 Hrs
13	Full Meta-Data Export	Source	3 Hrs
14	Create Dynamic Import Plug-In Par File	Source	1 Hr
15	Convert / Roll-Forward	Target	1.5 Hrs
16	Import Plug-In Step	Target	11 Hrs
17	Full Meta-Data Import	Target	2 Hrs
	*** Database Migrated ***		

Customer Case | Health Care

- Customer
- Project 2017
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Get the latest version of the PERL scripts

Plan for unexpected “features” to occur

- Data Pump patches for TTS migration may vary by version
- Bug fixes for migration scripts themselves

Customize the process for VLDBs

- Otherwise the Prepare Phase may take too long

Pushing the limits

ExaCC Migration for over 1500 databases with an ZDLRA

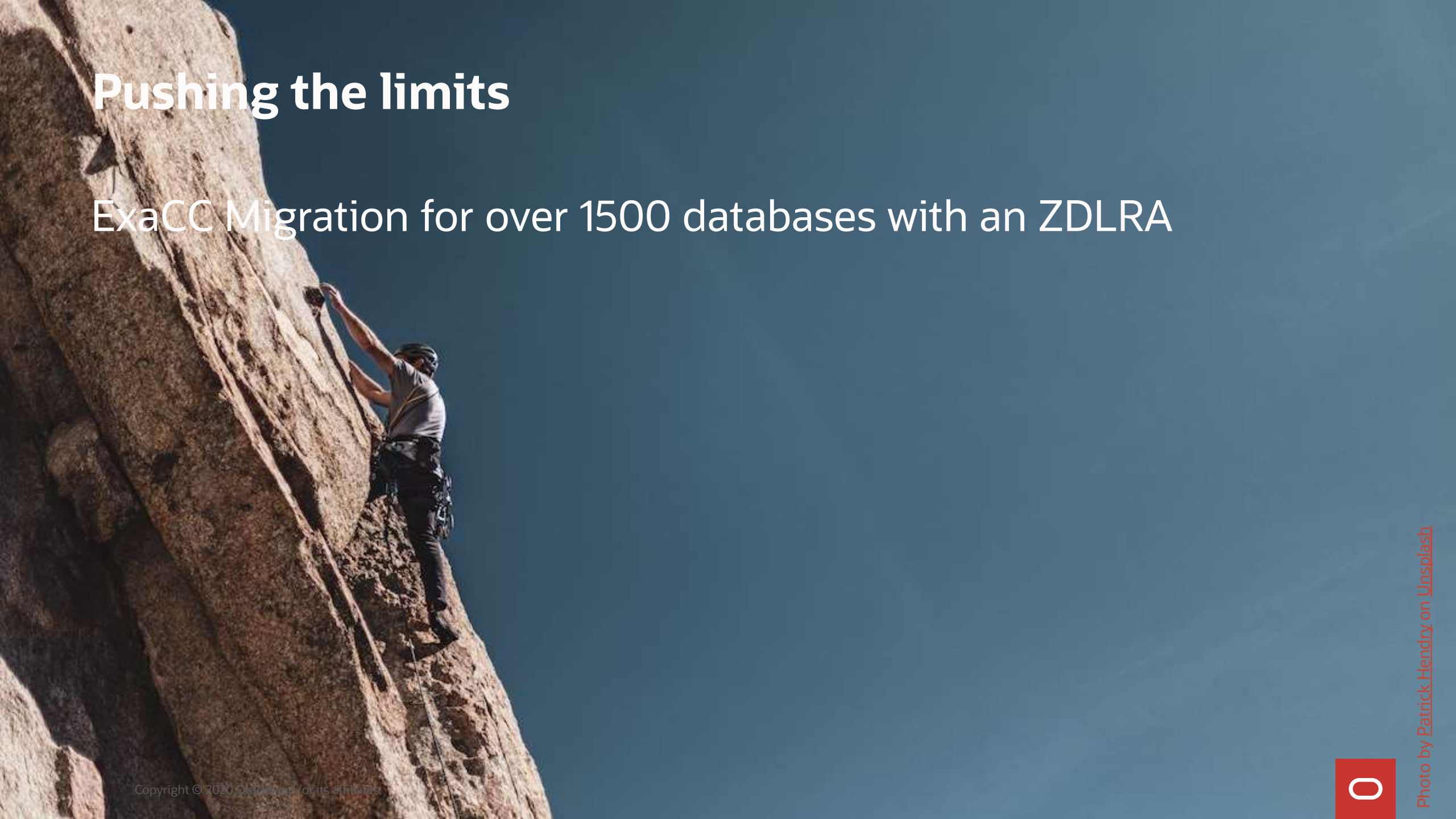


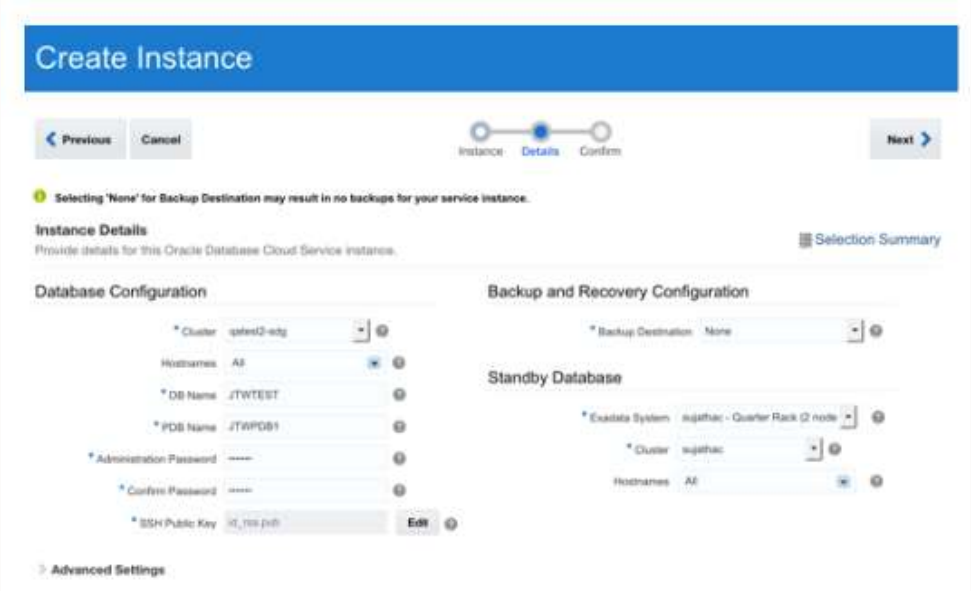
Photo by Patrick Hendry on Unsplash

ExaCC Migration | Exadata Cloud at Customer

https://www.oracle.com/technetwork/database/exadata/exacc-x7-ds-4126773.pdf

ORACLE

DATABASE
CLOUD SERVICE



ExaCC Migration | Project Goal

Cross platform migration

- AIX (Big Endian) to Linux (Little Endian)
- Options:
  - Data Pump
    - Simple but slow due to complexity and amount of data
  - Transportable Tablespaces
    - More prep work due to complexity but can deal with high amounts of data
  - Full Transportable Export/Import
    - Combines both but may be harder due to unforeseeable log errors

