

Migration

Non-CDB to PDB

Author	김누리
Creation Date	2020-12-28
Last Updated	
Version	1.0
Copyright(C) 2018 GoodusData Inc.	
All Rights Reserved	

Document Reference

Change Record

Date	Author	Version	Change Reference
2020-12-28	김누리	1.0	이전 문서 없음

Reviewers

Name	Position

Distribution

Copy No.	Name	Location

Contents

Chapter	PAGE
1. Migration Non-CDB to PDB	4
1.1. 개요	4
1.2. PDB 로 이관 방법	4
2. Migration Method	7
2.1. Datapump	7
2.1.1. Prerequisite	7
2.1.2. Migration Non-CDB to PDB Using Datapump	7
2.2. DB_LINK	12
2.2.1. Prerequisite	12
2.2.2. Migration Non-CDB to PDB Using DB Link	13
2.3. TTS	17
2.3.1. Prerequisite	17
2.3.2. Migration Non-CDB to PDB Using TTS	17
2.4. Hot Backup	24
2.4.1. Prerequisite	24
2.4.2. Migration Non-CDB to PDB Using HotBackup(Clone DB)	25
3. Reference	34

1. Migration Non-CDB to PDB

1.1. 개요

Oracle 12C New Feature 중의 하나인 Multitenant 는 Oracle 21c 부터 필수로 사용하게 되었습니다. 이 문서에서는 Oracle 21c 로의 업그레이드를 위한 non-cdb to cdb 로의 다양한 이관 방법을 설명 하였습니다.

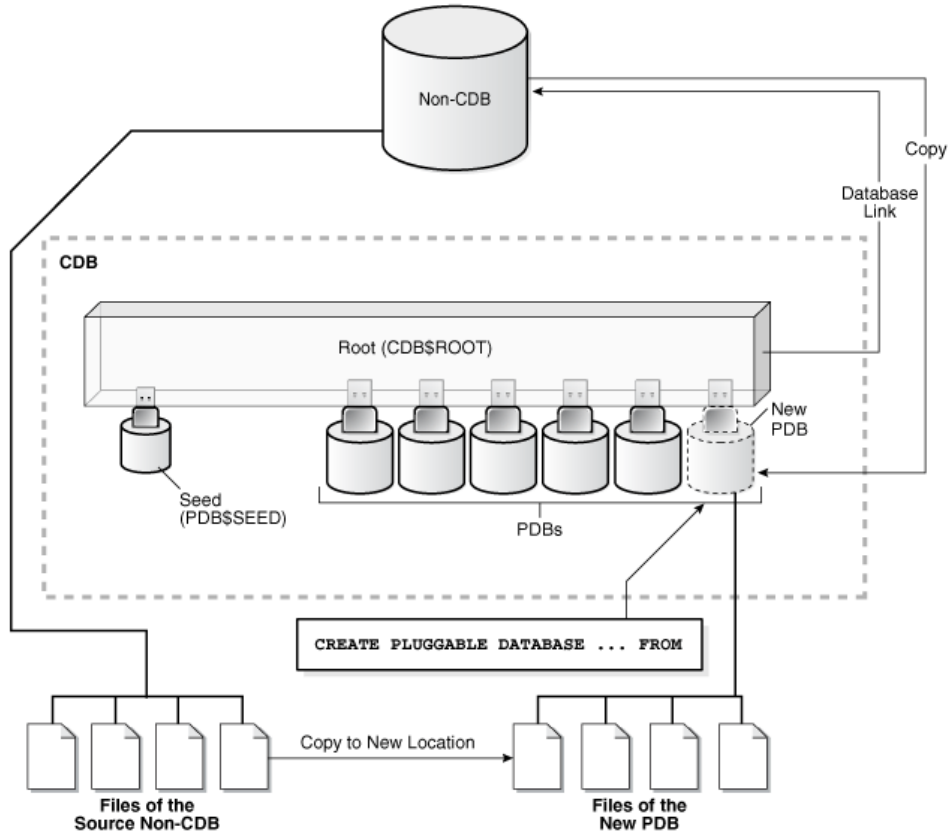
1.2. PDB로 이관 방법

Using Oracle Data Pump

- Oracle 에서 제공하는 Datapump Utility 를 이용하여 PDB 로 데이터를 이관하는 방법입니다.
- CDB 에 빈 PDB 를 생성 한 후 Oracle Data Pump 의 expdp/impdp 를 사용하여 데이터를 PDB 로 이동할 수 있습니다. Option 에 따라 Full, Schema, Tablespace, Table 단위로 데이터 이관이 가능하며 사용자 테이블 스페이스의 인덱스 구조를 다시 생성 할 필요가 없기 때문에 작업시간을 줄일 수 있습니다.
- Source OS 와 Target OS 의 Endian 이 동일해야 합니다.

Using Oracle Database Link

- CDB 에 Non-CDB 를 바라보는 dblink 를 생성하여 PDB 를 원격으로 구성할 수 있습니다.
- Non-CDB 에서 복제하는 경우 12.1.0.2 이상에서만 사용 가능하며 Source OS 와 Target OS 의 Endian 이 동일해야 합니다.



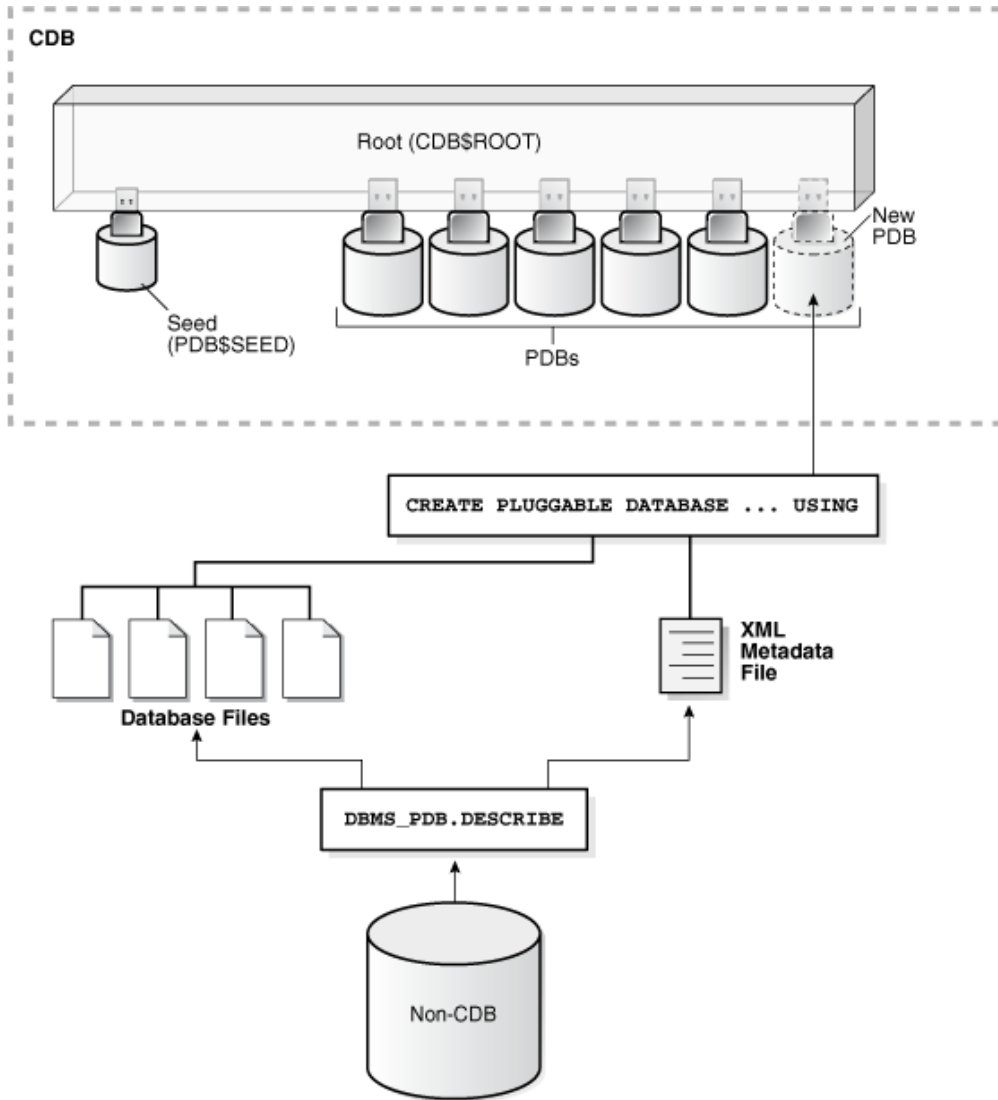
- Description of "Figure 38-5 Creating a PDB by Cloning a Non-CDB"

