

# Database

- Oracle
  - Acces base oracle
  - Vérifier et débloquent un compte verrouillé
  - Problème demarrage timezone
- MYSQL
  - PB SQL FAILOVER
  - Commandes General
  - Dump/Restore
  - Mysql innodb cache info
  - Modification du compte mysql\_monitor
- POSTGRES
  - TimescaleDB Update
  - Création user en lecture seul
  - Modification du compte Monitor
  - Vacuum
  - Splitbrain
  - Commande Postgresql
- SQLPLUS
- Notes Général

Oracle

Oracle

# Acces base oracle

Sudo su - oracle

Setup l'affichage et voir les session connecte à la basse :

```
SET pages 400 lines 200
COLUMN spid FORMAT A10
COLUMN username FORMAT A20
COLUMN program FORMAT A45
```

```
SELECT s.inst_id,
       s.sid,
       s.serial#,
       p.spid,
       s.username,
       s.program
FROM   gv$session s
       JOIN gv$process p ON p.addr = s.paddr AND p.inst_id = s.inst_id
WHERE  s.type != 'BACKGROUND';
```

Générer le path et sid

```
export PATH=$PATH:/app/oracle/product/11.2.0.4/dbhome_1/bin/
export ORACLE_SID=Nom database
export ORACLE_HOME=/app/oracle/product/11.2.0.4/dbhome_1/
```

. oraenv mettre le sid et path en rapport

```
sqlplus / as sysdba
```

```
drop user TABLE cascade;
```

# Vérifier et débloquent un compte verrouillé

- Se connecter au serveur en SSH via putty ou winscp (> putty).
- Se connecter en tant que compte oracle  
su oracle

- Ajouter les arguments pour les variables d'environnement pour se connecter à l'instance

```
export ORACLE_HOME=/u01/app/oracle/product/11gr2/db_1  
export PATH=$PATH:$ORACLE_HOME/bin  
export ORACLE_SID=gcldev
```

- Se connecter à l'instance SQL:

```
sqlplus /nolog  
connect / as sysdba
```

- Lister tous les comptes de l'instance SQL :

```
SELECT username, account_status FROM dba_users;
```

- Lister tous les comptes avec leurs dates de création, date d'expiration et l'état de verrouillage :

```
SELECT username, account_status, created, lock_date, expiry_date FROM dba_users WHERE account_status !=  
'OPEN';
```

- Requête pour débloquent un compte (nomducompte à remplacer) :

```
ALTER USER nomducompte ACCOUNT UNLOCK;
```

- Note en plus pour reset un password :

```
ALTER USER nomducompte BY new_password;
```

# Problème démarrage timezone

```
ps -ef | grep pmon
```

```
[root@prdorani1 diag]# ps -ef | grep pmon
oracle      5547      1  0 12:15 ?        00:00:00 ora_pmon_PIMS1
oracle     12253      1  0 12:23 ?        00:00:00 ora_pmon_RECPIMS1
grid       16367      1  0 11:25 ?        00:00:00 asm_pmon_+ASM1
grid       16851      1  0 11:25 ?        00:00:00 apx_pmon_+APX1
oracle     16863      1  0 11:25 ?        00:00:01 ora_pmon_ADIX51
oracle     16867      1  0 11:25 ?        00:00:01 ora_pmon_BIDATA1
oracle     16913      1  0 11:25 ?        00:00:01 ora_pmon_PRDBD111
oracle     17311      1  0 11:25 ?        00:00:01 ora_pmon_RHINTERM1
oracle     17406      1  0 11:26 ?        00:00:01 ora_pmon_PRDWEB1
oracle     17416      1  0 11:26 ?        00:00:01 ora_pmon_NODHOS1
oracle     17607      1  0 11:26 ?        00:00:01 ora_pmon_PRDBD1
root       27555  26395  0 13:56 pts/4    00:00:00 grep --color=auto pmon
```