All-purpose migration path

- Even for same-Endianness migrations

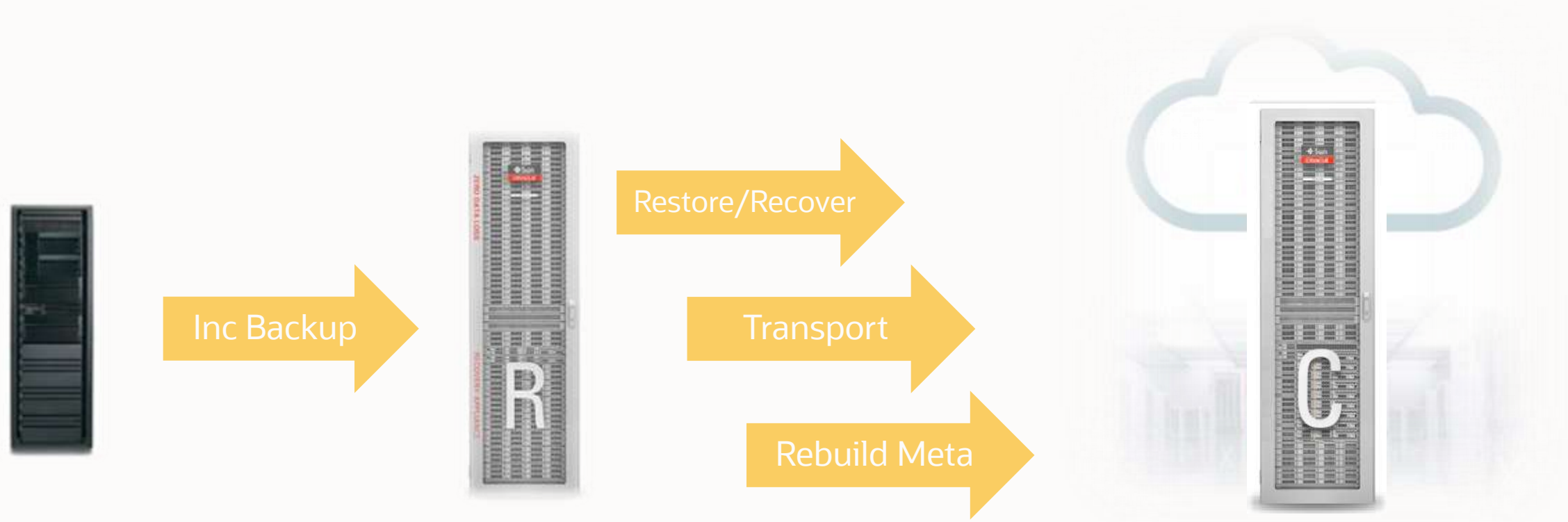
Decision

Build a universal Transportable Tablespaces solution

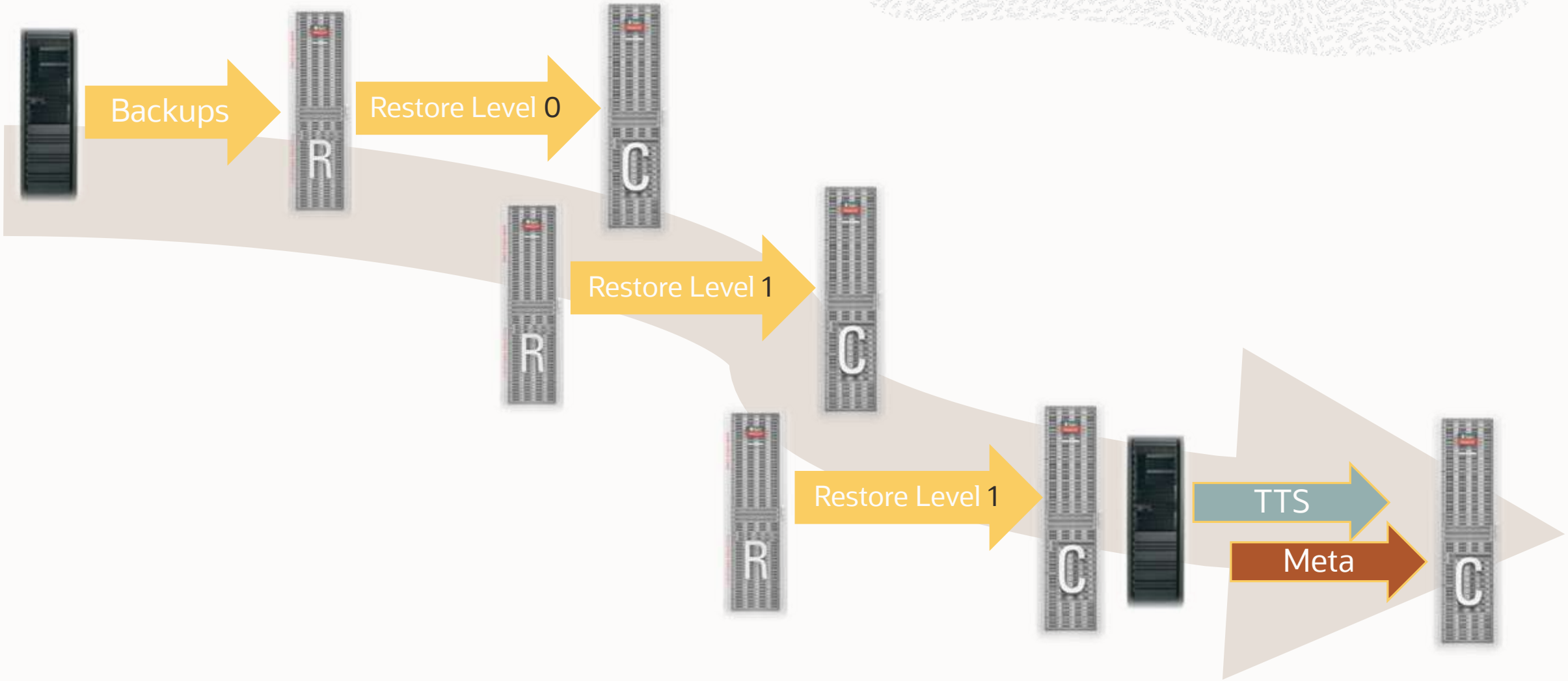
Utilize the ZDLRA for cross platform incremental backup/restore

Photo by Javier Allegue Barros on Unsplash

ExaCC Migration | Strategic Overview

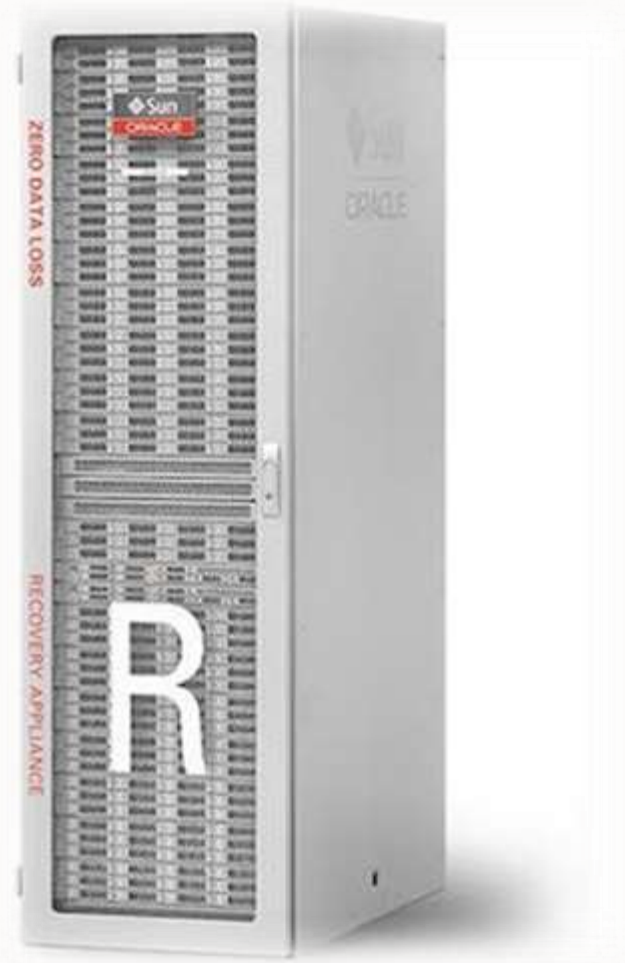


ExaCC Migration | Timeline

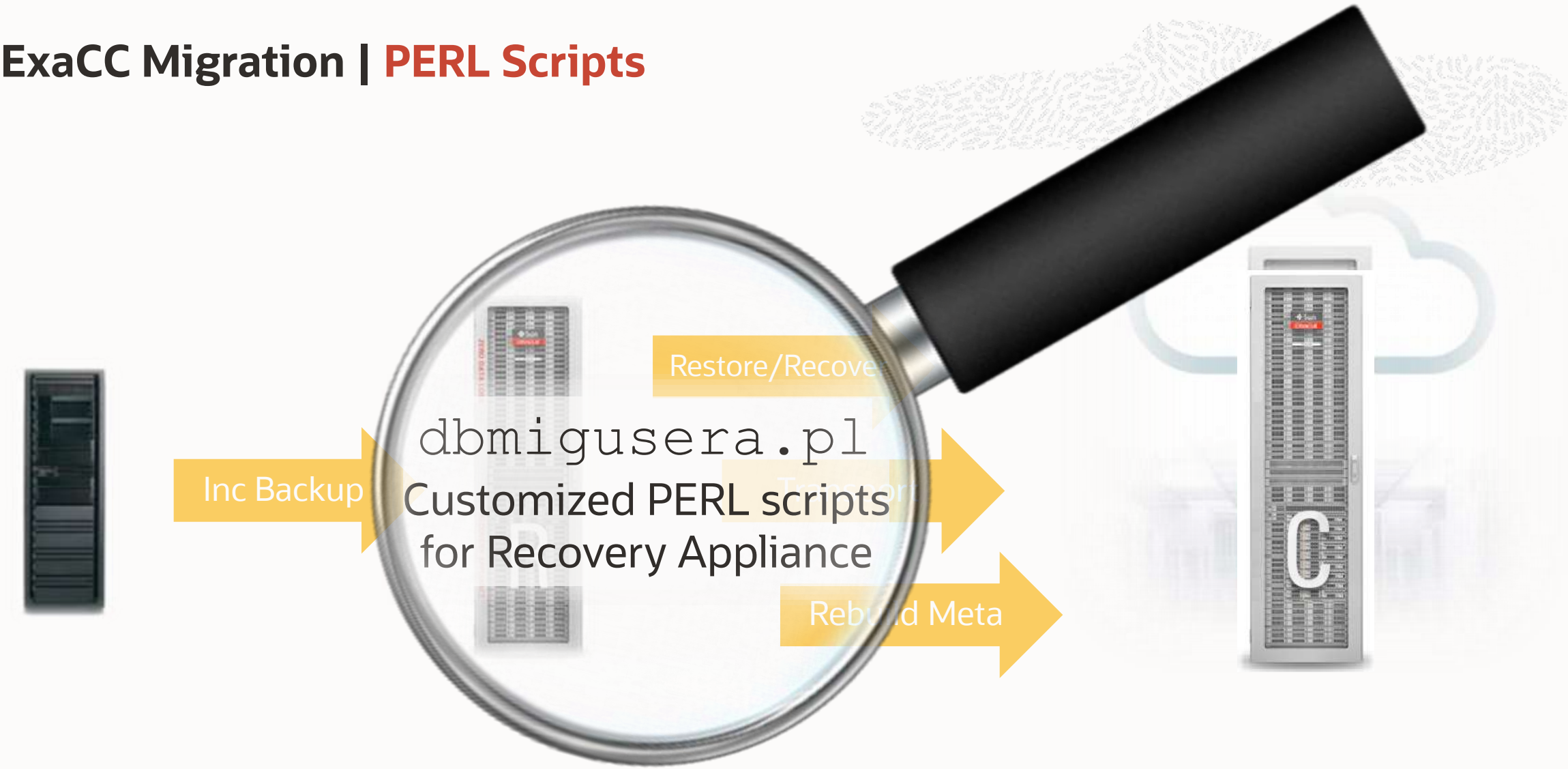


Migration Details

Part 1: The ZDLRA



ExaCC Migration | PERL Scripts

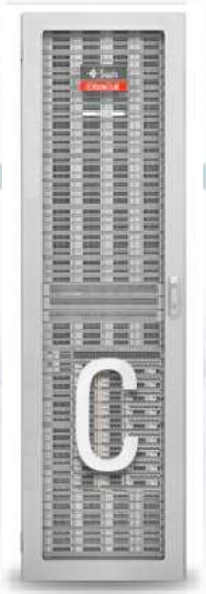


ExaCC Migration | libra.so

Install most recent `libra.so`

- MOS Note: 2219812.1
  ZDLRA: Download new sbt library

Recovery Appliance sbt library download locations:
==========
RA HPUX-IA64
RA Linux64
RA AIX-PPC64
RA Solaris-Sparc64
RA ZLinux64
RA Windows64
RA SolarisX64



ExaCC Migration | dbmigusera.pl

Download package `dbmigusera.pl` incl. `xtt.properties`

- MOS Note: 2460552.1 - Cross Platform Database Migration using ZDLRA
- Deploy package

ExaCC Migration | xtt.properties

Customize `xtt.properties`

```
# SBT parameter configuration to be used for restore and recover operations
sbtlibparms="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, …')"

# The number of datafiles that will be restored / recovered in parallel.
resparallel=16

# The list of tablespaces to be migrated
ttsnames=T14_1,T14_2,T14_3,T14_4,T14_5,T14_6,T14_7,T14_8,T14_9,T14_10,T14_11

# The directory to which the datafiles will be restored
storageondest=+DATAC1/dbmig/datafile

sourceplatid=2
dbid=4173218531
retrycount=2
```

