



Remerciements

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de l'institut supérieur d'ingénierie et des affaires ISGA, et tous les responsables de la filière d'ingénierie.

Nous remercions également Mr. Snineh, notre professeur de programmation, pour l'aide et les conseils concernant les missions évoquées dans ce mini-projet.

TABLE OF CONTENTS

1. Duplication de CDB Oracle	4
 1.1 Fondamentaux de l'Architecture Multitenant Oracle	4
 1.2 Principes de Duplication de CDB	6
2. Oracle Data Guard	25
 2.1 Introduction	25
 2.2 Composants	26
 2.3 Architecture et Processus	27
 2.4 Service Redo Apply	28
 2.5 Processus d'Archivage (ARCH)	30
Conclusion	35

Partie 1: Duplication de CDB Oracle

A. Concepts Théoriques

1. Fondamentaux de l'Architecture Multitenant Oracle Qu'est-ce que Oracle Multitenant? L'architecture Oracle Multitenant permet à une base de données conteneur (CDB) unique d'héberger plusieurs bases de données pluggables (PDB), offrant ainsi une consolidation des bases de données et une gestion simplifiée. **Composants Clés:**

Base de Données Conteneur (CDB):

La base de données racine qui contient les métadonnées système Gère les ressources partagées (mémoire, processus, stockage) Contient le conteneur racine (CDB\$ROOT) et la PDB modèle (PDB\$SEED)

Base de Données Pluggable (PDB):

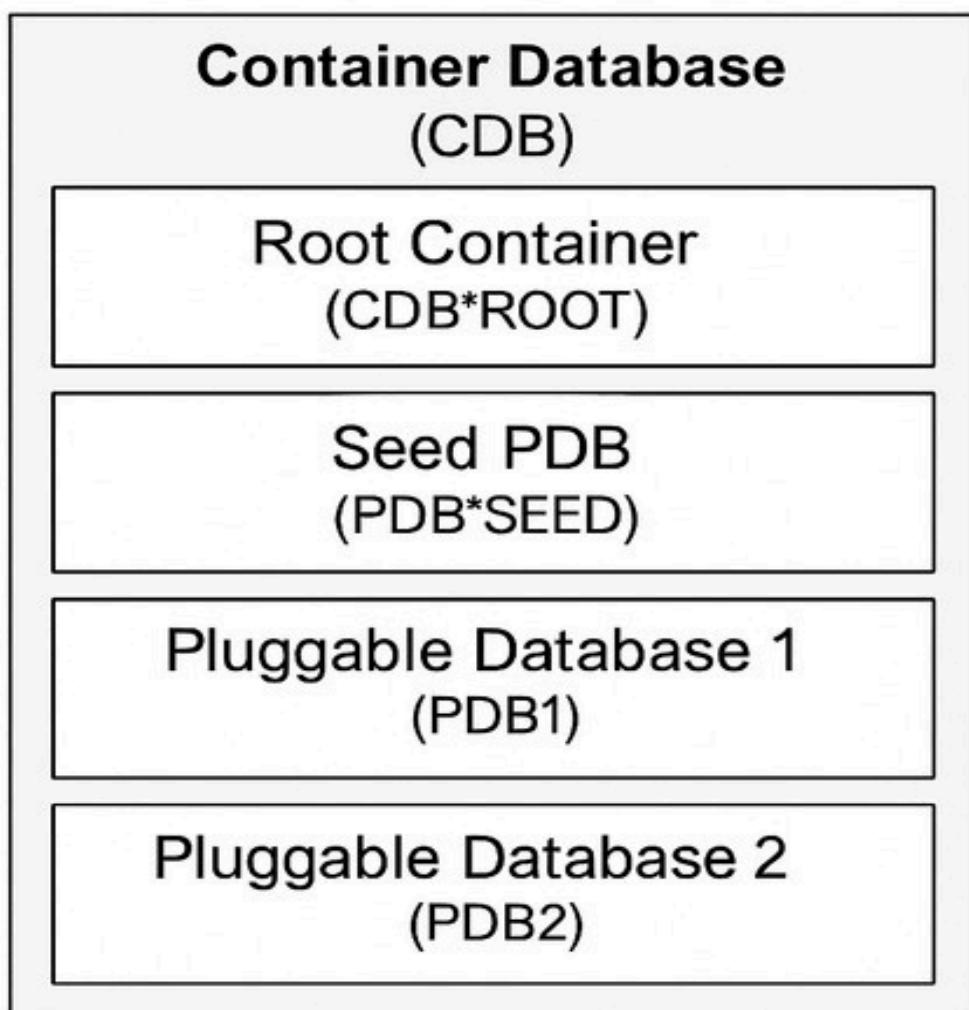
Base de données autonome au sein d'une CDB apparaît comme une base de données traditionnelle pour les applications Possède ses propres utilisateurs, schémas et objets Peut être facilement déplacée entre les CDB

Conteneur Racine (CDB\$ROOT):

Modèle pour créer de nouvelles PDB Base de données en lecture seule Contient les objets fournis par Oracle

Diagramme d'Architecture:

Diagramme d'Architecture:



Avantages de l'Architecture Multitenant:

Consolidation des Ressources: Plusieurs bases de données partagent la même instance

Gestion Simplifiée: Correctifs et mises à niveau au niveau de la CDB Approvisionnement

Rapide: Création rapide de PDB à partir du modèle

Sécurité Renforcée: Utilisateurs communs et sécurité centralisée

Réduction des Coûts: Moins d'instances de base de données à gérer Isolation: Chaque PDB

fonctionne indépendamment Evolutivité: Facile d'ajouter/supprimer des PDB

2. Principes de Duplication de CDB

Qu'est-ce que la Duplication de CDB? La duplication de CDB est le processus de création d'une copie exacte d'une Base de Données Conteneur à l'aide d'Oracle Recovery Manager (RMAN). Cela inclut toutes les PDB au sein de la CDB.