Pour redémarrer les bases manuellement :

```
export ORACLE_HOME=/app/oracle/product/10.2.0.5SE/dbhome_1
export ORACLE_BASE=/app/oracle/product/10.2.0.5SE/dbhome_1
export ORACLE_SID=PIMS1
sqlplus / as sysdba
```

Démarrer la base

```
startup
```

Pour arrêter la base:

```
shutdown immediate
```

```
=====
=====
```

Lorsque l'on exécute la commande (sqlplus / as sysdba) surtout sur la base l'on obtient le message d'erreur suivant :

**ORA-01804: failure to initialize timezone information**

Solution :

Prendre un autre fichier timezone sur n'importe quel répertoire :

```
export ORA_TZFILE=/app/oracle/product/11.2.0.4/dbhome_1/oracore/zoneinfo/timetzlrg_14.dat
```

=====

Pour supprimer les logs de listener de plus de Xjour (exemple +24)

```
cd /app/oracle/diag/tnslsnr/prdoran1/listener/alert
find . -name "log_*.xml" -ctime +24 -exec rm -f {} \;
```

Pour lister les traces +15 j :

```
find . -name "*.trc" -ctime +15 -exec ls -l {} \;
```

Avant de remplacer "ls -l" par "rm -f" assure toi que la base est démarrée depuis plus de 15 j

=====

Il existe des bases selon la version d'oracle, pour les lister :

```
cd /app/oracle/product/
```

Ensuite la version ex :

```
cd /app/oracle/product/11.2.0.4/dbhome_1/dbs/
```

Ce qui nous intéresse c'est le init\* pour savoir le nom de la base (ex: la base initNODHOS1.ora ça représente la base NODHOS1 donc pour l'utiliser on doit lancer l'export :

```
export ORACLE_HOME=/app/oracle/product/11.2.0.4/dbhome_1
```