ExaCC Migration | Level 0 Backup - Restore and Recover

Level 0 Backup

Restore/Recover

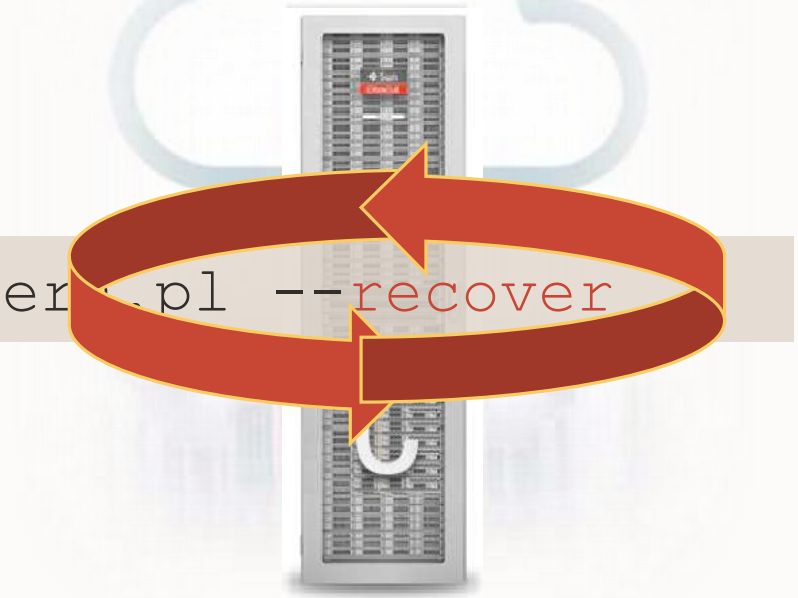
```
$ perl dbmigusera.pl --restore …
```

ExaCC Migration | Level 1 Backup - Restore and Recover

Inc Backup

Restore/Recover

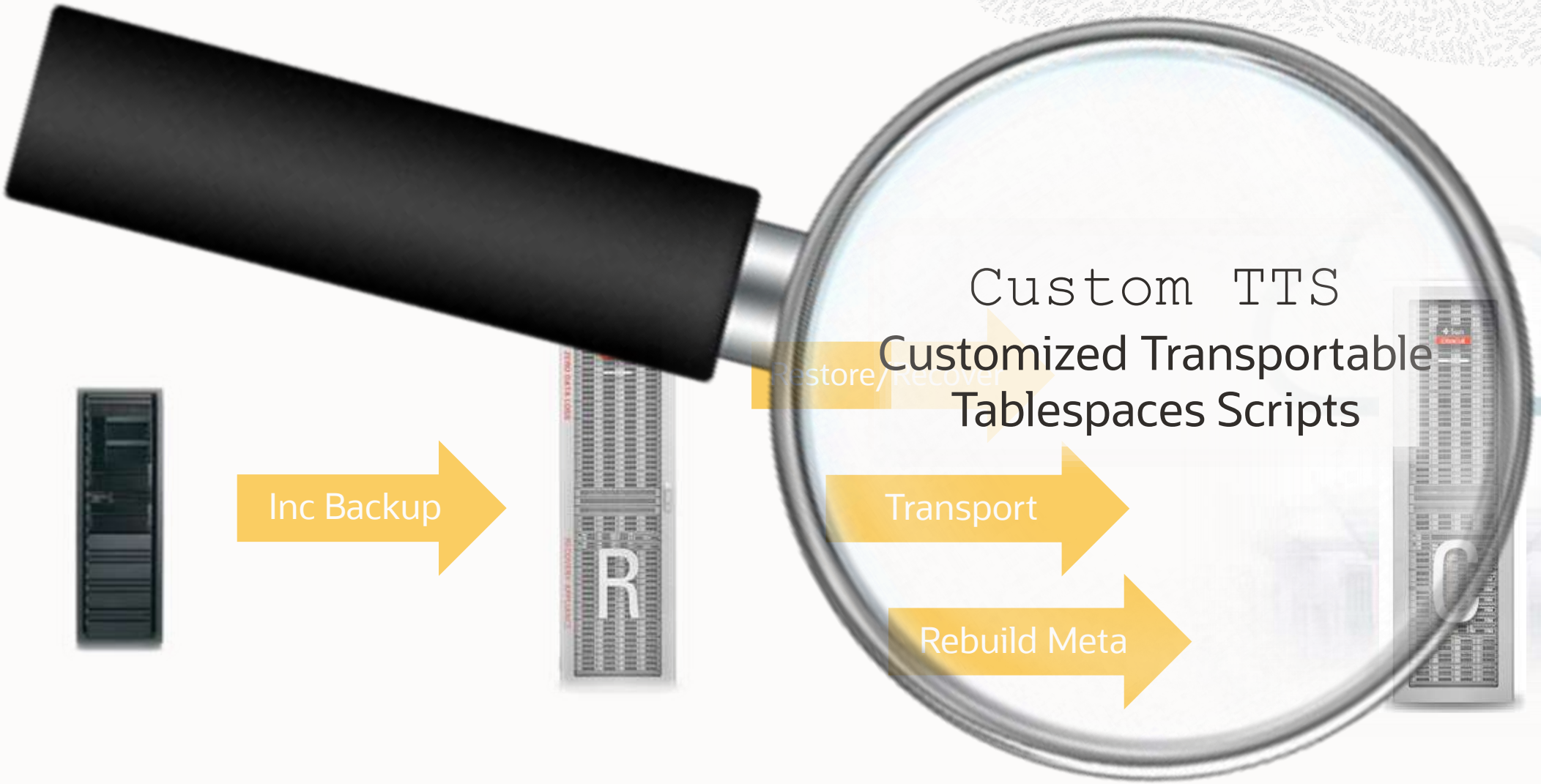
```
$ perl dbmiguser.pl --recover
```



Migration Details

Part 2: Transportable Tablespaces to ExaCC

ExaCC Migration | Transportable Tablespaces



ExaCC Migration | Check script for source

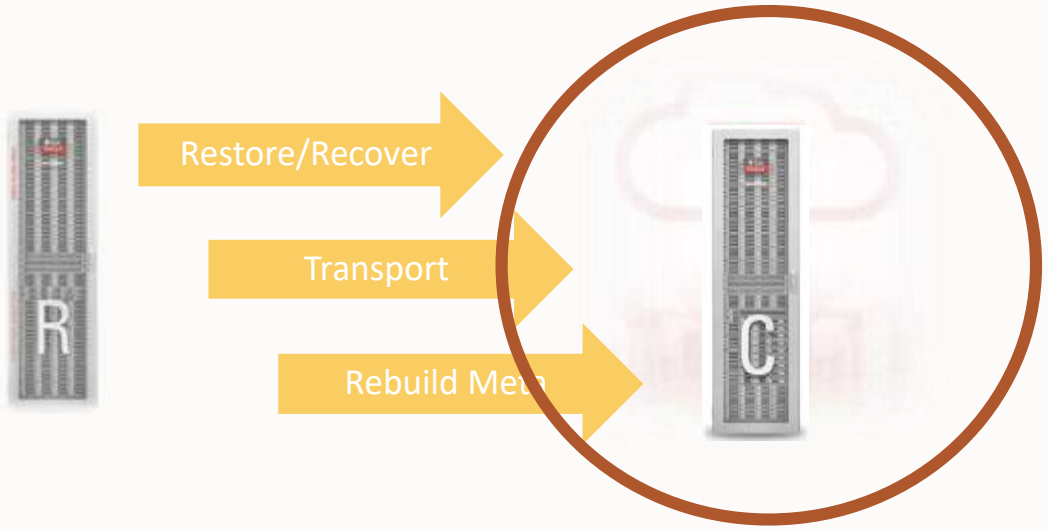
Check script

- Checks for objects in SYSTEM tablespace
- Size of database
- Tablespaces
- Object count meta objects
- And more …

AWR extract

Plan Capture

ExaCC Migration | Prep Work on ExaCC



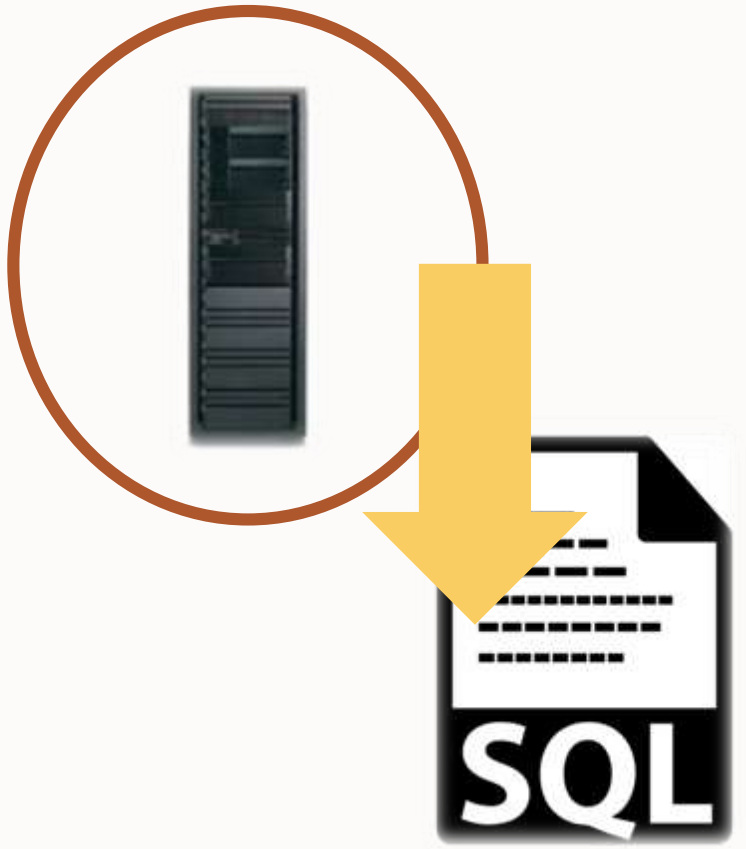
Build a new database

- As PDB
- Identical character set
- Identical national character set
- Identical time zone version