Using TTS

- Transportable Tablespaces(TTS)를 이용하여 PDB 로 이관하는 방법은 기존과 동일하며 impdp 를 진행 할 때 이관을 진행할 PDB 의 TNS 를 이용하여 진행한다는 점만 다릅니다.
- Source OS Platform 과 Target OS Platfrom 의 Endian 이 다를 경우 Rman convert 를 이용해서 변환시켜주어야 하며 11.2.0.3 이상 버전부터 Full TTS 를 사용할 수 있습니다.

Using Clone Database

- Hot Backup 을 이용한 Clone DB 구성하여 PDB 로 변경할 수 있습니다.
- 이때, DBMS_PDB Package 를 이용하여 XML 파일을 만들어 Non-CDB 를 PDB 로 Convert 할 수 있으며 Source OS 와 Target OS 의 Endian 이 동일해야 합니다.



2. Migration Method

2.1. Datapump

Oracle 에서 제공하는 Datapump Utility 를 이용하여 PDB 로 데이터를 이관하는 이관합니다. 방법은 Non-CDB to Non-CDB 방법과 동일하며 impdp 를 진행 할 때 이관을 진행할 pdb 의 TNS 를 이용하여 진행한다는 점만 다릅니다.

2.1.1. Prerequisite

- 각 PDB 로 접속하여 datapump 를 진행해야합니다.
- Source DB 와 Target DB 는 호환성있는 Character Set 과 Nationl Character Set 을 사용해야 합니다.
- 다른 Endian 을 가질 경우 불가합니다.

2.1.2. Migration Non-CDB to PDB Using Datapump

■ 사전 작업

- Target DB 의 Instance_name, Version, Edition 확인
- Characterset 확인
- Source TableSpace 및 Datafile 생성
- Source User Profile 설정
- Source User 생성 및 권한 부여

Common User :

사용자가 모든 컨테이너 (루트 및 모든 PDB)에 있습니다.

현재 컨테이너는 루트 컨테이너 여야합니다.

사용자 이름은 모든 컨테이너에서 고유해야합니다.

일반 사용자의 사용자 이름은 "C ##"또는 "c ##"로 시작해야하며 ASCII 또는 EBCDIC 문자 만 포함해야합니다.

```
ex> CREATE USER c##test_user1 IDENTIFIED BY password1 CONTAINER=ALL;  
GRANT CREATE SESSION TO c##test_user1 CONTAINER=ALL;
```

Local User :

사용자가 특정 PDB 에만 존재합니다. 동일한 사용자 이름이 여러 PDB 에 존재할 수 있지만 서로 관련이 없습니다.

로컬 사용자의 사용자 이름에는 "C ##"또는 "c ##"접두사가 붙어서는 안됩니다.

사용자 이름은 PDB 내에서 고유해야합니다.

CONTAINER=CURRENT 현재 컨테이너가 PDB 인 경우 기본 설정이므로 절을 지정 하거나 생략 할 수 있습니다.

```
ex> CREATE USER test_user2 IDENTIFIED BY password1 CONTAINER=CURRENT;
```

```
GRANT CREATE SESSION TO test_user3 CONTAINER=CURRENT;
```

- Temp 및 Redolog Size 확인
- tnsnames.ora , sqlnet.ora 확인 및 수정

■ 위와 같은 사전 작업 시, PDB 에서 진행해주어야 합니다

■ tns 를 이용한 pdb 접속

```
$ sqlplus sys/"PASSWORD"@TNS as sysdba
```

```
SQL>show con_name
```

```
CON_NAME
```

```
-----
```

```
PDB
```

■ cdb 로 접속하여 pdb 로 세션 변경

```
SQL> show pdbs
```

```
CON_ID      CON_NAME          OPEN MODE  RESTRICTED
```

```
-----
```

```
2          PDB$SEED         READ ONLY  NO
```

```
3          PDB              READ WRITE NO
```

```
SQL>
```

```
alter session set container=PDB;
```

```
SQL>show con_name
```

```
CON_NAME
```

```
-----
```

```
PDB
```

■ Directory 생성

Non-CDB, PDB 모두 Datapump 를 이용할 때 사용할 Directory 를 생성해줍니다. Oracle 유저가 읽고 쓸 수 있는 경로로 설정해주어야 합니다.

■ Directory 생성

```
SQL> create directory goodus as '/backup/goodus';
```

■ Non-CDB to PDB Migration Using Datapump

■ EXPDP IN Non-CDB


```

-모든데이터 EXPDP
$ expdp ₩"/ as sysdba₩" directory=GOODUS job_name=GOODUS_EXPDP DUMPFILE=FULL_20201214.dmp
logfile=expdp_full_20201214.log full=y
Export: Release 19.0.0.0.0 - Production on Thu Dec 17 20:11:08 2020
Version 19.7.0.0.0
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
FLASHBACK automatically enabled to preserve database integrity.
Starting "SYS"."GOODUS_EXPDP":  "/***** AS SYSDBA"
directory=GOODUSjob_name=GOODUS_EXPDP DUMPFILE=FULL_20201214.dmp
logfile=expdp_full_20201214.log full=y
Processing object type DATABASE_EXPORT/EARLY_OPTIONS/VIEWS_AS_TABLES/TABLE_DATA
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/TABLE_DATA
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/VIEWS_AS_TABLES/TABLE_DATA
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type DATABASE_EXPORT/STATISTICS/MARKER
Processing object type DATABASE_EXPORT/PRE_SYSTEM_IMPCALLOUT/MARKER
Processing object type DATABASE_EXPORT/PRE_INSTANCE_IMPCALLOUT/MARKER
Processing object type DATABASE_EXPORT/TABLESPACE
Processing object type DATABASE_EXPORT/PROFILE
Processing object type DATABASE_EXPORT/SCHEMA/USER
Processing object type DATABASE_EXPORT/RADM_FPTM
Processing object type DATABASE_EXPORT/GRANT/SYSTEM_GRANT/PROC_SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE
Processing object type DATABASE_EXPORT/SCHEMA/ON_USER_GRANT
....(생략)
Master table "SYS"."GOODUS_EXPDP" successfully loaded/unloaded
*****
Dump file set for SYS.GOODUS_EXPDP is:
  /export/goodus/FULL_20201214.dmp
Job "SYS"."GOODUS_EXPDP" successfully completed at Thu Dec 17 20:14:23 2020 elapsed 0
00:03:15

- SCHEMA 단위로 EXPDP
$ expdp ₩"/ as sysdba₩" directory=GOODUS job_name=GOODUS_EXPDP DUMPFILE=SCHEMA_20201214.dmp
logfile=expdp_schema_20201214.log SCHEMAS=GOODUS