Types de Duplication:

1. Duplication Active de la Base de Données:

La base de données source reste en ligne et opérationnelle Transfert direct des

fichiers de données via le réseau Aucun jeu de sauvegarde requis

Processus de duplication en temps réel

Impact minimal sur la base de données source

2. Duplication Basée sur Sauvegarde:

Utilise les jeux de sauvegarde RMAN existants La base de données source peut être en

ligne ou hors ligne Nécessite que les jeux de sauvegarde soient accessibles

Plus grande flexibilité de timing

3. Duplication Basée sur Sauvegarde avec Connexion Active:

La base de données source reste en ligne et opérationnelle Transfert direct des fichiers de

données via le réseau Aucun jeu de sauvegarde requis

Processus de duplication en temps réel

Impact minimal sur la base de données source

2. Duplication Basée sur Sauvegarde:

Utilise les jeux de sauvegarde RMAN existants La base de données source peut être en ligne ou hors ligne Nécessite que les jeux de sauvegarde soient accessibles Plus grande flexibilité de timing

3. Duplication Basée sur Sauvegarde avec Connexion Active:

Combine les jeux de sauvegarde avec une connexion active Utilise les sauvegardes pour les données en vrac, la connexion active pour les changements récents Offre flexibilité et efficacité

Prérequis pour la Duplication de CDB:

Exigences Réseau:

Connectivité réseau entre les systèmes source et cible Oracle Net Services correctement configuré Entrées TNS pour les deux bases de données

Listener en cours d'exécution sur les deux systèmes

Exigences Système de Fichiers:

Espace disque suffisant sur le système cible Structure de répertoires appropriée

Permissions de fichiers appropriées

Logiciel Oracle installé sur la cible

Exigences Base de Données:

Fichier de mots de passe pour l'instance auxiliaire Fichier de paramètres d'initialisation

Logiciel Oracle Database (même version ou version compatible)

Exigences RMAN:

Exécutable RMAN disponible Privilèges de connexion appropriés Base de données de catalogue (optionnelle mais recommandée)

Utilisation de RMAN dans la Duplication:

RMAN automatise le processus de duplication en:

- Cr  ant le fichier de param  tres de l'instance auxiliaire Restaurant et r  cup  rant les fichiers de donn  es.
- Cr  ant les fichiers de contr  le Configurant les fichiers de redo log Configurant la nouvelle base de donn  es

B. Partie pratique

Step 1 :

1.1 Create Docker Network, # Create a network for Oracle containers to communicate

```
docker network create oracle-net
```

1.2 Login to Oracle Container Registry,

```
# You need a free Oracle account docker login container-registry.oracle.com
```

Username: Your Oracle account email,

Password: Your Oracle account password

1.3 Pull Oracle 21c Database Image

```
# This will take time (several GB download)
```

```
docker pull container-registry.oracle.com/database/enterprise:21.3.0.0
```

Step 2 : CREATE SOURCE DATABASE (CDB1)

2.1 Start Source Container

```
docker run -d --name oracle-cdb1 \
--network oracle-net \
-p 1521:1521 \ -p 5500:5500 \ -e ORACLE_SID=CDB1 \
-e ORACLE_PDB=PDB1 \
-e ORACLE_PWD=Oracle123 \
-e ORACLE_CHARACTERSET=AL32UTF8 \
-v cdb1-data:/opt/oracle/oradata \ container-
registry.oracle.com/database/enterprise:21.3.0.0
```

2.2 Wait for Database Creation,

```
# Monitor the logs - this takes 10-15 minutes
```

```
docker logs -f oracle-cdb1
```

```
[2025:06:12 01:37:32]: Releasing lock .CDB1.create_lck
[2025:06:12 01:37:32]: Lock released .CDB1.create_lck
[2025:06:12 01:37:32]: Acquiring lock .CDB1.exist_lck with heartbeat 30 secs
[2025:06:12 01:37:32]: Lock acquired
[2025:06:12 01:37:32]: Starting heartbeat
[2025:06:12 01:37:32]: Lock held .CDB1.exist_lck

DONE: Executing user defined scripts

The Oracle base remains unchanged with value /opt/oracle
#####
DATABASE IS READY TO USE! ← # docker logs -f oracle-cdb1
#####
The following output is now a tail of the alert.log:
PDB1(3):CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/opt/oracle/oradata/CDB1/PDB1/users01.dbf' SIZE 5M REUSE AUTOEXTEND ON NEXT 1280K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
PDB1(3):Completed: CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/opt/oracle/oradata/CDB1/PDB1/users01.dbf' SIZE 5M REUSE AUTOEXTEND ON NEXT 1280K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
PDB1(3):ALTER DATABASE DEFAULT TABLESPACE "USERS"
PDB1(3):Completed: ALTER DATABASE DEFAULT TABLESPACE "USERS"
2025-06-12T01:37:24.682135+00:00
ALTER SYSTEM SET control_files='/opt/oracle/oradata/CDB1/control01.ctl' SCOPE=SPFILE;
2025-06-12T01:37:24.812950+00:00
ALTER SYSTEM SET local_listener='' SCOPE=BOTH;
ALTER PLUGGABLE DATABASE PDB1 SAVE STATE
Completed: ALTER PLUGGABLE DATABASE PDB1 SAVE STATE
```

2.3 Connect to Source Container

```
mac@Younes-farhat-3 ~ % docker exec -it oracle-cdb1 bash
bash-4.2$ source /home/oracle/.bashrc
bash-4.2$ export ORACLE_SID=CDB1
bash-4.2$ sqlplus / as sysdba
SQL*Plus: Release 21.0.0.0. - Production on Thu Jun 12 01:46:51 2025
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL>
```

← Connect to Source Container

↑ Connect to Database

2.4 Set Oracle Environment

```
bash-4.2$ source /home/oracle/.bashrc
bash-4.2$ export ORACLE_SID=CDB1
```

2.5 Connect to Database

```
bash-4.2$ sqlplus / as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Thu Jun 12 01:46:51 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL>
```

STEP 3: CONFIGURE SOURCE DATABASE

3.1 Enable Archive Log Mode

```
[SQL]> SELECT open_mode FROM v$database;
OPEN_MODE
-----
READ WRITE      is used to determine the current operational status of your Oracle database

[SQL]> SHUTDOWN IMMEDIATE;
Database closed.
Database dismounted.
ORACLE instance shut down.

[SQL]> STARTUP MOUNT;
ORACLE instance started.

Total System Global Area  935327096 bytes
Fixed Size                  9692536 bytes
Variable Size                247463936 bytes
Database Buffers            671088640 bytes
Redo Buffers                 7081984 bytes
Database mounted.
```

is used to determine the current operational status of your Oracle database

SQL command used in Oracle databases to stop the database instance gracefully but quickly