Adjustments

- Rename `USERS` tablespace to `USERS_DROP_ME_LATER`

ExaCC Migration | Prep Work on Source

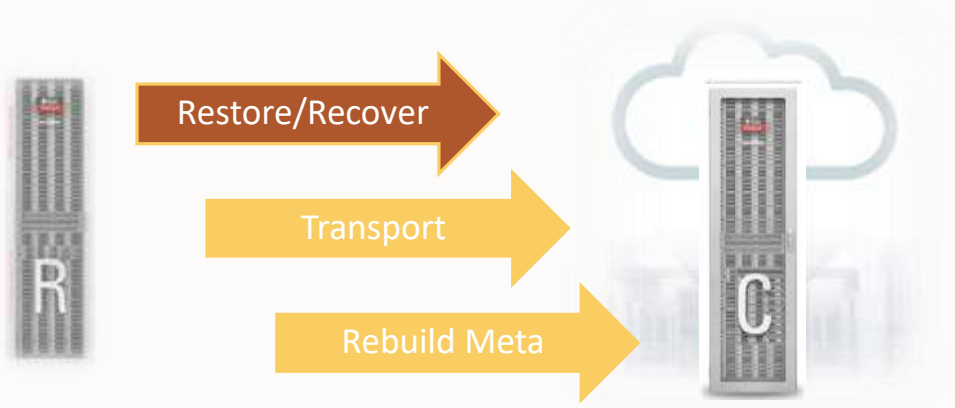


Move tables out of SYSTEM TS

Generate scripts

- Set tablespaces read/only, and reverse
- Create dummy tablespaces, and reverse
- Create temporary tablespaces
- Create User, create Profiles
- Create Grants, Roles, Directories
- Grants to create MVs, DB-Links
- Create Functions used in Tables/Indexes
- Create Global Temporary Tables
- `xtt.properties` tablespace list

ExaCC Migration | Inc Backup Restore



Level 0 backup restore

- `$ perl dbmigusera.pl --restore`

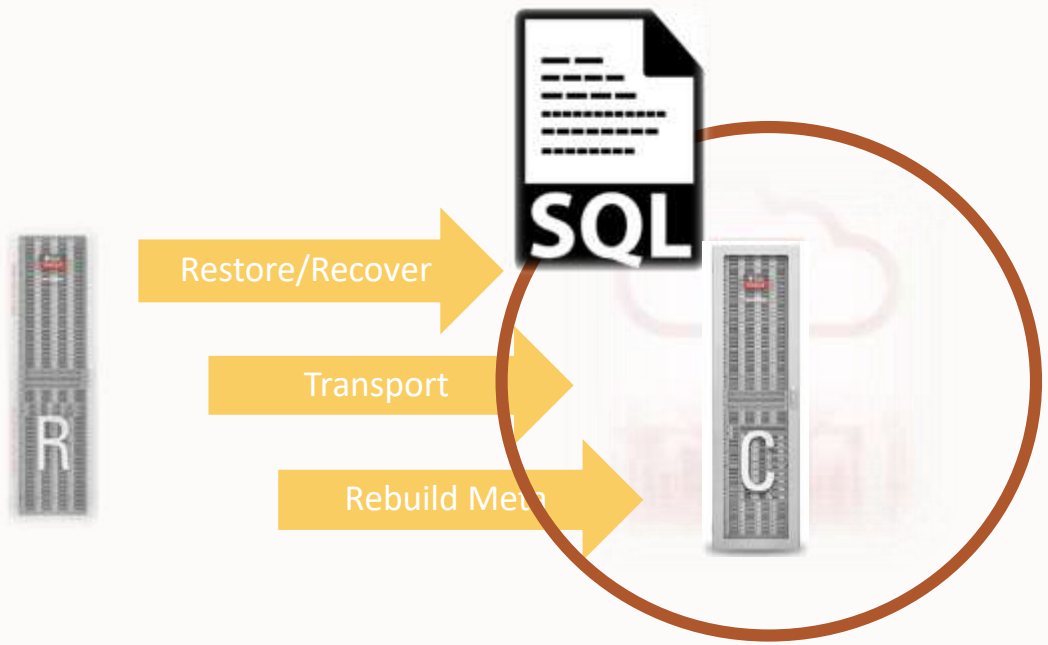
Level 1 backup restore/recover

- `$ perl dbmigusera.pl --recover`

Repeat …

No downtime required

ExaCC Migration | Prep Work on ExaCC



Create

- Profiles
- Dummy tablespaces
- Temporary tablespaces
- Users
- Directories

Drop

- Dummy tablespaces

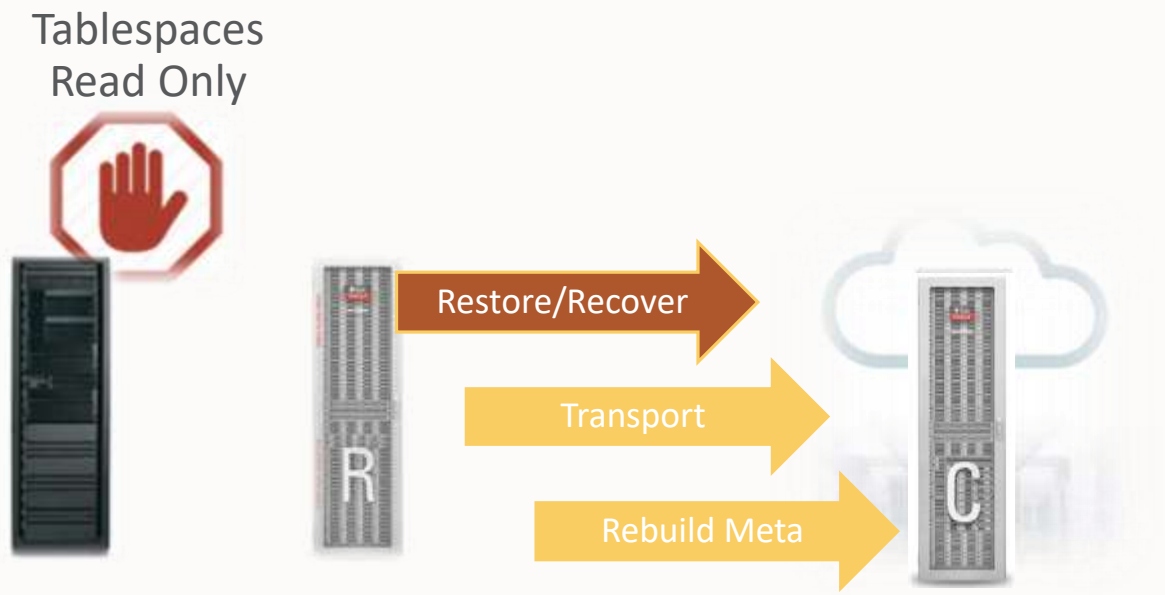
Create

- Grants, Roles, Directories, Functions for tables/indexes

From dfcopy.txt (xtt PERL):

- Generate TRANSPORT_DATAFILES strings

ExaCC Migration | Final Inc Backup-Restore

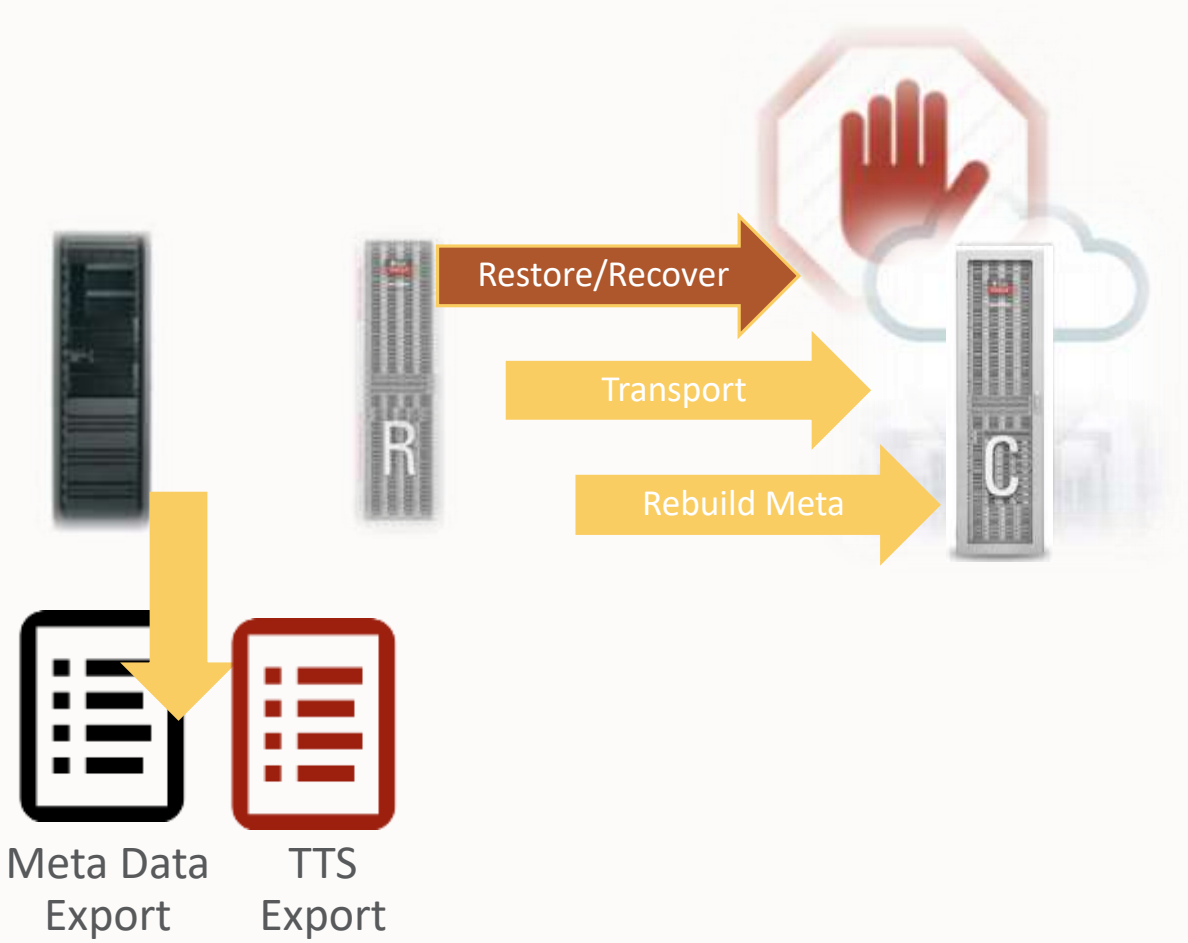


Level 1 backup restore/recover

- `$ perl dbmigusera.pl --recover`

Downtime!