ensuite :

```
export ORACLE_SID=NODHOS1
```

```

-rw-r----- 1 oracle oinstall      60 14 nov.  2019 initADIX51.ora
-rw-r----- 1 oracle oinstall      35 14 nov.  2019 initADIX51.ora.bak.prdoran1
-rw-r--r-- 1 oracle oinstall      62 16 oct.  2019 initBIDATA1.ora
-rw-r--r-- 1 oracle oinstall      37 16 oct.  2019 initBIDATA1.ora.bak.prdoran1
-rw-r--r-- 1 oracle oinstall      62 12 nov.  2019 initNODHOS1.ora
-rw-r--r-- 1 oracle oinstall      37 12 nov.  2019 initNODHOS1.ora.bak.prdoran1
-rw-r--r-- 1 oracle oinstall    2851 15 mai   2009 init.ora
-rw-r--r-- 1 oracle oinstall      64 14 nov.  2019 initPRDBD111.ora
-rw-r--r-- 1 oracle oinstall      39 14 nov.  2019 initPRDBD111.ora.bak.prdoran1
-rw-r--r-- 1 oracle oinstall      60 13 nov.  2019 initPRDBD1.ora
-rw-r--r-- 1 oracle oinstall      35 13 nov.  2019 initPRDBD1.ora.bak.prdoran1
-rw-r--r-- 1 oracle oinstall      62 16 oct.  2019 initPRDWEB1.ora
-rw-r--r-- 1 oracle oinstall      37 16 oct.  2019 initPRDWEB1.ora.bak.prdoran1
-rw-r--r-- 1 oracle oinstall      66 12 nov.  2019 initRHINTERM1.ora
-rw-r--r-- 1 oracle oinstall      41 12 nov.  2019 initRHINTERM1.ora.bak.prdoran1
-rw-r----- 1 oracle oinstall    3584 14 nov.  2019 orapwADIX5
-rw-r----- 1 oracle oinstall    3584 14 nov.  2019 orapwADIX51
-rw-r----- 1 oracle oinstall    3584 16 oct.  2019 orapwBIDATA
-rw-r----- 1 oracle oinstall    3584 16 oct.  2019 orapwBIDATA1
-rw-r----- 1 oracle oinstall    3584 16 oct.  2019 orapwMODHOS

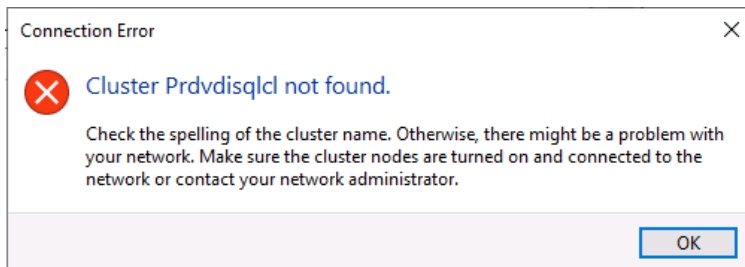
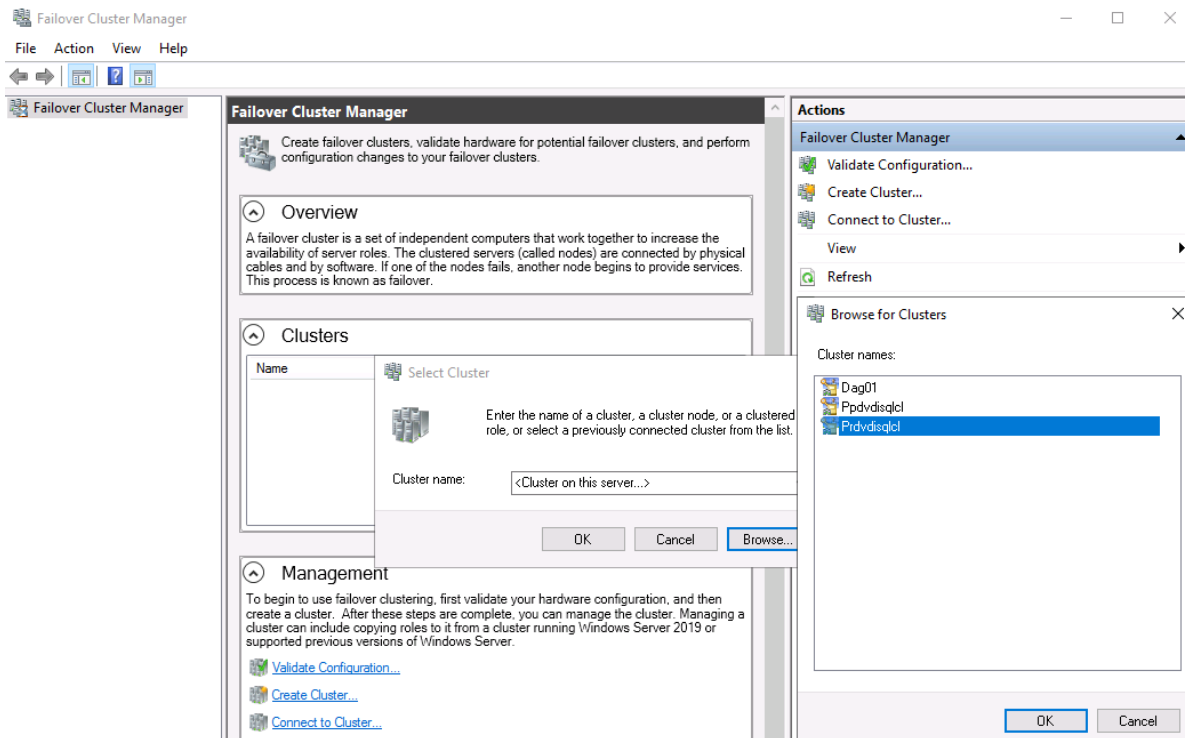
```