```

```

Export: Release 19.0.0.0.0 - Production on Thu Dec 17 20:16:37 2020
Version 19.7.0.0.0
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
FLASHBACK automatically enabled to preserve database integrity.
Starting "SYS"."GOODUS_EXPDP":  "/"***** AS SYSDBA" directory=GOODUS
job_name=GOODUS_EXPDP DUMPFILE=SCHEMA_20201214.dmp
logfile=expdp_schema_20201214.log SCHEMAS=GOODUS
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
..(생략)
*****
Dump file set for SYS.GOODUS_EXPDP is:
  /export/goodus/SCHEMA_20201214.dmp
Job "SYS"."GOODUS_EXPDP" successfully completed at Thu Dec 17 20:17:32 2020 elapsed 0
00:00:55

```

■ IMPDP IN PDB

```

-PDB 로 접속하는 TNS 를 이용하여 IMPDP
$ impdp W"sys/oracle"@cont as sysdbaW" directory=goodus dumpfile= FULL_20201214.dmp logfile=
impdp_20201214.log

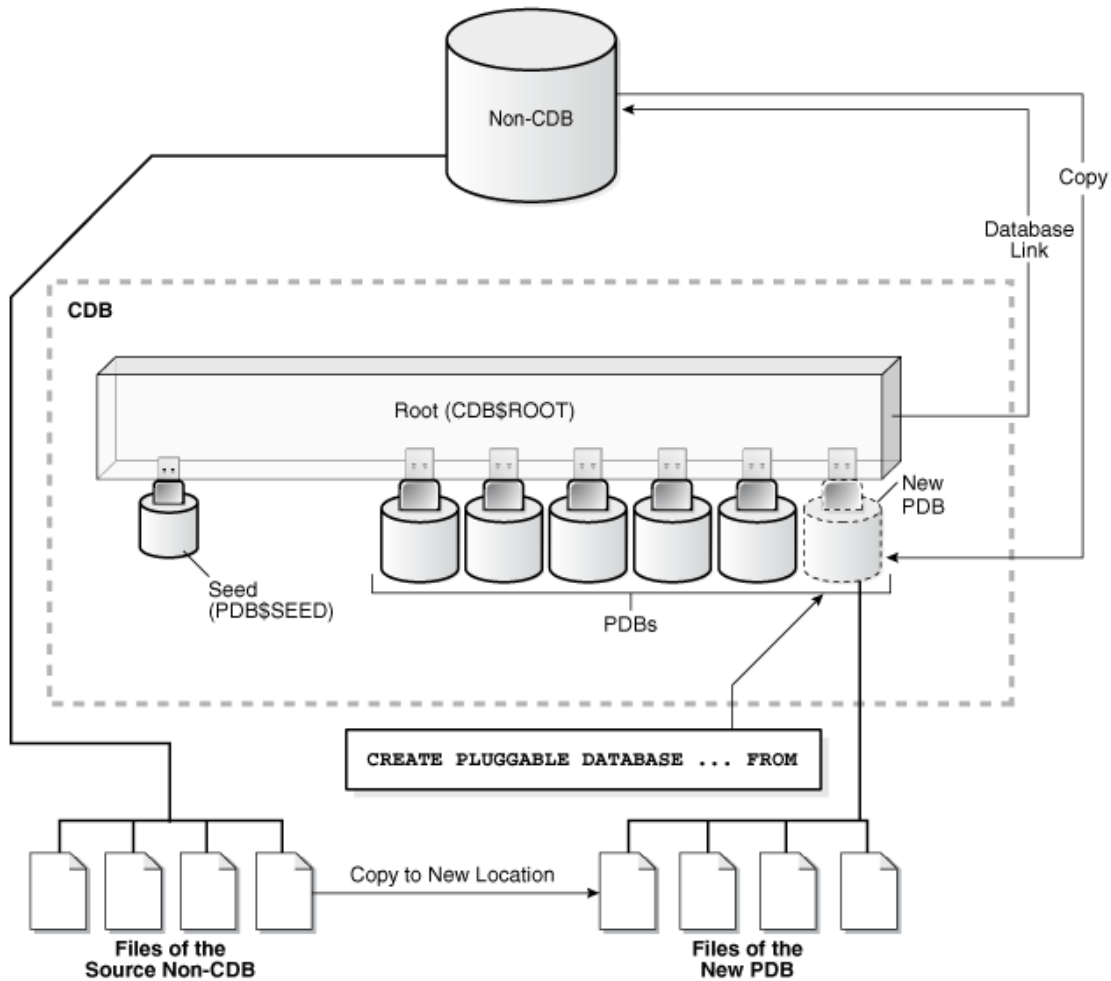
Import: Release 19.0.0.0.0 - Production on Thu Dec 17 20:11:05 2020
Version 19.7.0.0.0
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.
Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Master table "SYS"."SYS_IMPORT_FULL_01" successfully loaded/unloaded
Starting "SYS"."SYS_IMPORT_FULL_01":  "sys/*****@cont AS SYSDBA" directory=goodus
dumpfile=SCHEMA_20201214.dmp logfile=impdp_20201214.log
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/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE

```

Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
..(생략_
Job "SYS"."SYS_IMPORT_FULL_01" successfully completed at Thu Dec 17 20:11:12 2020 elapsed 0
00:01:06

2.2. DB_LINK

CDB 에 Non-CDB 를 바라보는 dblink 를 생성하여 PDB 를 원격으로 구성할 수 있습니다.



Description of "Figure 38-5 Creating a PDB by Cloning a Non-CDB"

2.2.1. Prerequisite

- Oracle 12.1 이상에서만 이용할 수 있으며 이하 버전일 경우 다른 방법을 이용하여 CDB 로 이관 또는 12.1 이상으로 Upgrade 후 진행 할 수 있습니다.
- DB Link 는 CDB 에서 Non-CDB 로 연결이 가능해야 합니다.
- DB Link 생성 시 Non-CDB 의 User 는 Create Pluggable Database 권한을 가지고 있어야 합니다.