ExaCC Migration | Export TTS and Meta



Metadata export

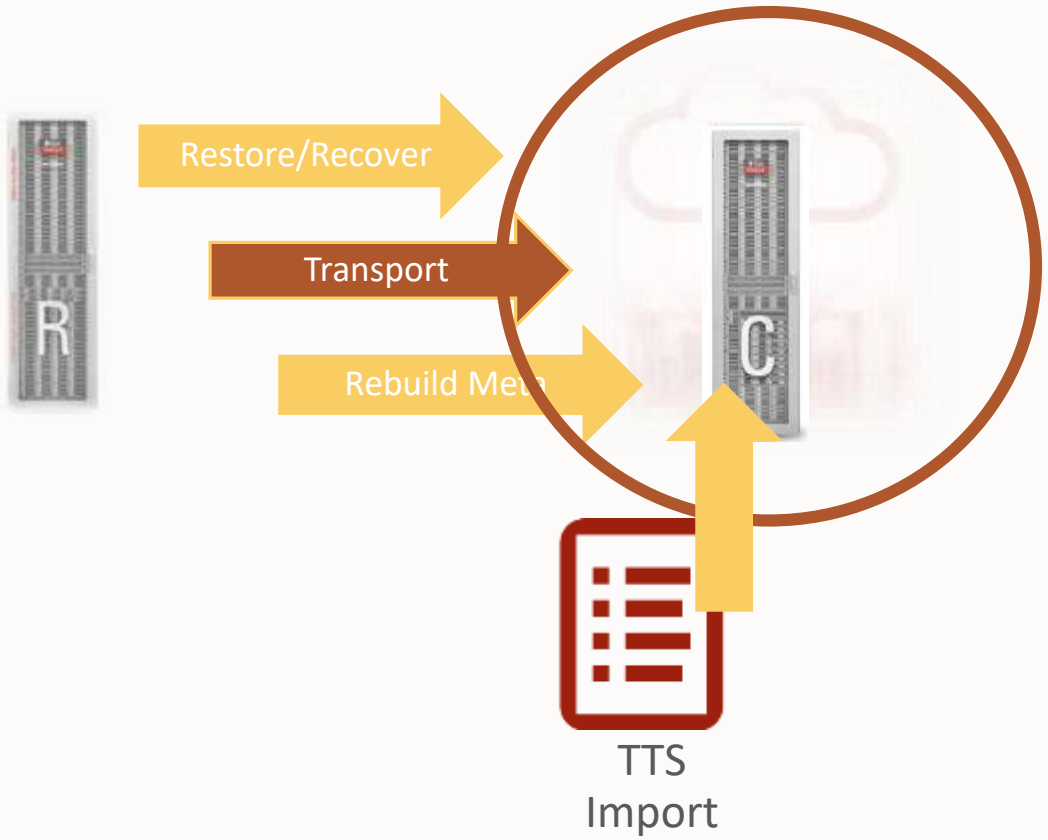
- function, package, procedure, database_link, sequence, view, synonym

TTS export

Important

- GATHER_SCHEMA_STATS('SYS')
- GATHER_SCHEMA_STATS('SYSTEM')

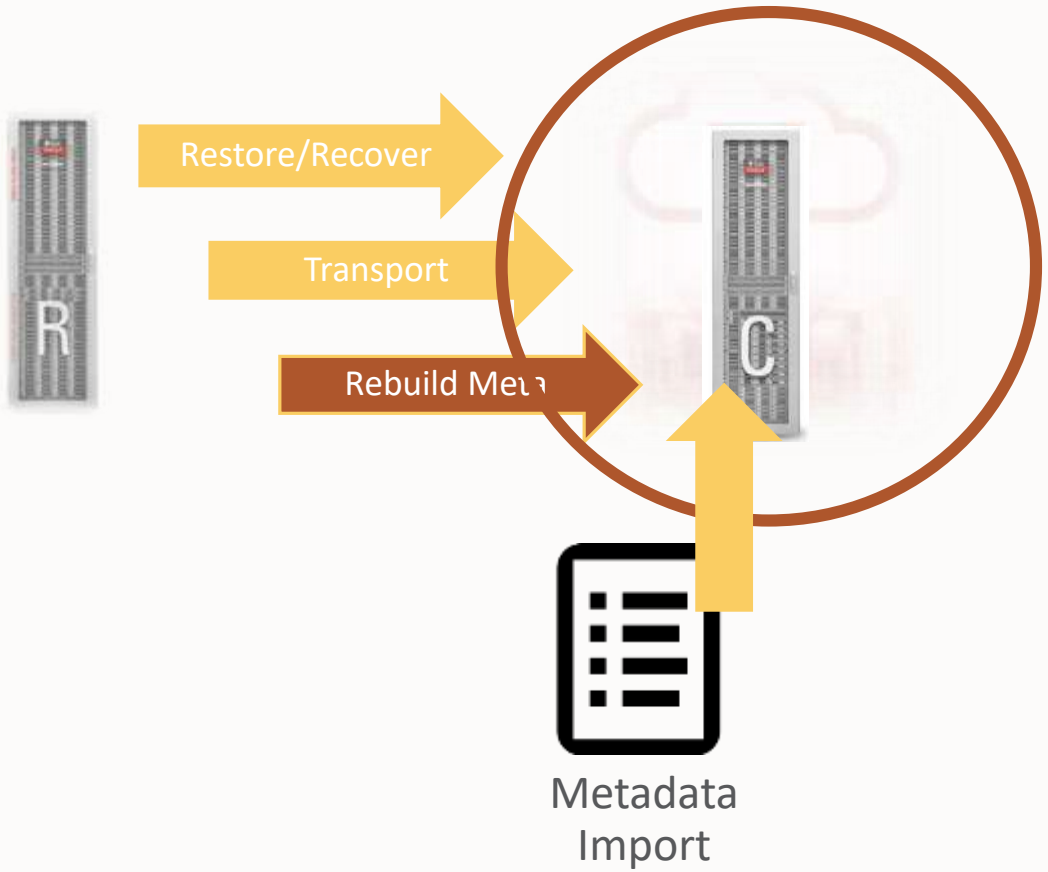
ExaCC Migration | TTS Import



TTS Import

- Afterwards tablespaces are attached

ExaCC Migration | Rebuild of Meta Info



Meta Import

- Rebuilds:
  - Functions
  - Packages
  - Procedures
  - Database links
  - Sequences
  - Views
  - Synonyms

ExaCC Migration | Check Script - Comparison

Clean up

Run check script

- Comparison Before/After

Online encryption on ExaCC

METRO NOM ABOUT BLOG CONTACT WORK APIS

Real World Check

One of the largest retailers of the world migrates >1500 databases to ExaCC

THE BIGGEST SOFTWARE COMPANY YOU NEVER HEARD ABOUT

Customer Case | Metro|nom

- Customer
- Project 2018
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Metro|nom

- 2000 employees
- IT Services, IT Solutions
- "*The biggest software company you never heard about*"

Belongs to Metro AG

- 4th largest retailer globally
- 150,000 employees
- €37 billion revenue
- 25 countries

Customer Case | Metronom

- Customer
- Project 2018
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Migrate and consolidate >1500 databases to ExaCC

- Mostly on IBM AIX
- Project timeline: >4 years

Develop a universal migration solution

- With minimal manual interaction

Customer Case | Metro|nom

- Customer
- Project 2018
- Constraints
- Preparation
- Migration
- Success?
- Remarks

13 DBAs for 2500 databases

Upper Management

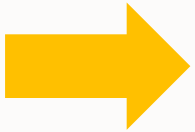
One Click Migration

No Downtime!!!

Experts

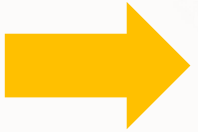
Migration in 3 Steps

Pretasks



Migration

Minimum Downtime!!!



PostTasks

Customer Case | Metro|nom

- Customer
- Project 2018
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

PERL scripts and newest `libra.so` on ZDLRA

Customized Transportable Tablespace scripts

- Script automation added

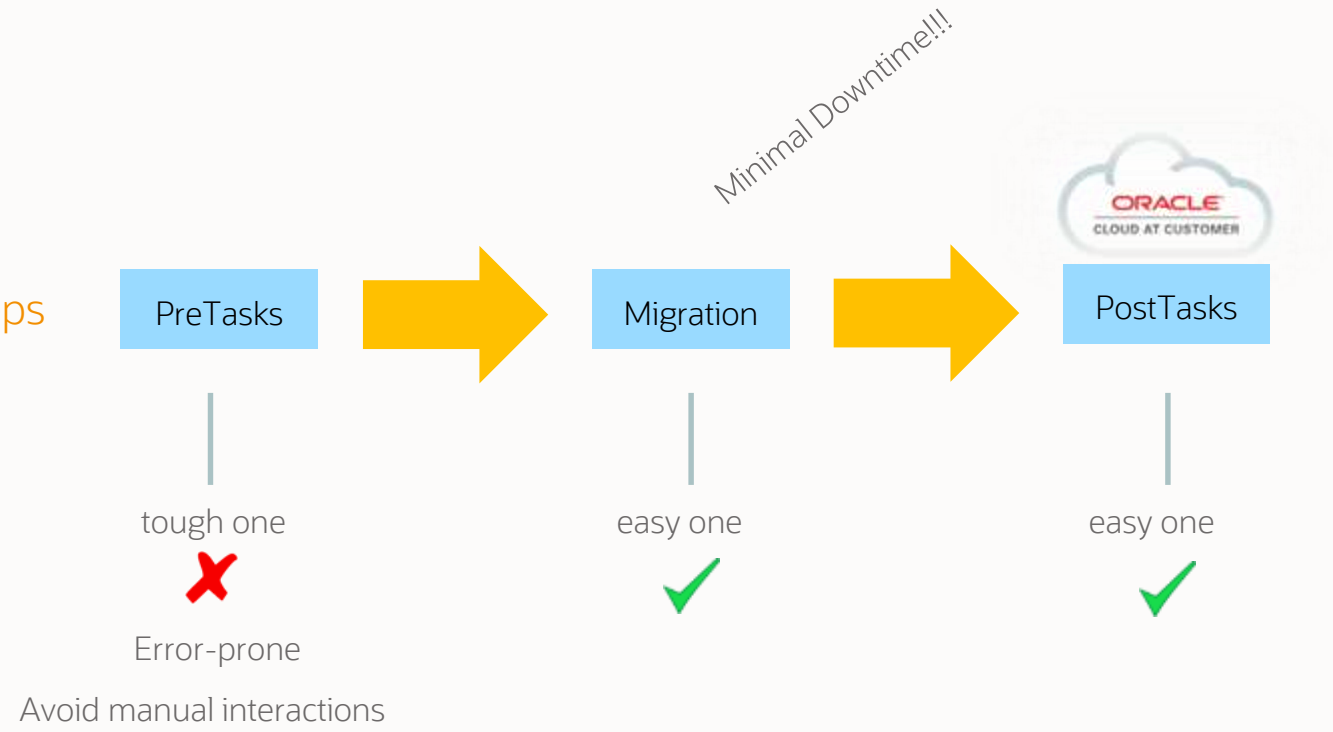
ExaCC setup, patches etc.