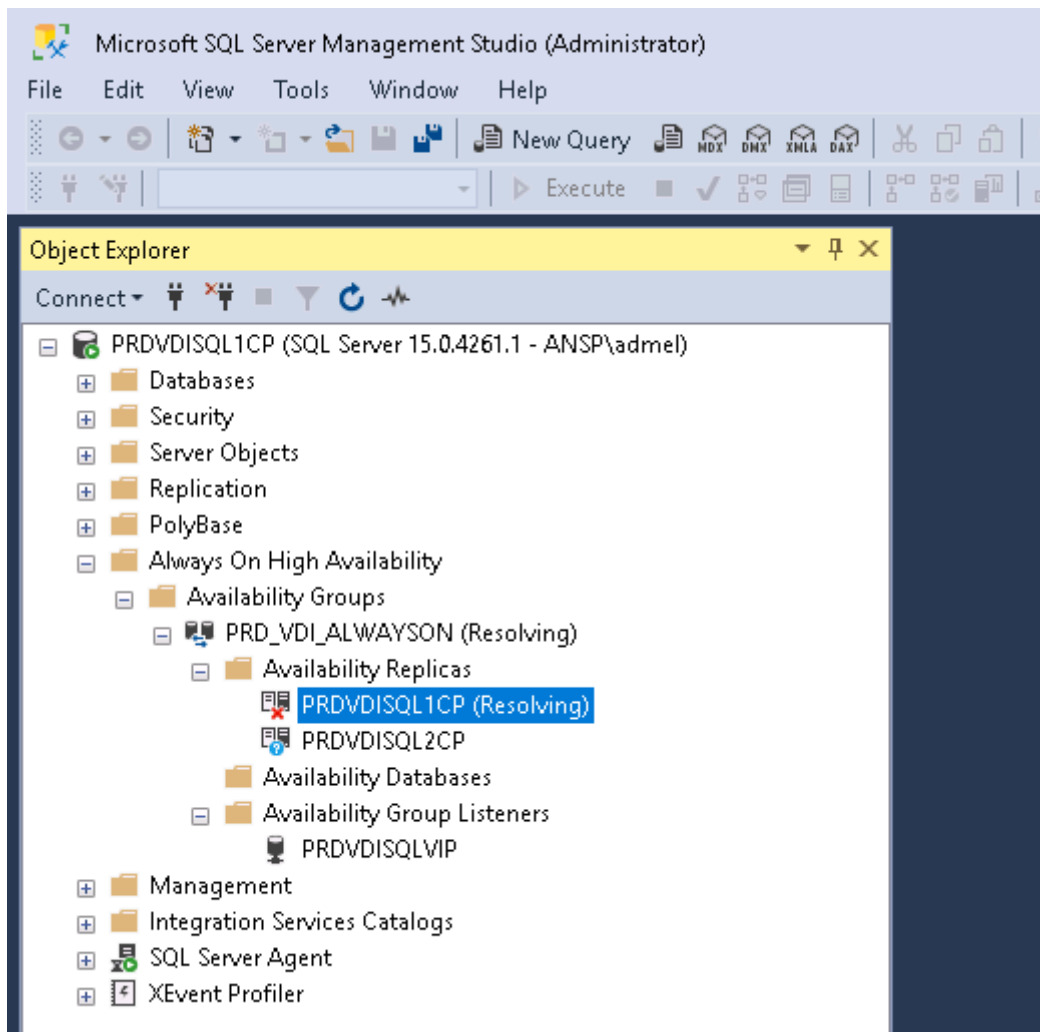
Après on va pouvoir lancer sqlplus en tant que NODHOS1 et démarrer la base