2.2.2. Migration Non-CDB to PDB Using DB Link

■ DB LINK 시 사용할 USER 생성

■ CREATE USER in Non-CDB

```
SQL> creat user goodus identified by goodus ;
SQL> grant create session, create pluggable database to goodus;
Grant succeeded.
```

■ Non-CDB OPEN in READ ONLY MODE

```
SQL> startup mount;
ORACLE instance started.
Total System Global Area 2147481656 bytes
Fixed Size                8898616 bytes
Variable Size             536870912 bytes
Database Buffers 1593835520 bytes
Redo Buffers              7876608 bytes
Database mounted.

SQL> alter database open read only;
Database altered.

SQL> select name, open_mode from v$database;

NAME                OPEN_MODE
-----
NUL19C              READ WRITE
```

■ TNS 설정 및 DB LINK 생성

■ CDB TNSNAMES.ORA

```
NONCDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 172.40.40.44)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = nul19c)
    )
  )
```

■ CREATE DB LINK to Non-CDB in PDB

```
SQL> create database link noncdb
connect to goodus identified by goodus using 'NONCDB';
Database link created.
```

■ DB Link 를 이용하여 PDB 로 이관

■ CREATE PDB

```
SQL> create pluggable database pdb_nul4 from non$cdb@noncdb
file_name_convert=('/oradata01/NUL19C/', '/DATA/PDB_4/');
Pluggable database created.
```

■ CREATE PLUGGABLE DATABASE

```
SQL> create pluggable database pdb_nul4 --새로 생성 될 PDB_NAME
from non$cdb@noncdb --NON$CDB@DBLINK_NAME
file_name_convert=('/oradata01/NUL19C/', '/DATA/PDB_4/'); --변경 될 DATAFILE DEST
```

■ ISSUE

■ Source user 의 create pluggable database 권한이 없을 시 권한 부족 Error 발생
ERROR at line 1:

```
ORA-17628: Oracle error 1031 returned by remote Oracle server
ORA-01031: insufficient privileges
```

*

■ PDB 에서 CREATE PLUAGGABLE DATABSASE 진행 시 발생
ERROR at line 1:
ORA-65040: operation not allowed from within a pluggable database

■ CREATE PDB 확인

■ Check the Created Datafile

```
[nul:/DATA/PDB_4]$ls -al
total 8634368
drwxr-xr-x 2 oracle dba      183 Dec 16 14:02 .
drwxr-xr-x 4 oracle dba      32 Dec 16 13:16 ..
-rw-r----- 1 oracle dba 104865792 Dec 16 14:12 goodus01.dbf
-rw-r----- 1 oracle dba 524296192 Dec 16 14:12 logmnr001.dbf
-rw-r----- 1 oracle dba 2736791552 Dec 16 14:12 sysaux01.dbf
-rw-r----- 1 oracle dba 870326272 Dec 16 14:12 system01.dbf
```

```
-rw-r----- 1 oracle dba 36708352 Dec 16 14:04 temp01.dbf
-rw-r----- 1 oracle dba 1048584192 Dec 16 14:12 test001.dbf
-rw-r----- 1 oracle dba 524296192 Dec 16 14:12 test002.dbf
-rw-r----- 1 oracle dba 3025149952 Dec 16 14:12 undotbs01.dbf
-rw-r----- 1 oracle dba 5251072 Dec 16 14:12 users01.dbf
```

■ Check the Created PDB

```
SQL> select name, open_mode from v$pdb;
```

```
NAME      OPEN_MODE
-----
PDB$SEED  READ ONLY
PDB       READ WRITE
PDB_NUL4  MOUNTED
```

■ Change Non-CDB to PDB Using NONCDB_TO_PDB Scripts

Noncdb_to_pdb.sql script 는 Non-CDB 에서 PDB 로 변경해주는 스크립트로 PDB 가 Open 되기 전에 꼭 한번 수행되어야 하는 스크립트이며 sysdba 권한을 가진 유저로 PDB 에서 수행되어야합니다.

■ PDB 세션 변경

```
SQL> alter session set container=pdb_nul4;
Session altered.
```

■ Change Non-CDB to PDB Using NONCDB_TO_PDB Scripts

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

■ Noncdb_to_pdb 스크립트를 돌려 해결

```
SQL> select name, cause, type, status, message
       from PDB_PLUG_IN_VIOLATIONS;
```

NAME	CAUSE	TYPE	STATUS	MESSAGE
PDB_NUL4	Parameter	WARNING	RESOLVED	CDB parameter nls_language mismatch: Previous 'KOREAN' Current 'AMERICAN'
PDB_NUL4	Parameter	WARNING	RESOLVED	CDB parameter nls_territory mismatch: Previous 'KOREA' Current 'AMERICA'
PDB_NUL4	Parameter	WARNING	RESOLVED	CDB parameter sga_target mismatch: Previous 2G Current 1792M
PDB_NUL4	Parameter	WARNING	RESOLVED	CDB parameter pga_aggregate_target mismatch: Previous 681M Current 597M
PDB_NUL4	Non-CDB to PDB	ERROR	RESOLVED	PDB plugged in is a non-CDB, requires

noncdb_to_pdb.sql be run.

PDB_NUL4 OPTION WARNING PENDING Database option CONTEXT mismatch: PDB installed
version NULL. CDB installed version 19.0.0.0. - Component 는 추가적으로 설치 진행 해야합니다.

6 rows selected.

■ ISSUE

DB LINK User 에 Create Pluggable Database 권한이 없을 경우 발생

ERROR at line 1:

ORA-17628: Oracle error 1031 returned by remote Oracle server

ORA-01031: insufficient privileges

Known Issues: While running "noncdb_to_pdb.sql" (Doc ID 2288370.1)

Bug 18189497 - ORA-4068 during noncdb_to_pdb.sql (Doc ID 18189497.8)

2.3. TTS

TTS 를 이용하여 PDB 로 이관하는 방법은 기존과 동일하며 impdp 를 진행 할 때 이관을 진행할 pdb 의 tns 를 이용하여 진행한다는 점만 다릅니다.

2.3.1. Prerequisite

- Target DB 는 Source DB 의 버전과 같거나 더 높아야합니다.
- SYSTME 과 SYSAUX Tablespace 는 TTS 가 불가합니다.
- TTS 로 Migration 진행 할 Tablespace 는 Read Only Mode 이어야 합니다.
- 같은 이름을 가지고 있는 Tablespace 가 이미 존재하면 Target Database 에 Tablespace 를 이전할 수 없습니다.
- Tablespace 의 객체, Schema 가 Target Database 에 존재하지 않으면 impdp 전에 User 를 생성해주어야 합니다.
- Source DB 와 Target DB 는 호환성있는 Character Set 과 Nationl Character Set 을 사용해야 합니다.
- Endian 이 다를 경우 Rman 에서 Convert 작업을 진행해야합니다.