ExaCC Multitenant setup

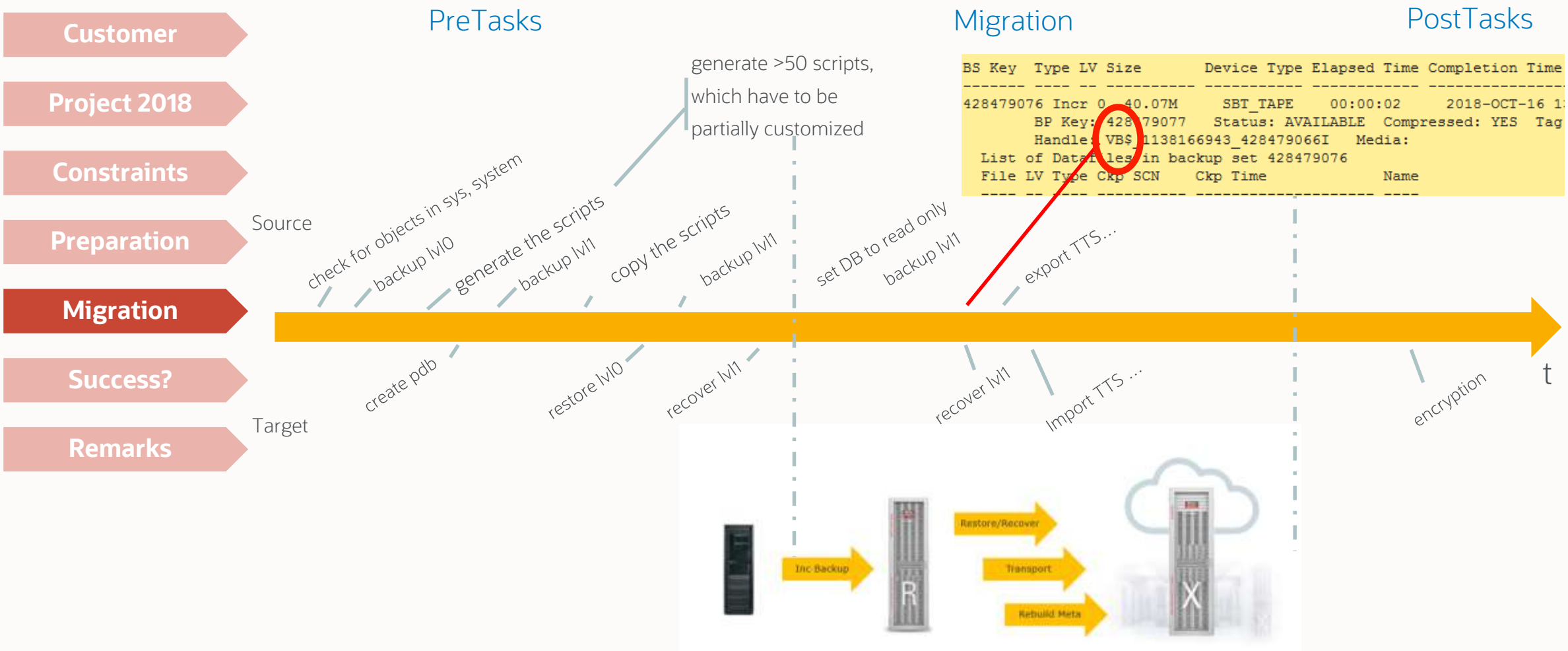
Customer Case | Metronom

- Customer
- Project 2018
- Constraints
- Preparation
- **Migration**
- Success?
- Remarks

Migration in 3 Steps



Customer Case | Metro|nom



Customer Case | Metro|nom

- Customer
- Project 2018
- Constraints
- Preparation
- Migration
- **Success?**
- Remarks

Plenty of database migrated
From non-CDB to PDB

Project is ongoing

Customer Case | Metro|nom

- Customer
- Project 2018
- Constraints
- Preparation
- Migration
- Success?
- **Remarks**

Best practice recommendations by the customer

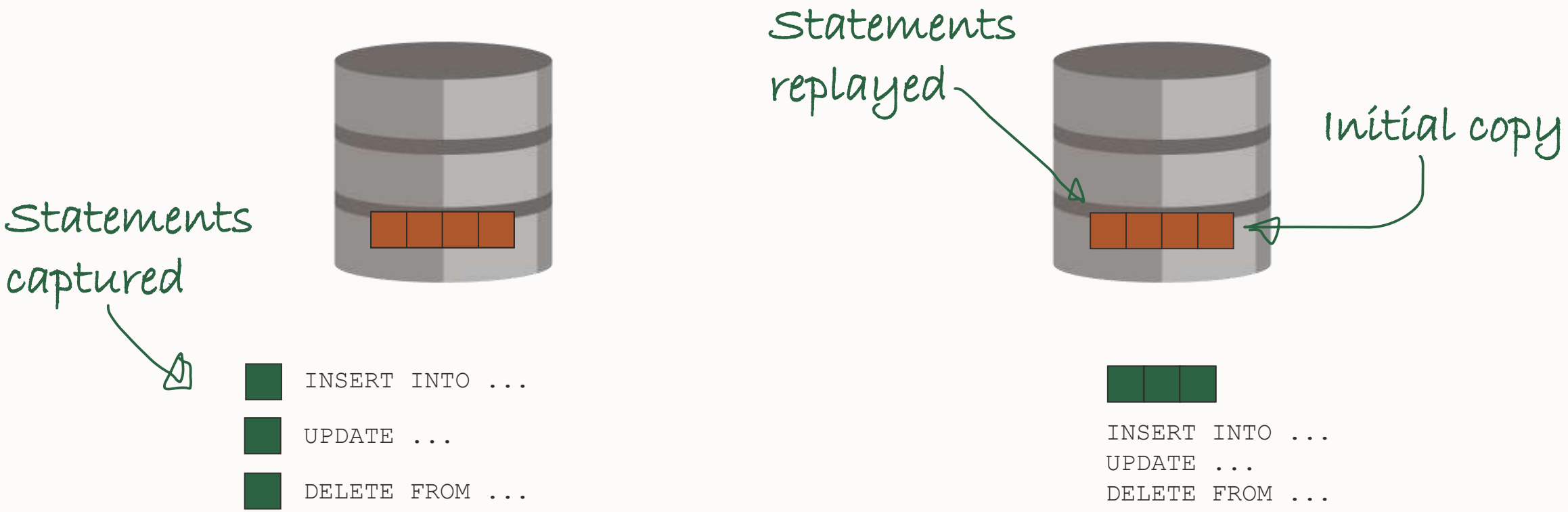
- Automate your pre-tasks
  - Avoid manual modifications of generated scripts to avoid human error
- Complete pre-tasks 2-3 days before actual migration
- Don't concentrate on databases with issues
  - Go for the "as many as possible"
- Always have a fallback in place
- Check the load of your ZDLRA
- Scan your logfiles - and filter out uncritical errors
- Order of imports are important
  - You can't grant privileges when the user does not exist
- Install the most recent `libra.so`

Photo by Chris Leipelt on Unsplash

Migration Strategies

Oracle GoldenGate

GoldenGate | Explained



GoldenGate

Advantages

- True zero downtime
- Extremely flexible
- Handles any migrations
- Handles upgrades from old versions
- Great fallback capabilities
- Active Data Guard included in OGG license

Considerations

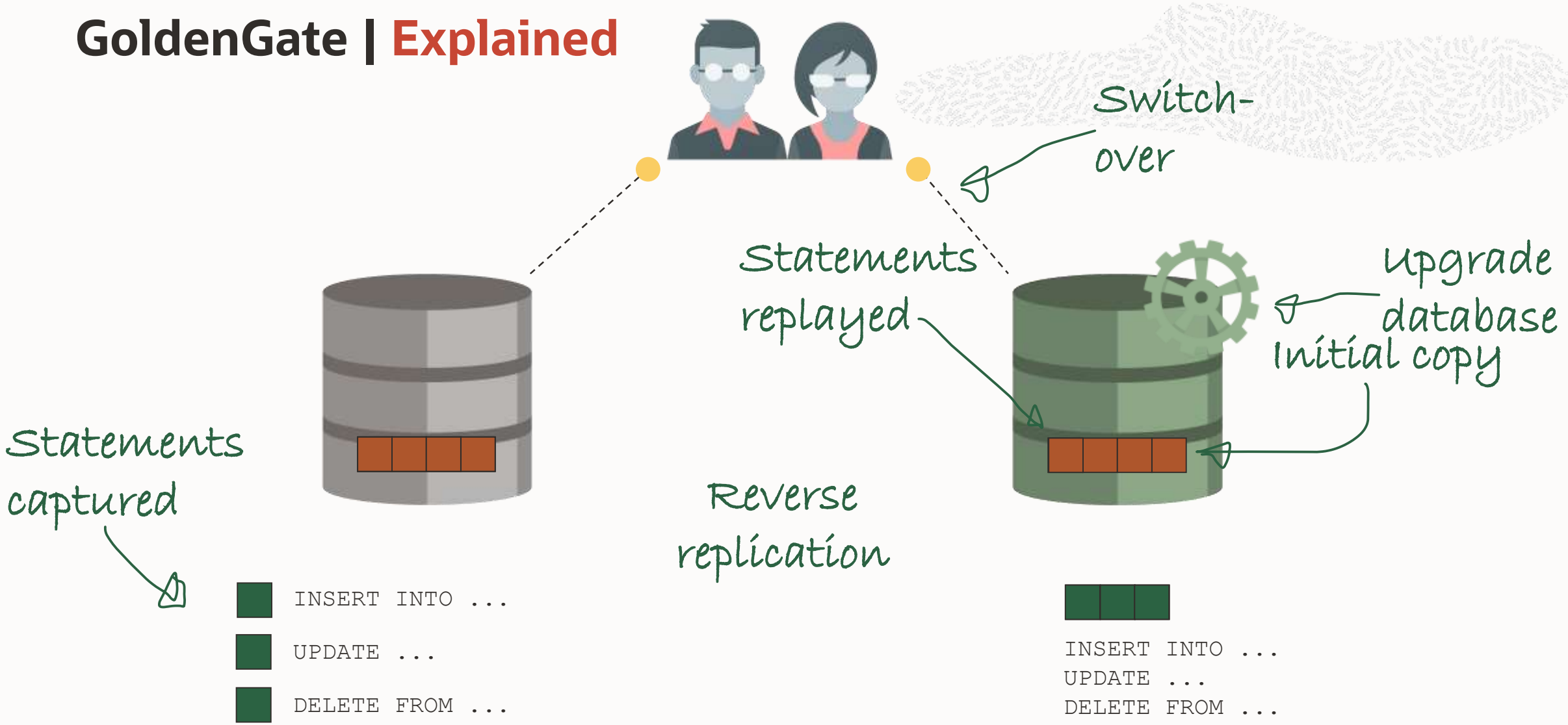
- Additional license required
- Added complexity