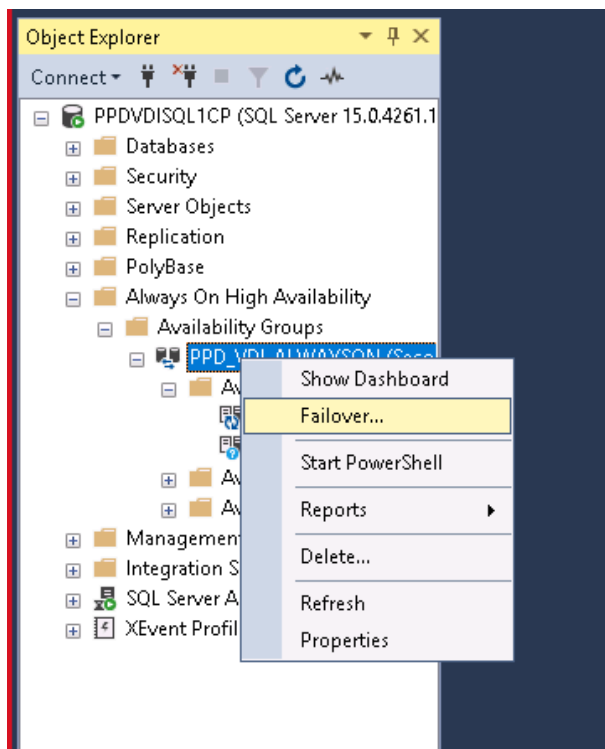
# MYSQL

# PB SQL FAILOVER

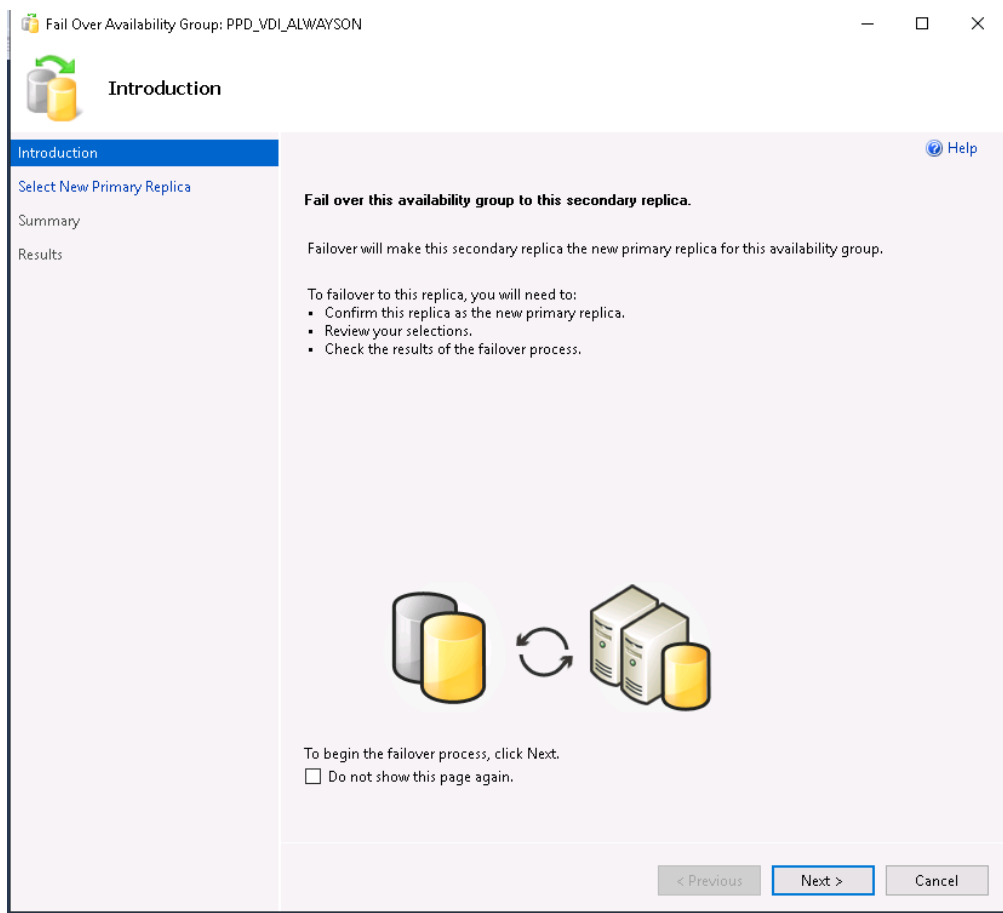




Clic droit --> Failover...