2.3.2. Migration Non-CDB to PDB Using TTS

■ 사전 작업

- Target DB 의 Instance_name, Version, Edition 확인
- 다른 Platform 일때 RMAN 을 통해 변환 작업 진행
- Characterset 확인
- Source User 생성 및 권한 부여

■ 위와 같은 사전 작업 시, Target 의 pdb 에서 진행해주어야 합니다

■ tns 를 이용한 pdb 접속

```
$ sqlplus sys/"PASSWORD"@TNS as sysdba
```

■ CDB TNSNAMES.ORA

```
CONT =  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = 172.40.40.46)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = pdb)  
  )  
)
```

■ cdb 로 접속하여 pdb 로 세션 변경

```

SQL> show pdbs
CON_ID      CON_NAME          OPEN MODE  RESTRICTED
-----
2          PDB$SEED         READ ONLY  NO
3          PDB              READ WRITE NO

SQL>
alter session set container=PDB;

SQL> show con_name
CON_NAME
-----
PDB

```

■ 이관을 진행할 Tablespace 확인

■ NON CDB

```

SQL> select tp.PLATFORM_NAME, tp.ENDIAN_FORMAT from v$database d, v$transportable_platform
tp where d.platform_name=tp.platform_name;

PLATFORM_NAME      ENDIAN_FORMAT
-----
Linux x86 64-bit   Little

SQL> select tablespace_name, file_name from dba_data_files;

TABLESPACE_NAME      FILE_NAME
-----
SYSTEM                /oradata01/NUL19C/system01.dbf
SYSAUX                /oradata01/NUL19C/sysaux01.dbf
UNDOTBS1              /oradata01/NUL19C/undotbs01.dbf
USERS                 /oradata01/NUL19C/users01.dbf
GOODUS                /oradata01/NUL19C/goodus01.dbf

5 rows selected

SQL> select owner, table_name, tablespacE_name
      from dba_tables where tablespace_name='GOODUS'

OWNER      TABLE_NAME      TABLESPACE_NAME
-----

```

- TTS Violations 확인

TTS 진행 시, 호환되지 않거나 문제가 있는지 DBMS_TTS.TRANSPOT_SET_CHECK 를 이용하여 확인 할 수 있으며 TRANSPORT_SET_VIOLATIONS 로 자세한 내용을 확인 할 수 있습니다.

```
SQL> execute dbms_tts.transport_set_check('GOODUS');
```

```
PL/SQL procedure successfully completed.
```

```
SQL> SELECT * FROM SYS.TRANSPORT_SET_VIOLATIONS;
```

```
no rows selected
```

- TTS 를 이용하여 Non-CDB to PDB 이관

- Non-CDB Tablespace Read Only Mode

```
SQL> alter tablespace goodus read only;
```

```
Tablespace altered.
```

```
SQL> select tablespace_name, status from dba_tablespaces;
```

TABLESPACE_NAME	STATUS
SYSTEM	ONLINE
SYSAUX	ONLINE
TEMP	ONLINE
USERS	ONLINE
GOODUS	READ ONLY

- TTS 를 이용하여 Meta data expdp

```
$ expdp ₩"/ as sysdba₩" directory=goodus dumpfile=goodus_ts.dmp logfile=goodus_ts.log
transport_tablespace=y tablespaces=goodus
```

```
Export: Release 19.0.0.0.0 - Production on Thu Dec 17 11:23:51 2020
```

```
Version 19.7.0.0.0
```

```
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
```

```
Legacy Mode Active due to the following parameters:
```

```
Legacy Mode Parameter: "transport_tablespace=TRUE" Location: Command Line, Replaced with:
"transport_tablespaces=goodus"
```

```
Legacy Mode has set reuse_dumpfiles=true parameter.
```

```

Starting "SYS"."SYS_EXPORT_TRANSPORTABLE_01": "/***** AS SYSDBA" directory=goodus
dumpfile=goodus_ts.dmp logfile=goodus_ts.log tablespaces=goodus reuse_dumpfiles=true
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/MARKER
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Master table "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/unloaded
*****
Dump file set for SYS.SYS_EXPORT_TRANSPORTABLE_01 is:
  /export/goodus/goodus_ts.dmp
*****
Datafiles required for transportable tablespace GOODUS:
  /oradata01/NUL19C/goodus01.dbf
Job "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Thu Dec 17 11:24:29 2020
elapsed 0 00:00:34

```

■ DumpFile Copy or SCP

```

$ scp * 172.40.40.46:/export/goodus
oracle@172.40.40.46's password:
goodus_ts.dmp                               100% 172KB 74.4MB/s 00:00

[nul:/home/oracle]$ls -al /export/goodus
total 180
drwxr-xr-x 2 oracle dba      73 Dec 17 12:20 .
drwxr-xr-x 3 oracle dba     20 Dec 17 11:34 ..
-rwxr-xr-x 1 oracle dba 176128 Dec 17 11:48 goodus_ts.dmp

```

■ Tablespace Datafile Copy or SCP

TTS 대상인 Tablespace 의 DataFile 을 Target 경로에 맞게 copy 합니다.

```

$ scp /oradata01/NUL19C/goodus01.dbf 172.40.40.46:/oracle/mes/oradata/NUL/pdb/goodus01.dbf
oracle@172.40.40.46's password:
goodus01.dbf

$ ls -al
total 543584
drwxr-x--- 2 oracle dba      103 Dec 17 11:45 .
drwxr-x--- 4 oracle dba     226 Dec 15 19:50 ..
-rw-r----- 1 oracle dba 104865792 Dec 17 11:45 goodus01.dbf

```

```
-rw-r----- 1 oracle dba 173023232 Dec 17 11:44 sysaux01.dbf
-rw-r----- 1 oracle dba 272637952 Dec 17 11:44 system01.dbf
-rw-r----- 1 oracle dba 35659776 Dec 16 22:00 temp01.dbf
-rw-r----- 1 oracle dba 5251072 Dec 16 22:06 users01.dbf
```

만약, Endian 이 다르면 Source rman 에서 convert datafile command 를 이용하여 Target DB 에 맞는 파일로 변환 및 재생성하여 넘겨준 후 변환된 File 을 Target DB 에서 Convert 작업을 수행합니다.

Source

RMAN>

```
convert datafile 'GOODUS' to platform 'Linux x86 64-bit' format '/export/goodus/goodus01.dbf';
```

Target

```
RMAN> convert datafile '/export/goodus/goods01.dbf' format
'/oracle/mes/oradata/NUL/pdb/goodus01.dbf';
```

ASM 으로 이관 시, Target Rman 에서 +Diskgroup 으로 변환 작업을 진행 할 수 있습니다.

```
RMAN> convert datafile '/export/goodus/goodus01.dbf' format '+DATA/NUL/GOODUS.dbf';
```

■ Metadata impdp Using TTS

```
$ impdp ♯"sys/oracle@cont as sysdba♯" directory=goodus1 dumpfile=goodus_ts.dmp
logfile=goodus_ts_imp.log transport_tablespace=y
datafiles='/oracle/mes/oradata/NUL/pdb/goodus01.dbf'
```

Import: Release 19.0.0.0.0 - Production on Thu Dec 17 12:21:30 2020

Version 19.7.0.0.0

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

Legacy Mode Active due to the following parameters:

Legacy Mode Parameter: "datafiles=/oracle/mes/oradata/NUL/pdb/goodus01.dbf" Location:

Command Line, Replaced with: "transport_datafiles=/oracle/mes/oradata/NUL/pdb/goodus01.dbf"

Legacy Mode Parameter: "transport_tablespace=TRUE" Location: Command Line, ignored.