SQL command used in Oracle databases to bring the database instance to the mounted state

```
[SQL]> ALTER DATABASE ARCHIVELOG;
Database altered.
```

to change the archiving mode of the database to ARCHIVELOG mode.


```
[SQL]> ALTER DATABASE OPEN;
Database altered.
```

that makes the database fully accessible for normal operations, allowing users to connect and interact with the data.


```
[SQL]> ALTER PLUGGABLE DATABASE ALL OPEN;
Pluggable database altered.
```

powerful command for quickly bringing all your PDBs online in an Oracle Multitenant environment.


```
[SQL]> ALTER DATABASE FORCE LOGGING;
Database altered.
```

used to ensure that all database operations, even those explicitly designed to be "nologging," generate redo records and are written to the redo logs.

3.2 Verify Database Status

```
[SQL]> SELECT log_mode FROM v$database;
```

LOG_MODE
ARCHIVELOG

This v\$database column indicates the current redo log archiving mode of your Oracle database.


```
[SQL]> SELECT name, open_mode, cdb FROM v$database;
```

NAME	OPEN_MODE	CDB
CDB1	READ WRITE	YES

This command queries the V\$database dynamic performance view to retrieve the database's name


```
[SQL]> SELECT name, open_mode FROM v$pdb$;
```

NAME
PDB\$SEED
READ ONLY
PDB1
READ WRITE

This command queries the V\$PDBS dynamic performance view to list the names and current operational status (e.g., READ WRITE, READ ONLY, MOUNTED) of all Pluggable Databases (PDBs) within the current Container Database (CDB).

3.3 Create Test Data

```
[SQL]> ALTER SESSION SET CONTAINER=PDB1;
Session altered. This command switches your current SQL session's context from the Container
[SQL]> CREATE USER test_user IDENTIFIED BY test_pwd;
User created. This SQL command creates a new database user named test_user and
[SQL]> GRANT CREATE SESSION, CREATE TABLE, UNLIMITED TABLESPACE TO test_user;
Grant succeeded.

[SQL]> CONNECT test_user/test_pwd@localhost:1521/PDB1
Connected.

[SQL]> CREATE TABLE test_duplication (
      id NUMBER PRIMARY KEY,
      name VARCHAR2(100),
      created_date DATE DEFAULT SYSDATE
);
2    3    4    5
Table created.
```

This command attempts to establish a connection to the Oracle database as test_user, using test_pwd as the password, connecting to the Pluggable Database (PDB) named PDB1 on the local machine (localhost) via port 1521.

```
[SQL]> INSERT INTO test_duplication VALUES (1, 'Test Record 1', SYSDATE);
1 row created.

[SQL]> INSERT INTO test_duplication VALUES (2, 'Test Record 2', SYSDATE);
1 row created.

[SQL]> INSERT INTO test_duplication VALUES (3, 'Test Record 3', SYSDATE);
1 row created.

[SQL]> COMMIT;
Commit complete.
```

```
|SQL> COMMIT;  
Commit complete.  
  
|SQL> SELECT * FROM test_duplication;  
  
          ID  
-----  
        NAME  
-----  
CREATED_D  
-----  
          1  
Test Record 1  
12-JUN-25  
  
          2  
Test Record 2  
12-JUN-25  
  
          ID  
-----  
        NAME  
-----  
CREATED_D  
-----  
          3  
Test Record 3  
12-JUN-25
```

3.4 Exit SQL*Plus

```
EXIT;
```

4.1 Configure TNS Names

```
# Still inside oracle-cdb1 container  
vi $ORACLE_HOME/network/admin/tnsnames.ora  
Add this content (replace existing content):
```

```
bash-4.2$ cat > $ORACLE_HOME/network/admin/tnsnames.ora << 'EOF'
CDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oracle-cdb1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDB1)
    )
  )
CDB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oracle-cdb2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDB2)
    )
  )
PDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oracle-cdb1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
    )
  EOF ) (SERVICE_NAME = PDB1)

This command uses the cat utility to create or
overwrite the tnsnames.ora file, adding network
service definitions for three Oracle database
connections: CDB1, CDB2, and PDB1.
```

4.2 Configure Listener

vi \$ORACLE_HOME/network/admin/listener.ora

Add this content:

```
bash-4.2$ cat > $ORACLE_HOME/network/admin/listener.ora << 'EOF'  
> LISTENER =  
>   (DESCRIPTION_LIST =  
>     (DESCRIPTION =  
>       (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))  
>     )  
>   )  
>  
> SID_LIST_LISTENER =  
>   (SID_LIST =  
>     (SID_DESC =  
>       (GLOBAL_DBNAME = CDB1)  
>       (ORACLE_HOME = /opt/oracle/product/21c/dbhome_1)  
>       (SID_NAME = CDB1)  
>     )  
>   )  
[> EOF  
bash-4.2$
```

This command uses the cat utility to create or overwrite the listener.ora file, configuring the Oracle Net Listener to accept incoming connections for the CDB1 database instance on port 1521, making it accessible from any IP address.