Next



Next



## Select New Primary Replica

Introduction

Select New Primary Replica

Summary

Results

Help

### Select the new primary replica for this availability group.

Current Primary Replica: PPDVDISQL2CP  
Primary Replica Status: Synchronous commit and Unknown  
Quorum Status: Normal Quorum

Choose new primary replica:

	Server Instance	Availability Mode	Failover Mode	Failover Read...	Warnings	R
<input checked="" type="checkbox"/>	PPDVDISQL1CP	Synchronous co...	Automatic	No data loss		Se

< >

Refresh

< Previous

Next >

Cancel

Finish



## Summary

Introduction

Select New Primary Replica

Summary

Results

Help

### Verify the choices made in this wizard.

Click Finish to perform the following actions:

- Current Primary Replica: PPDVDISQL2CP
- New Primary Replica: PPDVDISQL1CP
- Failover Actions: No data loss
- Affected Databases
  - PPDLogging
  - PPDReporting
  - PPDSite
  - PPDWem

Script ▼

< Previous

Finish

Cancel



## Progress

Introduction

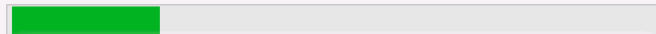
Select New Primary Replica

Summary

Results

Help

Performing manual failover to secondary replica 'PPDVDISQL1CP'