Master table "SYS"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/unloaded

Starting "SYS"."SYS_IMPORT_TRANSPORTABLE_01": "sys/*****@cont AS SYSDBA"

directory=goodus1 dumpfile=goodus_ts.dmp logfile=goodus_ts_imp.log

transport_datafiles=/oracle/mes/oradata/NUL/pdb/goodus01.dbf

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 "SYS"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Thu Dec 17 12:21:57 2020

elapsed 0 00:00:25

- Non-CDB 의 Tablespace 가 PDB 로 이관 된 것을 확인
- PDB DATA Check

```
$ sqlplus sys/oracle@cont as sysdba
```

```
SQL> show con_name
```

```
CON_NAME
```

```
-----  
PDB
```

```
SQL> select tablespace_name, file_name from dba_data_files
```

```
TABLESPACE_NAME
```

```
FILE_NAME
```

```
-----  
SYSTEM
```

```
/oracle/mes/oradata/NUL/pdb/system01.dbf
```

```
SYSAUX
```

```
/oracle/mes/oradata/NUL/pdb/sysaux01.dbf
```

```
USERS
```

```
/oracle/mes/oradata/NUL/pdb/users01.dbf
```

```
GOODUS
```

```
/oracle/mes/oradata/NUL/pdb/goodus01.dbf
```

```
SQL> select tablespace_name, status from dba_tablespaces;
```

```
TABLESPACE_NAME
```

```
STATUS
```

```
-----  
SYSTEM
```

```
ONLINE
```

```
SYSAUX
```

```
ONLINE
```

```
TEMP
```

```
ONLINE
```

```
USERS
```

```
ONLINE
```

```
GOODUS
```

```
READ ONLY
```

```
SQL> select * from goodus.tts;
```

```
NO ID
```

```
-----  
1 a
```

```
2 b
```

```
3 c
```

```
SQL> alter tablespace goodus read write;
```

```
Tablespace altered.
```

```
SQL> select tablespace_name, status from dba_tablespaces;
```

TABLESPACE_NAME	STATUS
-----------------	--------

SYSTEM	ONLINE
SYSAUX	ONLINE
TEMP	ONLINE
USERS	ONLINE
GOODUS	ONLINE

■ ISSUE

Tablespace 가 Read Only Mode 가 아닐 경우 발생

ORA-39123: Data pump transportable tablespace job aborted.

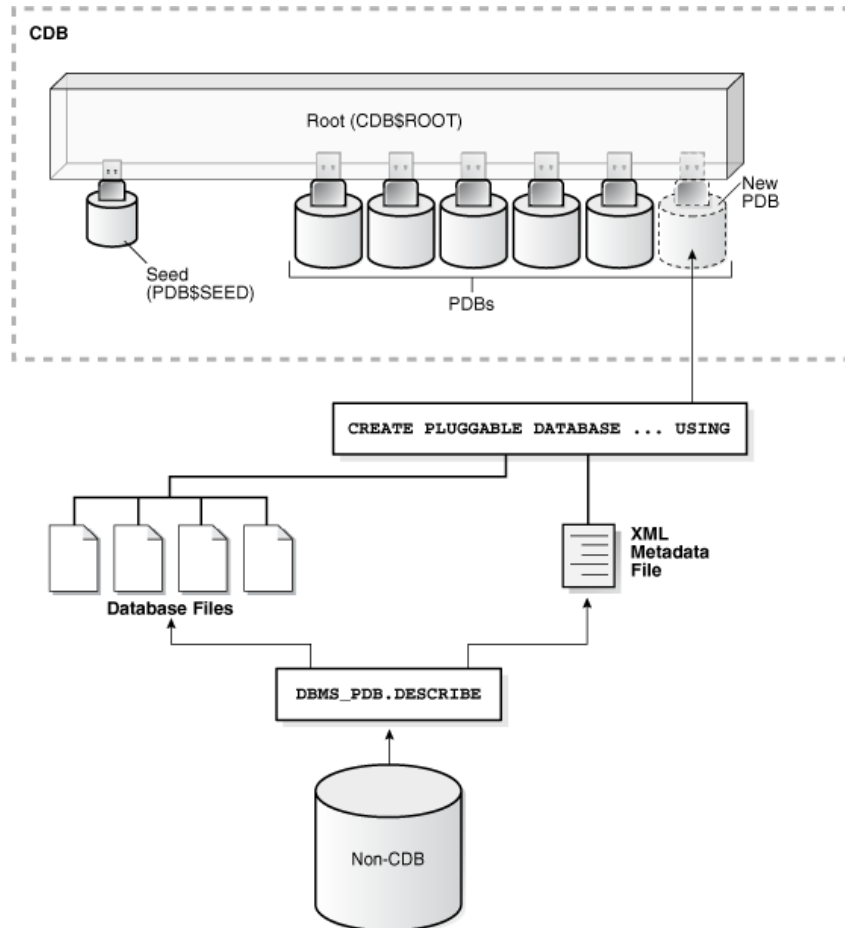
ORA-39185: The transportable tablespace failure list is

ORA-29335: tablespace 'xxx' is not read only

2.4. Hot Backup

Hot Backup 을 이용한 Clone DB 구성하여 pdb 로 변경 할 수 있습니다.

이때, DBMS_PDB Package 를 이용하여 Non-CDB 를 PDB 로 Convert 할 수 있습니다.



2.4.1. Prerequisite

- Non-CDB 의 Metadata 구조인 xml 파일은 CDB 가 ACCESS 할 수 있는 경로에 있어야 합니다.
- Non-CDB 와 PDB 의 Endian 은 동일해야합니다.
- CDB 에 설치 되어있는 Database Option 은 Non-CDB 에 설치된 Option 과 동일하거나 포함해야 합니다.

2.4.2. Migration Non-CDB to PDB Using HotBackup(Clone DB)

■ Hot Backup 을 이용한 Clone DB 구성

■ Hot backup 진행

```
SQL> alter database begin backup;  
Database altered.
```

■ 생성된 PDB 확인

```
$ ls /oradata01/NUL19C/*  
control01.ctl goodus01.dbf redo01.log redo03.log system01.dbf test001.dbf  
undotbs01.dbf  
control02.ctl logmnr001.dbf redo02.log sysaux01.dbf temp01.dbf test002.dbf users01.dbf  
  
$ cp * /backup/pdb/  
  
$ ls -al /backup/pdb  
total 9280660  
drwxr-xr-x 2 oracle dba      4096 Dec 15 18:09 .  
drwxr-xr-x 4 oracle dba      4096 Dec 15 13:17 ..  
-rw-r----- 1 oracle dba 10993664 Dec 15 18:08 control01.ctl  
-rw-r----- 1 oracle dba 10993664 Dec 15 18:08 control02.ctl  
-rw-r----- 1 oracle dba 104865792 Dec 15 18:08 goodus01.dbf  
-rw-r----- 1 oracle dba 524296192 Dec 15 18:08 logmnr001.dbf  
-rw-r----- 1 oracle dba 209715712 Dec 15 18:08 redo01.log  
-rw-r----- 1 oracle dba 209715712 Dec 15 18:08 redo02.log  
-rw-r----- 1 oracle dba 209715712 Dec 15 18:08 redo03.log  
-rw-r----- 1 oracle dba 2736791552 Dec 15 18:09 sysaux01.dbf  
-rw-r----- 1 oracle dba 870326272 Dec 15 18:09 system01.dbf  
-rw-r----- 1 oracle dba 36708352 Dec 15 18:09 temp01.dbf  
-rw-r----- 1 oracle dba 1048584192 Dec 15 18:09 test001.dbf  
-rw-r----- 1 oracle dba 524296192 Dec 15 18:09 test002.dbf  
-rw-r----- 1 oracle dba 3025149952 Dec 15 18:09 undotbs01.dbf  
-rw-r----- 1 oracle dba 5251072 Dec 15 18:09 users01.db
```