4.3 Restart Listener

```
bash-4.2$ lsnrctl stop      This command stops the Oracle Net Listener process, which is  
                             responsible for accepting incoming connection requests from  
                             clients to the Oracle database.  
LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 02:17:07  
  
Copyright (c) 1991, 2021, Oracle. All rights reserved.  
  
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))  
The command completed successfully
```

```
bash-4.2$ lsnrctl start This command starts the Oracle Net Listener process,  
LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 02:17:12  
Copyright (c) 1991, 2021, Oracle. All rights reserved.  
Starting /opt/oracle/product/21c/dbhome_1/bin/tnslsnr: please wait...  
  
TNSLSNR for Linux: Version 21.0.0.0.0 - Production  
System parameter file is /opt/oracle/homes/OraDB21Home1/network/admin/listener.ora  
Log messages written to /opt/oracle/diag/tnslsnr/dbcf8f1456e9/listener/alert/log.xml  
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))  
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))  
  
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))  
STATUS of the LISTENER  
-----  
Alias LISTENER  
Version TNSLSNR for Linux: Version 21.0.0.0.0 - Production  
Start Date 12-JUN-2025 02:17:12  
Uptime 0 days 0 hr. 0 min. 0 sec  
Trace Level off  
Security ON: Local OS Authentication  
SNMP OFF  
Listener Parameter File /opt/oracle/homes/OraDB21Home1/network/admin/listener.ora  
Listener Log File /opt/oracle/diag/tnslsnr/dbcf8f1456e9/listener/alert/log.xml  
Listening Endpoints Summary...  
 (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))  
 (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))  
The listener supports no services  
The command completed successfully
```

```
bash-4.2$ lsnrctl status This command displays the current status of the Oracle Net Listener  
LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 02:17:16  
Copyright (c) 1991, 2021, Oracle. All rights reserved.  
  
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))  
STATUS of the LISTENER  
-----  
Alias LISTENER  
Version TNSLSNR for Linux: Version 21.0.0.0.0 - Production  
Start Date 12-JUN-2025 02:17:12  
Uptime 0 days 0 hr. 0 min. 4 sec  
Trace Level off  
Security ON: Local OS Authentication  
SNMP OFF  
Listener Parameter File /opt/oracle/homes/OraDB21Home1/network/admin/listener.ora  
Listener Log File /opt/oracle/diag/tnslsnr/dbcf8f1456e9/listener/alert/log.xml  
Listening Endpoints Summary...  
 (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))  
 (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))  
The listener supports no services  
The command completed successfully
```

4.4 Exit Container

EXIT ;

5.1 Start Target Container

```
SQL> SQL> Disconnected from Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
The Oracle base remains unchanged with value /opt/oracle

executing user defined scripts
/opt/oracle/runUserScripts.sh: running /opt/oracle/scripts/extensions/setup/swapLocks.sh
[2025:06:12 02:35:23]: Releasing lock .CDB2.create_lck
[2025:06:12 02:35:23]: Lock released .CDB2.create_lck
[2025:06:12 02:35:23]: Acquiring lock .CDB2.exist_lck with heartbeat 30 secs
[2025:06:12 02:35:23]: Lock acquired
[2025:06:12 02:35:23]: Starting heartbeat
[2025:06:12 02:35:23]: Lock held .CDB2.exist_lck

DONE: Executing user defined scripts

The Oracle base remains unchanged with value /opt/oracle
#####
DATABASE IS READY TO USE!
#####

The following output is now a tail of the alert.log:
ORCLPDB1(3):CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/opt/oracle/oradata/CDB2/ORCLPDB1/users01.dbf' SIZE 5M REUSE AUTOEXTEND ON N
PACE MANAGEMENT AUTO
ORCLPDB1(3):Completed: CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/opt/oracle/oradata/CDB2/ORCLPDB1/users01.dbf' SIZE 5M REUSE AUTO
SEGMENT SPACE MANAGEMENT AUTO
ORCLPDB1(3):ALTER DATABASE DEFAULT TABLESPACE "USERS"
ORCLPDB1(3):Completed: ALTER DATABASE DEFAULT TABLESPACE "USERS"
025-06-12T02:35:20.380577+00:00
ALTER SYSTEM SET control_files='/opt/oracle/oradata/CDB2/control01.ctl' SCOPE=SPFILE;
025-06-12T02:35:20.442979+00:00
ALTER SYSTEM SET local_listener='' SCOPE=BOTH;
ALTER PLUGGABLE DATABASE ORCLPDB1 SAVE STATE
Completed: ALTER PLUGGABLE DATABASE ORCLPDB1 SAVE STATE
```

The "DATABASE IS READY TO USE!" message is the most important part; it means the core Oracle database instance and its components are up and running and ready for connections.

5.2 Wait for Database Creation

Monitor logs -

wait for database ready message

docker logs -f oracle-cdb2

```
SQL> SQL> Disconnected from Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
The Oracle base remains unchanged with value /opt/oracle

executing user defined scripts
/opt/oracle/runUserScripts.sh: running /opt/oracle/scripts/extensions/setup/swapLocks.sh
[2025:06:12 02:35:23]: Releasing lock .CDB2.create_lck
[2025:06:12 02:35:23]: Lock released .CDB2.create_lck
[2025:06:12 02:35:23]: Acquiring lock .CDB2.exist_lck with heartbeat 30 secs
[2025:06:12 02:35:23]: Lock acquired
[2025:06:12 02:35:23]: Starting heartbeat
[2025:06:12 02:35:23]: Lock held .CDB2.exist_lck

DONE: Executing user defined scripts

The Oracle base remains unchanged with value /opt/oracle
#####
DATABASE IS READY TO USE!
#####

The following output is now a tail of the alert.log:
ORCLPDB1(3):CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/opt/oracle/oradata/CDB2/ORCLPDB1/users01.dbf' SIZE 5M REUSE AUTOEXTEND ON NEXT 1280K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT S
PACE MANAGEMENT AUTO
ORCLPDB1(3):Completed: CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/opt/oracle/oradata/CDB2/ORCLPDB1/users01.dbf' SIZE 5M REUSE AUTOEXTEND ON NEXT 1280K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO
ORCLPDB1(3):ALTER DATABASE DEFAULT TABLESPACE "USERS"
ORCLPDB1(3):Completed: ALTER DATABASE DEFAULT TABLESPACE "USERS"
025-06-12T02:35:20.380577+00:00
ALTER SYSTEM SET control_files='/opt/oracle/oradata/CDB2/control01.ctl' SCOPE=SPFILE;
025-06-12T02:35:20.442979+00:00
ALTER SYSTEM SET local_listener='' SCOPE=BOTH;
ALTER PLUGGABLE DATABASE ORCLPDB1 SAVE STATE
Completed: ALTER PLUGGABLE DATABASE ORCLPDB1 SAVE STATE
```

STEP 6: PREPARE TARGET DATABASE

6.1 Connect to Target Container

```
docker exec -it oracle-cdb2 bash
```

```
source /home/oracle/.bashrc
```

```
export ORACLE_SID=CDB2
```