GoldenGate | Oracle Cloud Infrastructure

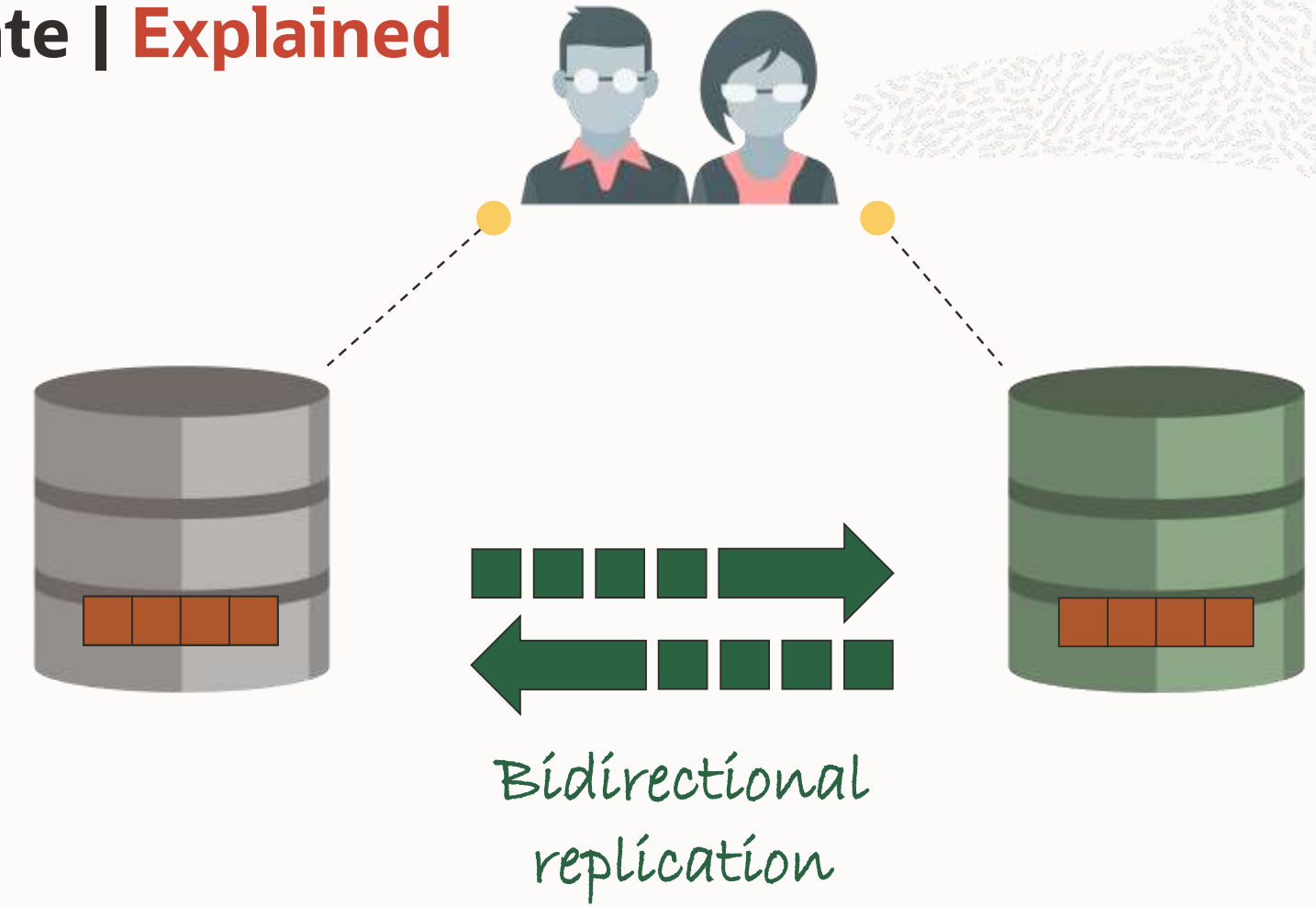
Free when migrating to Oracle Cloud Infrastructure

- Using OCI Marketplace image
- Terms apply
- Available until end of 2020

GoldenGate | Explained



GoldenGate | Explained



GoldenGate | Data Pump Integration

No longer needed to specify from which CSN replication should start

New Replicat parameter
`DBOPTIONS ENABLE_INSTANTIATION_FILTERING`

Requires Oracle GoldenGate 12.2

MOS Note: 1276058.1

```
SQL> select source_object_name,
instantiation_scn from
dba_apply_instantiated_objects where
source_object_owner = 'APPS' ;

SOURCE_OBJECT_NAME INSTANTIATION_SCN
-----------------------------------
TCUSTMER                  829723224
TCUSTORD                  829723223



2017-07-17 15:02:51 INFO OGG-10155
Instantiation CSN filtering is enabled
on table APPS.TCUSTMER at CSN
829,723,224.

2017-07-17 15:02:51 INFO OGG-10155
Instantiation CSN filtering is enabled
on table APPS.TCUSTORD at CSN
829,723,223.
```

GoldenGate | Technical Brief

Technical brief:
Zero Downtime Database Upgrade Using Oracle GoldenGate

Migration with Oracle GoldenGate

Amadeus
OOW Presentation 2012

Source: https://amadeus.com/images/en/amadeus/offices/amadeus-madrid-sign.jpg

Customer Case | Amadeus

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Amadeus is a leading transaction processor for the global travel and tourism industry

DISTRIBUTION BUSINESS

711 airlines
110,000+ hotel properties
30 car rental companies
50+ cruise and ferry lines
207 tour operators
24 insurance companies
95 railways

IT SOLUTIONS

Inventory
Departure Control
e-Commerce

Airlines
Airports
Hotels
Rail

20,000+ tx/sec (peak)
< 0.3 sec response time
10 Petabytes of storage
3+ million net bookings/day
> 1 billion tx/day

** All numbers are from 2012*

Customer Case | Amadeus

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- Success?
- Remarks

Migrate Oracle 10g production databases to Oracle 11g

Migrate to new HW and/or OS platform

Source		Target
Oracle 10.2.0.3 RAC HPUX v2	→	Oracle 11.2.0.2/3 RAC HPUX v3
	→	Oracle 11.2.0.2/3 RAC RHE Linux
Oracle 10.2.0.3 Single Instance HPUX v2	→	Oracle 11.2.0.2/3 RAC One RHE Linux

Customer Case | Amadeus

- Customer
- Project 2012
- **Constraints**
- Preparation
- Migration
- Success?
- Remarks

Fixed quarterly outage windows

Maximum of 5 minutes database downtime

No service impact outside the outage window

Endian change: HP-UX ⇨ to Linux (big ⇨ little endian)

Possibility of fallback during and after the outage

High volume of DB changes (redo of up to 20MB/sec)

Large database sizes (up to 14TB)

Possibility for physical re-organization
- Fresh data dictionary
- Tablespace and partitioning redesign

Customer Case | Amadeus

- Customer
- Project 2012
- Constraints
- **Preparation**
- Migration
- Success?
- Remarks

In-depth proof of concept (supported by Oracle)

- Focusing on functional aspects
- Focusing on data volume

Standardized migration process model with timeline

Home-made scripts and procedures to support setup, monitoring, tuning and switch over

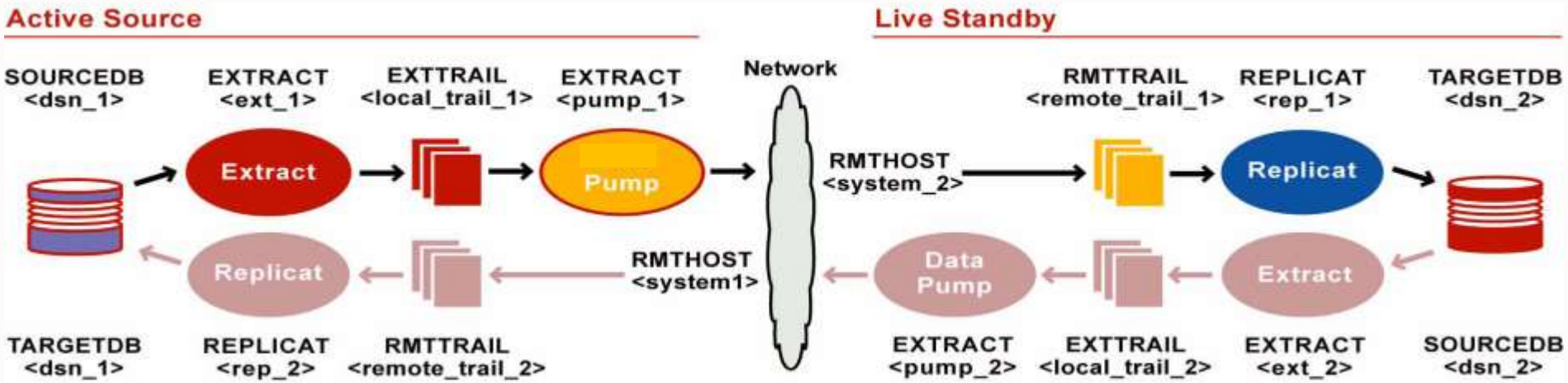
Training of in-house specialist supporting the DBAs