■ End backup

```
SQL> alter database end backup;  
Database altered.
```

■ HOT Backup 확인

```
SQL> select a.file_name, a.tablespace_name, b.*  
from dba_Data_files a, v$backup b
```

```
where a.file_id = b.file#;
```

FILE_NAME	TABLESPACE_NAME	FILE#	STATUS	CHANGE#	TIME	CON_ID
/oradata01/NUL19C/system01.dbf	SYSTEM	1	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/sysaux01.dbf	SYSAUX	2	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/undotbs01.dbf	UNDOTBS1	3	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/users01.dbf	USERS	4	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/test001.dbf	RECOVER_TEST	5	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/test002.dbf	RECOVER_TEST	6	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/logmnr001.dbf	TBS_LOGMNR	7	NOT ACTIVE	18006350	15-DEC-20	0
/oradata01/NUL19C/goodus01.dbf	GOODUS	8	NOT ACTIVE	18006350	15-DEC-20	0

8 rows selected.

■ CREATE PFILE

```
SQL> create pfile='/backup/pdb/p.ora' from spfile;
```

File created.

■ CDB 가 설치되어있는 서버로 SCP 하여 RESTORE 진행

```
$ ls /backup/pdb/*
```

```
control01.ctl goodus01.dbf p.ora redo02.log sysaux01.dbf temp01.dbf test002.dbf  
users01.dbf control02.ctl logmnr001.dbf redo01.log redo03.log system01.dbf test001.dbf  
undotbs01.dbf
```

```
$ scp * 172.40.40.46:/DATA/PDB_3/
```

```
oracle@172.40.40.46's password:
```

```
control01.ctl      100%  10MB  95.7MB/s   00:00  
control02.ctl      100%  10MB  226.4MB/s  00:00  
goodus01.dbf       100% 100MB 155.8MB/s  00:00  
logmnr001.dbf      100% 500MB 211.5MB/s  00:02  
p.ora              100%  10KB  768.8KB/s  00:00  
redo01.log         100% 200MB 149.1MB/s  00:01  
redo02.log         100% 200MB 222.3MB/s  00:00  
redo03.log         100% 200MB 290.9MB/s  00:00  
sysaux01.dbf       100% 2610MB 210.6MB/s  00:12  
system01.dbf       100%  830MB 195.2MB/s  00:04  
temp01.dbf         100%   35MB 183.2MB/s  00:00  
test001.dbf        100% 1000MB 202.6MB/s  00:04  
test002.dbf        100%  500MB 185.0MB/s  00:02  
undotbs01.dbf      100% 2885MB 227.1MB/s  00:12  
users01.dbf        100% 5128KB  50.3MB/s  00:00
```

■ Clone DB Open

```
SQL> startup pfile=/DATA/PDB_3/p.ora mount;
ORACLE instance started.

Total System Global Area 2147481656 bytes
Fixed Size                  8898616 bytes
Variable Size                536870912 bytes
Database Buffers 1593835520 bytes
Redo Buffers                 7876608 bytes
Database mounted.

SQL> alter database open;
Database altered.
```

■ Non-CDB to PDB Using DBMS_PDB DESCRIBE

DBMS_PDB.DESCRIBE 프로시저를 이용하여 Non-cdb 에서 사용중인 DB 구성 메타데이터를 xml 파일을 생성하여 pdb 로 변경 할 수 있습니다. 새로운 Datafile 의 DESTINATION 값을 데이터베이스 파일을 복사할 만한 충분한 공간이 있는 경로로 설정해야 합니다.

■ NON-PDB DB 를 내린 후 exclusive mode 로 올려줍니다.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup pfile=/DATA/PDB_3/p.ora mount exclusive;
ORACLE instance started.

Total System Global Area 2147481656 bytes
Fixed Size                  8898616 bytes
Variable Size                486539264 bytes
Database Buffers 1644167168 bytes
Redo Buffers                 7876608 bytes
Database mounted.

SQL> alter database open read only;
Database altered.
```

■ DBMS_PDB.DESCRIBE 를 이용하여 해당 DB 의 xml 파일을 추출 합니다.

Pdb_descr_file : Source DB 의 xml 파일이 생길 경로를 지정합니다.

```
SQL> exec dbms_pdb.describe(pdb_descr_file => '/home/oracle/nul.xml');  
PL/SQL procedure successfully completed.
```

■ Xml 파일을 이용한 Non-CDB Architecture 확인

```
$ cat nul.xml  
<?xml version="1.0" encoding="UTF-8"?>  
<PDB>  
  <xmlversion>1</xmlversion>  
  <pdbname>nul19c</pdbname> -- instance_name  
  <cid>0</cid>  
  <byteorder>1</byteorder>  
  <vsn>318767104</vsn>  
  <vsns>  
    <vsnum>19.0.0.0.0</vsnum>  
    <cdbcompt>19.0.0.0.0</cdbcompt>  
    <pdbcompt>19.0.0.0.0</pdbcompt>  
    <vsnlibnum>0.0.0.0.24</vsnlibnum>  
    <vsnsql>24</vsnsql>  
    <vsnbsv>8.0.0.0.0</vsnbsv>  
  </vsns>  
  <dbid>1880484371</dbid> -- dbid  
  <ncdb2pdb>1</ncdb2pdb>  
  <cdbid>1880484371</cdbid>  
  <guid>AA1A0CDBF0193E27E0532C2828AC058D</guid>  
  <uscnbas>17960084</uscnbas>  
  <uscnwrp>0</uscnwrp>  
  <undoscn>8</undoscn>  
  <rdba>4194824</rdba>  
  <tablespace> -- tablespace info  
    <name>SYSTEM</name>  
    <type>0</type>  
    <tsn>0</tsn>  
    <status>1</status>  
    <issft>0</issft>  
    <isnft>0</isnft>  
    <encts>0</encts>  
    <flags>0</flags>  
    <bmunitsize>8</bmunitsize>  
    <file>  
      <path>/oradata01/NUL19C/system01.dbf</path> -- tablespace dest
```