6.2 Configure TNS Names (same as source)

```
bash-4.2$ cat > $ORACLE_HOME/network/admin/tnsnames.ora << 'EOF'  
> CDB1 =  
>   (DESCRIPTION =  
>     (ADDRESS = (PROTOCOL = TCP)(HOST = oracle-cdb1)(PORT = 1521))  
>     (CONNECT_DATA =  
>       (SERVER = DEDICATED)  
>       (SERVICE_NAME = CDB1)  
>     )  
>   )  
> )  
This command uses the cat utility to create or overwrite the  
tnsnames.ora file, defining network service aliases for three Oracle  
database connections: CDB1, CDB2, and PDB1.  
> CDB2 =  
>   (DESCRIPTION =  
>     (ADDRESS = (PROTOCOL = TCP)(HOST = oracle-cdb2)(PORT = 1521))  
>     (CONNECT_DATA =  
>       (SERVER = DEDICATED)  
>       (SERVICE_NAME = CDB2)  
>     )  
>   )  
>  
> PDB1 =  
>   (DESCRIPTION =  
>     (ADDRESS = (PROTOCOL = TCP)(HOST = oracle-cdb1)(PORT = 1521))  
>     (CONNECT_DATA =  
>       (SERVER = DEDICATED)  
>       (SERVICE_NAME = PDB1)  
>     )  
>   )  
[> EOF  
bash-4.2$
```

6.3 Configure Listener

```
bash-4.2$ cat > $ORACLE_HOME/network/admin/listener.ora << 'EOF'  
> LISTENER =  
>   (DESCRIPTION_LIST =  
>     (DESCRIPTION =  
>       (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))  
>     )  
>   )  
>  
> SID_LIST_LISTENER =  
>   (SID_LIST =  
>     (SID_DESC =  
>       (GLOBAL_DBNAME = CDB2)  
>       (ORACLE_HOME = /opt/oracle/product/21c/dbhome_1)  
>       (SID_NAME = CDB2)  
>     )  
>   )  
[> EOF  
bash-4.2$
```

This command uses the cat utility to create or overwrite the listener.ora file, configuring the Oracle Net Listener to specifically manage the CDB2 database instance.

6.4 Restart Listener

```
bash-4.2$ lsnrctl stop  
  
LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 02:40:20  
  
Copyright (c) 1991, 2021, Oracle. All rights reserved.  
  
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))  
The command completed successfully
```

```
bash-4.2$ lsnrctl start

LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 02:40:27

Copyright (c) 1991, 2021, Oracle. All rights reserved.

Starting /opt/oracle/product/21c/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 21.0.0.0.0 - Production
System parameter file is /opt/oracle/homes/OraDB21Home1/network/admin/listener.ora
Log messages written to /opt/oracle/diag/tnslsnr/a3a5ab7dddef/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))
STATUS of the LISTENER
-----
Alias                      LISTENER
Version                    TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Start Date                 12-JUN-2025 02:40:27
Uptime                     0 days 0 hr. 0 min. 0 sec
Trace Level                off
Security                   ON: Local OS Authentication
SNMP                       OFF
Listener Parameter File   /opt/oracle/homes/OraDB21Home1/network/admin/listener.ora
Listener Log File          /opt/oracle/diag/tnslsnr/a3a5ab7dddef/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))
The listener supports no services
The command completed successfully
```

```
bash-4.2$ lsnrctl status

LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 02:40:31

Copyright (c) 1991, 2021, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))
STATUS of the LISTENER
-----
Alias                      LISTENER
Version                    TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Start Date                 12-JUN-2025 02:40:27
Uptime                     0 days 0 hr. 0 min. 3 sec
Trace Level                off
Security                   ON: Local OS Authentication
SNMP                       OFF
Listener Parameter File   /opt/oracle/homes/OraDB21Home1/network/admin/listener.ora
Listener Log File          /opt/oracle/diag/tnslsnr/a3a5ab7dddef/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))
The listener supports no services
The command completed successfully
bash-4.2$
```

6.5 Shutdown Target Database

```
bash-4.2$ sqlplus / as sysdba ←  
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Jun 12 02:41:41 2025  
Version 21.3.0.0.0  
Copyright (c) 1982, 2021, Oracle. All rights reserved.  
This command launches the SQL*Plus command-line interface and connects to the Oracle database instance with SYSDBA administrative privileges, typically using operating system authentication.  
Connected to:  
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production  
Version 21.3.0.0.0  
SQL> SHUTDOWN IMMEDIATE;  
Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SQL> STARTUP NOMOUNT;  
ORACLE instance started.  
Role: This command starts the Oracle database instance processes and allocates its memory (SGA) but does not yet associate it with a specific database.  
Total System Global Area 935327096 bytes  
Fixed Size 9692536 bytes  
Variable Size 247463936 bytes  
Database Buffers 671088640 bytes  
Redo Buffers 7081984 bytes  
SQL> EXIT;  
Disconnected from Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production  
Version 21.3.0.0.0  
bash-4.2$
```

6.6 Create Initialization File

```
bash-4.2$ cat > $ORACLE_HOME/dbs/initCDB2.ora << 'EOF'  
> db_name=CDB2  
> db_unique_name=CDB2  
> control_files='/opt/oracle/oradata/CDB2/control01.ctl','/opt/oracle/oradata/CDB2/control02.ctl'  
> db_file_name_convert='/opt/oracle/oradata/CDB1/','/opt/oracle/oradata/CDB2/'  
> log_file_name_convert='/opt/oracle/oradata/CDB1/','/opt/oracle/oradata/CDB2/'  
> audit_file_dest='/opt/oracle/admin/CDB2/adump'  
> diagnostic_dest='/opt/oracle'  
> dispatchers='(PROTOCOL=TCP) (SERVICE=CDB2XDB)'  
> enable_pluggable_database=TRUE  
> EOF  
This command uses the cat utility to create or overwrite the initCDB2.ora file, which is an initialization parameter file (PFILE) for the Oracle database instance named CDB2.  
bash-4.2$
```

6.7 Create Required Directories

```
[bash-4.2$ mkdir -p /opt/oracle/oradata/CDB2
[bash-4.2$ mkdir -p /opt/oracle/admin/CDB2/adump
[bash-4.2$ mkdir -p /opt/oracle/fast_recovery_area/CDB2
[bash-4.2$ exit
exit
mac@Younes-farhat-3 ~ %
```

In essence, these `mkdir -p` commands are preparing the necessary directory structure on the file system to house the critical files for your Oracle CDB2 database instance.