Customer Case | Amadeus

- Customer
- Project 2012
- Constraints
- Preparation
- **Migration**
- Success?
- Remarks

Instantiation of new 11g database: expdp from Physical Standby

Installation, configuration, tuning of GG replication



Comparison of source/target DB content (Veridata)

Rehearsals of switch over and fallback

Switch over: Stop replication / Start reverse-replication

Customer Case | Amadeus

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- **Success?**
- Remarks

15 databases successfully migrated, so far (*Oct 2012*)

Source	Target	Migrated
Oracle 10.2.0.3 RAC HPUX v2	Oracle 11.2.0.2/3 RAC HPUX v3	6 ✓
	Oracle 11.2.0.2/3 RAC RHE Linux	3 ✓
Oracle 10.2.0.3 Single Instance HPUX v2	Oracle 11.2.0.2/3 RAC One RHE Linux	6 ✓

Switchover duration: 2-6 minutes

No fallback performed

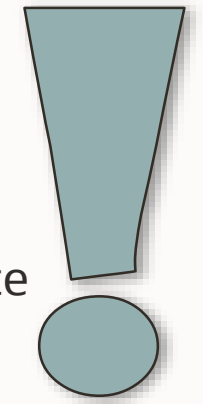
Customer Case | Amadeus

- Customer
- Project 2012
- Constraints
- Preparation
- Migration
- Success?
- **Remarks**

The concept proved to handle a smooth and secure migration across different DB versions and HW/OS platforms

To be considered …

- Instantiation of target database (incl. Plan Stability)
- Customized GG setup per database
- Handling of unsupported data types (e.g. ANYDATA)
- Impact of supplemental logging on source DB
- Effort of tuning GG for DBs with high DML rate (e.g. parallel replicate processes)



Oracle GoldenGate | Further Information

TB: Zero Downtime Database Upgrade Using Oracle GoldenGate

MOS Note: 1448324.1
GoldenGate Integrated Capture and Integrated Replicat Healthcheck Script

MOS Note: 2193391.1
Latest GoldenGate/Database (OGG/RDBMS) Patch recommendations

Complete Database Profile OGG readiness check

- MOS Note:1298562.1:
  Oracle GoldenGate database Complete Database Profile check script for Oracle DB (All Schemas) Classic Extract

Check OGG readiness for Schema Only

- MOS Note: 1296168.1
  Oracle GoldenGate database Schema Profile check script for Oracle DB

Oracle Streams | Desupport in Oracle 19c

> Starting in Oracle Database 19c, Oracle Streams is desupported. Oracle GoldenGate is the replication solution for Oracle Database.
>
> Note that Oracle Database Advanced Queuing is not deprecated, and is fully supported in Oracle Database 19c. Oracle Streams did not support features added in Oracle Database 12c (12.1) and later releases, including the multitenant architecture, LONG VARCHAR, and other new features. Oracle Streams replication functionality is superseded by GoldenGate.

Release 19, Database Upgrade Guide

Summary

Further Information

Migration | Techniques

	Data Pump	Data Guard	Transient Logical Standby	Transportable Tablespaces	Full Transportable Export/Import	Incremental Backups	GoldenGate
Standalone Migration	+	+	+	+	+		
Simplicity	+	+	-	-	o	o	-
Downtime	-	+	+	o	o	+	+
Version Change	+	+	+	+	+		+
Same-Endianness OS Change	+	+	o	+	+	+	+
Big/Little Endianness OS Change	+			+	+	+	+
Same Hardware	+	+		+	+	+	+
Hardware Exchange	+	+	+	+	+	+	+
non-CDB to CDB/PDB	+			+	+	+	+
Encryption	+						+
Fallback Option	+	o	o	o	o		+
Character Set Change	+						+

Migration | Keep It Simple

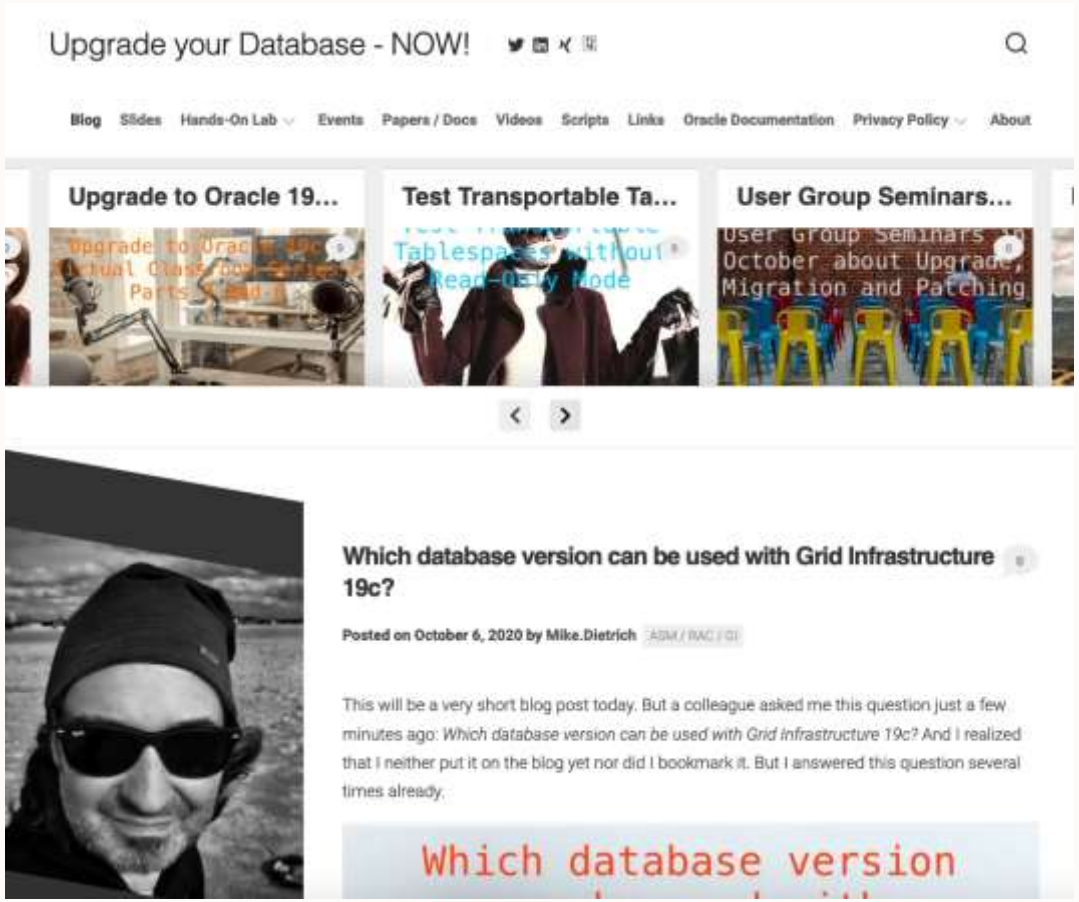
Photo by Jonathan Velasquez on Unsplash

Thursday, October 15, 2020 at 11:00h CEST

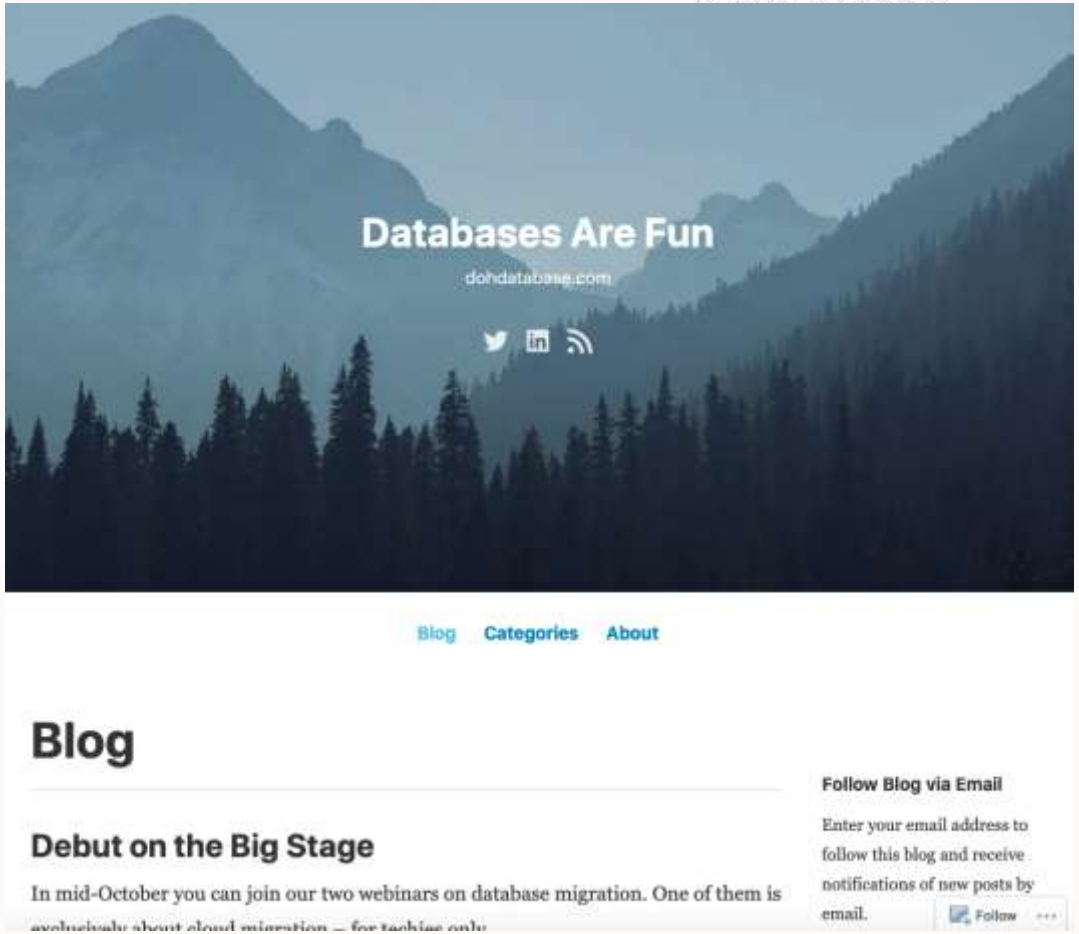
Migration to the Cloud

For Techies only

https://MikeDietrichDE.com



https://DOHdatabase.com



Thank you!