```

<afn>1</afn>
<rfn>1</rfn>
<createscnbas>8</createscnbas>
<createscnwrp>0</createscnwrp>
<status>1</status>
<fileblocks>106240</fileblocks>
<blocksize>8192</blocksize>
<vsn>318767104</vsn>
<fdbid>1880484371</fdbid>
<fcpsb>17960083</fcpsb>
<fcpsw>0</fcpsw>
<frlsb>1</frlsb>
<frlsw>0</frlsw>
<frlt>1045446227</frlt>
<autoext>1</autoext>
<maxsize>4194302</maxsize>
<incsize>1280</incsize>
<plugscn>0</plugscn>
<plugafn>0</plugafn>
<plugdbid>0</plugdbid>
<dfflags>1</dfflags>
</file>
... Tablespace 생략
<recover>0</recover>
<optional>
  <ncdb2pdb>1</ncdb2pdb>
  <csid>873</csid>
  <ncsid>2000</ncsid>
  <options>
    <option>CATALOG=19.0.0.0.0</option> --component Info
    <option>CATJAVA=19.0.0.0.0</option>
    <option>CATPROC=19.0.0.0.0</option>
    <option>JAVAVM=19.0.0.0.0</option>
    <option>OWM=19.0.0.0.0</option>
    <option>XDB=19.0.0.0.0</option>
    <option>XML=19.0.0.0.0</option>
  </options>
  <dv>0</dv>
  <APEX>NULL</APEX>
  <parameters> --parameter Check
  <parameter>processes=320</parameter>

```

```

<parameter>nls_language='KOREAN'</parameter>
<parameter>nls_territory='KOREA'</parameter>
<parameter>sga_target=2147483648</parameter>
<parameter>db_block_size=8192</parameter>
<parameter>compatible='19.0.0'</parameter>
<parameter>open_cursors=300</parameter>
<parameter>pga_aggregate_target=714080256</parameter>
</parameters>
<sqlpatches> --patch Info
  <sqlpatch>19.7.0.0.0 Release_Update 2004040350 (RU): APPLY SUCCESS</sqlpatch>
  <sqlpatch>Interim patch 30805684/23401476 (OJVM RELEASE UPDATE: 19.7.0.0.200414
(30805684)): APPLY SUCCESS</sqlpatch>
</sqlpatches>
<tzvers> --timezone Check
  <tzver>primary version:32</tzver>
  <tzver>secondary version:0</tzver>
</tzvers>
<walletkey>0</walletkey>
<services>
  <service>SYS$BACKGROUND,</service>
  <service>SYS$USERS,</service>
  <service>nul19cXDB,nul19cXDB</service>
  <service>nul19c,nul19c</service>
</services>
<opatches/>
<hasclob>1</hasclob>
<awr/>
<hardvsnchk>0</hardvsnchk>
<localundo>1</localundo>
<apps/>
<dbedition>8</dbedition>
<dvopsctl>2</dvopsctl>
<clnupsrcpal>1</clnupsrcpal>
</optional>
</PDB>

```

- DBMS_PDB.CHECK_PLUG_COMPATIBILITY 를 이용한 호환성 체크

```

SQL> DECLARE
  compatible CONSTANT VARCHAR2(3) :=
  CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
  pdb_descr_file => '/home/oracle/nul.xml',

```

```

pdb_name => 'PDB_NUL')
WHEN TRUE THEN 'YES'
ELSE 'NO'
END;
BEGIN
DBMS_OUTPUT.PUT_LINE(compatible);
END;
/

```

PL/SQL procedure successfully completed

■ PDB_PLUG_IN_VIOLATIONS

PDB_PLUG_IN_VIOLATIONS 를 통해 호환되지 않는 이유를 확인할 수 있습니다.
DB 의 Version, Patch 가 동일해야하는 등 다양한 원인을 알려줍니다.

```
SQL> SELECT * FROM PDB_PLUG_IN_VIOLATIONS;
```

■ CDB 의 SID 로 변경 후 PLUGGABLE DATABASE 를 추가해 줍니다..

```
$ export ORACLE_SID=orcl
```

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB_NUL	READ WRITE	NO

```
SQL> create pluggable database pdb_nul2 using '/home/oracle/nul.xml'
copy file_name_convert = ('/oradata01/NUL19C/', '/oracle/mes/oradata/NUL/pdb_nul2/');
Pluggable database created.
```

```
$ ls /oracle/mes/oradata/NUL/pdb_nul2/*
```

```
goodus01.dbf logmnr001.dbf sysaux01.dbf system01.dbf temp01.dbf test001.dbf test002.dbf
undotbs01.dbf users01.dbf
```

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB_NUL	READ WRITE	NO
4	PDB_NUL2	MOUNTED	

```
SQL> alter session set container=PDB_NUL2 ;
```

Session altered.

Noncdb_to_pdb.sql script 는 Non-CDB 에서 PDB 로 변경해주는 스크립트로 PDB 가 Open 되기 전에 꼭 한번 수행되어야 하는 스크립트이며 sysdba 권한을 가진 유저로 PDB 에서 수행되어야합니다.

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

```
SQL> show con_name
```

```
CON_NAME
-----
PDB_NUL2
```

```
SQL> select instance_name, version, status from v$instance;
```

```
INSTANCE_NAME VERSION          STATUS
-----
nul            19.0.0.0.0          MOUNTED
```

```
SQL> alter database open;
```

Database altered.

```
SQL> select instance_name, version, status from v$instance;
```

```
INSTANCE_NAME VERSION          STATUS
-----
nul            19.0.0.0.0          OPEN
```

```
SQL> select count(*) from dba_objects where owner='GOODUS';
```

```
COUNT(*)
-----
35
```

■ ISSUE

Mismatch Xml metadata file and data file.

xml 을 생성 할 때 non-cdb database 에서 read only mode 가 아닐 때 xml 을 뽑아 datafile 과 metadata 가 맞지 않음

```
SQL> create pluggable database pdb_nul2 using '/home/oracle/nul.xml'
```

```
copy file_name_convert = ('/oradata01/NUL19C/', '/oracle/mes/oradata/NUL/pdb_nul2/');
```


ERROR at line 1:

ORA-65139: Mismatch between XML metadata file and data file
/oradata01/NUL19C/system01.dbf for value of fcpsb (17958602 in the plug XML
file, 17960083 in the data file)

ORA-65139: Mismatch Between XML Metadata File And Data File. Database Not Open Read Only
(Doc ID 1963139.1)

3. Reference

- <https://oracle-base.com/articles/12c/multitenant-manage-users-and-privileges-for-cdb-and-pdb-12cr1#create-common-users>
- <https://oracle-base.com/articles/12c/multitenant-migrate-non-cdb-to-pdb-12cr1#clone-remote-non-cdb>
- https://docs.oracle.com/database/121/SQLRF/statements_6010.htm#SQLRF55686
- <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/sqlrf/CREATE-PLUGGABLE-DATABASE.html#GUID-F2DBA8DD-EEA8-4BB7-A07F-78DC04DB1FFC>
- Best-practice Order for Multiple Changes Involving nonCDB to PDB Migration (Doc ID 2534697.1)
- ORA-65122: Pluggable Database GUID Conflicts With The GUID Of An Existing Container (Doc ID 2416798.1)
- https://docs.oracle.com/database/121/ADMIN/cdb_plug.htm#ADMIN13549
- <https://oracle-base.com/articles/12c/multitenant-hot-clone-remote-pdb-or-non-cdb-12cr2>
- Known Issues: While running "noncdb_to_pdb.sql" (Doc ID 2288370.1)
- Bug 18189497 - ORA-4068 during noncdb_to_pdb.sql (Doc ID 18189497.8)
- PDB Transportable Tablespace Export Failed With ORA-29335: Tablespace 'TBS1' is not READ ONLY (Doc ID 2572737.1)