6.8 Exit Container

EXIT;

STEP 7: PERFORM RMAN DUPLICATION

7.1 Connect to Source Container

```
mac@Younes-farhat-3 ~ % docker exec -it oracle-cdb1 bash
bash-4.2$ source /home/oracle/.bashrc
bash-4.2$ tnsping CDB2

TNS Ping Utility for Linux: Version 21.0.0.0.0 - Production on 12-JUN-2025 03:36:18

Copyright (c) 1997, 2021, Oracle. All rights reserved.

Used parameter files:
/opt/oracle/homes/OraDB21Home1/network/admin/sqlnet.ora

TNS-03505: Failed to resolve name
bash-4.2$ rman TARGET / AUXILIARY sys/Oracle123@oracle-cdb2:1521/CDB2

Recovery Manager: Release 21.0.0.0.0 - Production on Thu Jun 12 03:36:53 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle and/or its affiliates. All rights reserved.

connected to target database: CDB1 (DBID=1177274117)
connected to auxiliary database: CDB2 (DBID=950607555)

RMAN> %
```

We are accessing the `oracle-cdb1` container, setting up its Oracle environment, and then attempting to verify network connectivity from `oracle-cdb1` to `CDB2` (which is likely in a separate container, `oracle-cdb2`). This is a crucial step to ensure that your containers can communicate with each other's database services.

7.2 Test Network Connectivity tnsping CDB2

7.3 Create RMAN Script

```
vi /home/oracle/duplicate_cdb.rman Add this content: DUPLICATE DATABASE TO CDB2 FROM ACTIVE  
DATABASE PASSWORD FILE SPFILE  
NOFILENAMECHECK  
SET DB_CREATE_FILE_DEST='/opt/oracle/oradata/CDB2'  
SET DB_RECOVERY_FILE_DEST='/opt/oracle/fast_recovery_area/CDB2'  
SET DB_RECOVERY_FILE_DEST_SIZE='10G' SET  
CONTROL_FILES='/opt/oracle/oradata/CDB2/control01.ctl','/opt/oracle/oradata/CDB2/control02.ctl' SET  
LOG_FILE_NAME_CONVERT='/opt/oracle/oradata/CDB1/','/opt/oracle/oradata/CDB2/' SET  
DB_FILE_NAME_CONVERT='/opt/oracle/oradata/CDB1/','/opt/oracle/oradata/CDB2/' SET  
AUDIT_FILE_DEST='/opt/oracle/admin/CDB2/adump' SET DIAGNOSTIC_DEST='/opt/oracle'; 7.4 Run RMAN  
Duplication  
rman TARGET / AUXILIARY sys/Oracle123@CDB2
```

At RMAN prompt:

```
@/home/oracle/duplicate_cdb.rman
```

STEP 8: VERIFY DUPLICATION

8.1 Connect to Target Database

```
# Exit
```

RMAN first, then:

```
exit
```

```
docker exec -it oracle-cdb2 bash
```

```
source /home/oracle/.bashrc
```

```
export ORACLE_SID=CDB2
```

```
sqlplus / as sysdba
```

8.2 Check Database Status

```
SELECT name, open_mode, cdb FROM v$database; SELECT name, open_mode FROM v$pdbs;
```

8.3 Open All PDBs

```
ALTER PLUGGABLE DATABASE ALL OPEN; SELECT name, open_mode FROM v$pdbs;
```

8.4 Verify Test Data -- Connect to PDB1 ALTER SESSION SET

```
CONTAINER=PDB1;
```

```
-- Connect as test_user CONNECT
```

```
test_user/test_pwd@localhost:1521/PDB1
```

```
-- Check data SELECT * FROM test_duplication;
```

8.5 Test Independence

```
-- Add new data to CDB2 to prove independence
```

```
INSERT INTO test_duplication VALUES (4, 'CDB2 Only Record', SYSDATE);
```

```
COMMIT;
```

```
SELECT * FROM test_duplication;
```

8.6 Exit SQL*Plus

```
EXIT; exit
```

STEP 9: FINAL VERIFICATION

9.1 Check Both Databases

```
# Check CDB1 docker exec -it oracle-cdb1 sqlplus test_user/test_pwd@localhost:1521/PDB1
```

```
SELECT * FROM test_duplication;
```

```
-- Should show only 3 records
```

```
EXIT;
```

```
# Check CDB2
```

```
docker exec -it oracle-cdb2 sqlplus test_user/test_pwd@localhost:1521/PDB1
```

```
SELECT * FROM test_duplication; -- Should show 4 records EXIT;
```

Partie 2: Théorie d'Oracle Data Guard

1. Introduction à Oracle Data Guard Qu'est-ce qu'Oracle Data Guard? Oracle Data Guard est une solution complète de reprise après sinistre et de haute disponibilité qui protège les bases de données Oracle contre la perte de données et les temps d'arrêt. Il maintient une ou plusieurs bases de données standby synchronisées pour protéger contre les pannes, les sinistres, les erreurs et la corruption des données.

Objectifs:

Haute Disponibilité:

Minimiser les temps d'arrêt planifiés et non planifiés Capacités de basculement automatique

Récupération rapide après les pannes

Opérations continues de la base de données Assurer la continuité des opérations commerciales

critiques Minimiser l'impact des sinistres Maintenir les niveaux de service pendant les

interruptions

Prise en charge des sites géographiquement distribués.

Reprise après Sinistre:

Protection contre les pannes de site RéPLICATION des données entre les sites Procédures de récupération rapide
Sauvegarde et récupération complètes

2. Composants

Base de Données Primaire

Base de données de production gérant toutes les opérations d'écriture

Source des données redo transmises aux bases de données standby

Peut être une instance unique ou Real Application Clusters (RAC) Maintient les journaux de transaction et les fichiers de données Bases de Données Standby

Copie bloc par bloc identique de la base de données primaire Maintenue par le processus d'application des redo Peut être ouverte en lecture seule tout en appliquant les redo (Active Data Guard) Utilisée pour la reprise après sinistre, les rapports et les opérations de sauvegarde.

Base de Données Standby Logique:

Contient les mêmes informations logiques que la primaire La structure physique peut différer de la primaire Maintenue par le processus SQL Apply Peut être ouverte pour des opérations en lecture/écriture sur des objets locaux Prend en charge des index supplémentaires et des vues matérialisées .

Base de Données Standby Instantanée:

Base de données standby entièrement modifiable Créeée en convertissant une standby physique Reçoit les données redo mais ne les applique pas

Utilisée pour les tests et le développement

Peut être reconvertie en standby physique Services.

Service Redo Apply:

Applique les données redo aux bases de données standby physiques Maintient une copie identique bloc par bloc Prend en charge l'application en temps réel Gère automatiquement la résolution des écarts

Service SQL Apply:

Convertit les données redo en instructions SQL Applique les instructions SQL à la standby logique Prend en charge la transformation des données Permet des modifications locales

Rôle Primaire:

Base de données acceptant les opérations de lecture et d'écriture Génère des données redo pour la transmission Gère le traitement des transactions
Coordonne avec les bases de données standby

Base de données recevant et appliquant les changements Peut être ouverte pour un accès en lecture seule Maintient la synchronisation avec la primaire
Prête pour une transition de rôle

Observateur de Basculement Rapide:

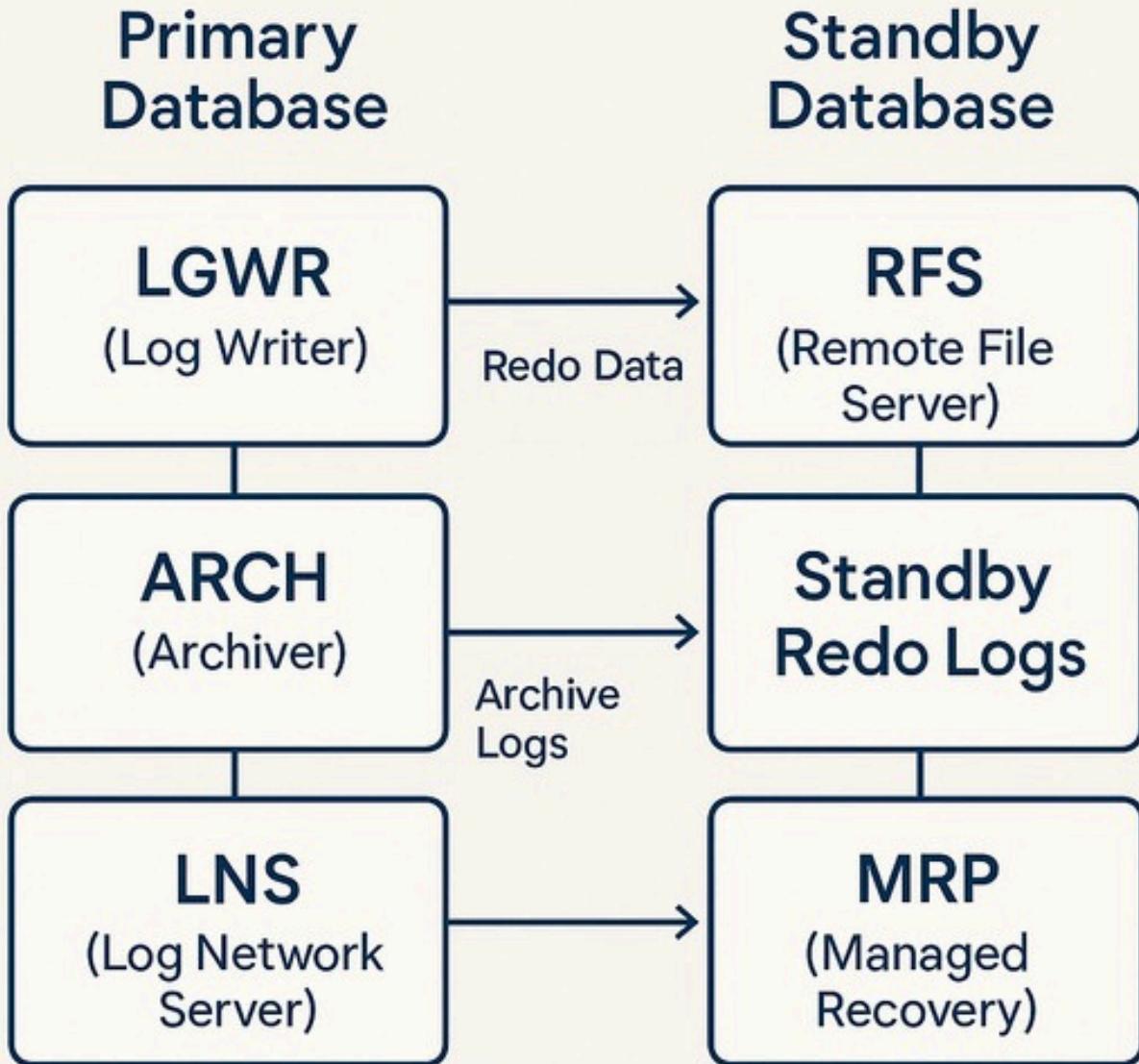
Surveille la configuration Data Guard Initie le basculement automatique Maintient les métadonnées de configuration

Assure la haute disponibilité

3. Architecture et Processus

Architecture de Transport Redo

Architecture de Transport Redo



Processus d'Archivage (ARCH):

Processus Clés

Log Writer (LGWR):

Ecrit les entrées redo dans les journaux redo en ligne Peut transmettre les redo directement à la standby (mode SYNC) Assure la cohérence des transactions Gère le traitement des validations

Processus d'Archivage (ARCH):

Archive les journaux redo en ligne remplis Transmet les journaux archivés à la standby (mode ASYNC) Gère la résolution des écarts Gère les numéros de séquence des journaux

Serveur de Réseau de Journaux (LNS):

Processus dédié pour la transmission des redo Gère la communication réseau Gère les échecs de connexion Optimise l'utilisation de la bande passante réseau

Processus Serveur Réseau (NSP):

Gère les opérations d'E/S réseau Gère plusieurs connexions Fournit un pool de connexions Assure une transmission fiable des données

Processus d'Application Redo:

Processus de Récupération Géré (MRP):

Serveur de Fichiers Distants (RFS):

Reçoit les données redo sur la base de données standby Exécute les redo dans les journaux redo standby Gère les connexions entrantes Gère les erreurs réseau et la récupération

Processus de Récupération Géré (MRP):

Applique les données redo à la standby physique Maintient la cohérence de la base de données Gère les opérations de récupération Gère la surveillance du retard d'application

Processus Standby Logique (LSP):

Convertit les redo en instructions SQL Applique les SQL à la standby logique Gère les conversions de type de données Gère les règles de transformation

4. Types de Configuration Protection Maximale Garantie de zéro perte de données

La base de données primaire s'arrête si elle ne peut pas écrire sur la standby

Nécessite un transport redo SYNC vers au moins une standby

Niveau de protection des données le plus élevé

Peut impacter les performances de la base de données primaire

Utilisé pour les applications critiques Disponibilité Maximale Zéro perte de données dans

des conditions normales La primaire continue si toutes les standbys deviennent

indisponibles Se resynchronise automatiquement lorsque la connectivité est rétablie

Équilibre protection et disponibilité Préféré pour la plupart des environnements de

production Offre le meilleur des deux mondes Performance Maximale Impact minimal

sur les performances de la base de données primaire

Utilise un transport redo asynchrone (ASYNC)

Risque de perte de données minimale pendant les pannes

Mode de protection par défaut

Meilleures caractéristiques de performance Adapté à la plupart des applications 5.

Gestion du Switchover et du Failover

Switchover (Transition de Rôle Planifiée)

Caractéristiques:

Inversion de rôle planifiée entre primaire et standby Opération à zéro perte de données

Les deux bases de données restent disponibles Utilisé pour la maintenance, les tests ou
l'équilibrage de charge

Procédure:

Vérifier l'état de la configuration Changer les destinations du

transport redo Convertir la primaire en rôle standby

Convertir la standby en rôle primaire

Démarrer le transport redo vers la nouvelle standby Vérifier la

nouvelle configuration **Commandes:**

-- *Utilisation de DGMGRL*

DGMGRL> SWITCHOVER TO 'standby_db_name';

-- *Switchover manuel*

-- **Sur la primaire:**

ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY;

-- **Sur la standby:**

ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;

Failover (Transition de Rôle Non Planifiée)

Transition non planifiée lorsque la primaire échoue La base de données standby devient la nouvelle primaire La primaire d'origine doit être réintégrée ou reconstruite Peut impliquer une certaine perte de données selon le mode de protection

Types:

Failover Complet: Tous les redo disponibles sont appliqués Failover Immédiat:

Activation immédiate sans appliquer tous les redo Failover avec Term: Failover avec un identifiant de terme spécifique

Procédure:

Basculement Rapide

Basculement automatique sans intervention humaine Nécessite le processus Observateur Data Guard Conditions et seuils de basculement configurables Maintient la disponibilité des applications

Exigences:

Configuration Data Guard Broker Processus Observateur en cours d'exécution
Basculement Rapide activé Mode de protection approprié

Configuration:

DGMGRL> ENABLE FAST_START FAILOVER;

DGMGRL> START OBSERVER;

6. Outils de Gestion

DGMGRL (Data Guard Manager)

Fonctionnalités Clés:

Interface de gestion centralisée Tâches de configuration automatisées Surveillance de l'état et alertes Transitions de rôle simplifiées Implémentation des meilleures pratiques

Commandes Courantes:

-- Créer une configuration

```
CREATE CONFIGURATION 'DGConfig' AS  
PRIMARY DATABASE IS 'primary_db'  
CONNECT IDENTIFIER IS 'primary_tns';
```

-- Ajouter une base de données standby

```
ADD DATABASE 'standby_db' AS  
CONNECT IDENTIFIER IS 'standby_tns';
```

-- Activer la configuration

ENABLE CONFIGURATION;

SHOW CONFIGURATION;

SHOW DATABASE 'database_name';-- *Surveiller le retard*

SHOW DATABASE 'standby_db' 'ApplyLag';

SHOW DATABASE 'standby_db' 'TransportLag';-- *Effectuer un switchover*

SWITCHOVER TO 'standby_db';

EDIT CONFIGURATION SET PROTECTION MODE AS MAXIMUM AVAILABILITY;

Oracle Enterprise Manager**Capacités:**

Interface utilisateur graphique

Surveillance en temps réel Analyse des performances

Tâches de gestion automatisées

Intégration avec d'autres outils Oracle

Considérations de Sécurité et Performance

Sécurité:

Chiffrement réseau pour le transport redo

Authentification entre primaire et standby

Canaux de communication sécurisés

Contrôle d'accès et audit **Performance:**

Exigences de bande passante réseau

Surveillance du retard d'application

Optimisation du transport redo

Planification de l'allocation des ressources

Conclusion

La duplication de CDB Oracle et Oracle Data Guard sont deux solutions complémentaires essentielles pour assurer la haute disponibilité, la continuité d'activité et la protection des données dans les environnements Oracle modernes.

La duplication de CDB permet de créer rapidement une copie exacte d'une base de données conteneur, facilitant ainsi les tests, le développement, et la mise en place de bases de données de secours. Grâce à RMAN, ce processus est automatisé et flexible, offrant plusieurs modes adaptés à différents besoins.

Oracle Data Guard, quant à lui, constitue une solution robuste pour la reprise après sinistre et la gestion des basculements entre bases primaires et standby. Son architecture sophistiquée et ses différents modes de protection permettent d'équilibrer performance, disponibilité et sécurité des données, tout en minimisant les risques de perte d'information. En combinant ces deux technologies, les organisations peuvent renforcer significativement la résilience de leur infrastructure Oracle, garantir la continuité de leurs services critiques, et réduire les temps d'arrêt en cas d'incident.