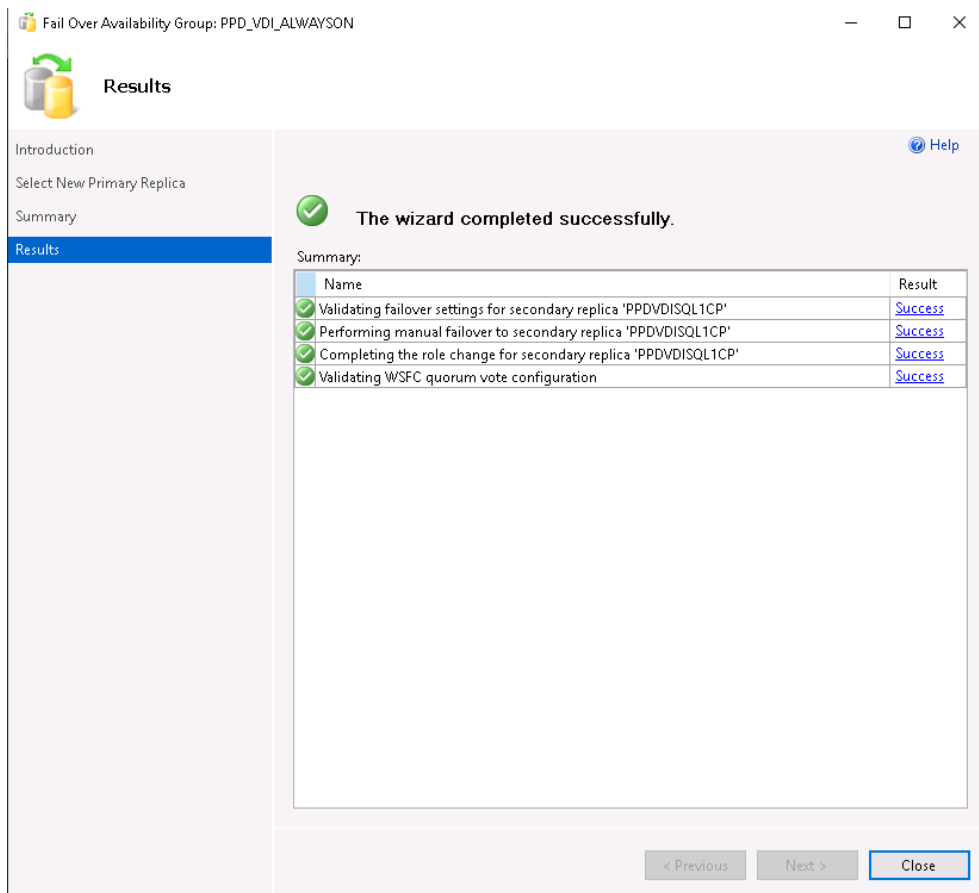


More details

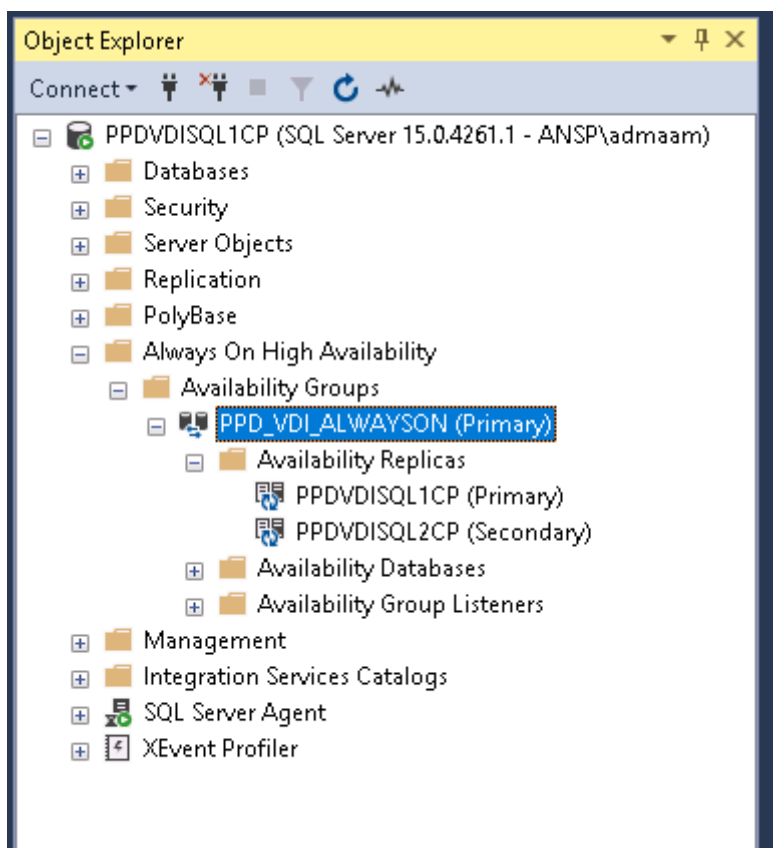
< Previous

Next >

Cancel



Vérifier que le 1 est bien passé en Primary ainsi que le 2 en Secondary



# Commandes General

## Poids base de données

```
SELECT sum( data_length + index_length) / 1024 / 1024 "Nom-de-la-base Taille en Mo" FROM  
information_schema.TABLES WHERE table_schema = "nom-de-base" GROUP BY table_schema;
```

## Read permission

```
GRANTSELECTON DBNAME.TABLE_NAME TO USERNAME
```

## liste toutes les BDD

```
show databases;
```

## se met sur la BDD qui nous intéresse

```
Use nomdelabdd;
```

## Suppression de la BDD

```
Drop database nomdelabdd;
```

## Création de la BDD

```
Create database nomdelabdd;
```

## Dans le cas où le compte mysqldump n'a pas les droits de restauration de dump

```
grant all privileges on *.* to mysqldump@localhost;
```

## Restauration du dump dans la BDD

```
mysql -u mysqldump -p "nomdelabdd" < "/production/mysql_dump/nomdelabdd.sql"
```

## Vérification de la bonne restauration du dump



```
ls -alsh /production/mysql/nomdelabdd/
```

### Pour se connecter

```
mysql
```

### Pour avoir la taille de toutes les BDD:

```
SELECT table_schema AS "Database", ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)" FROM information_schema.TABLES GROUP BY table_schema;
```

### Pour avoir la taille de toutes les tables d'une BDD:

```
SELECT table_name AS "Table", ROUND(((data_length + index_length) / 1024 / 1024), 2) AS "Size (MB)" FROM information_schema.TABLES WHERE table_schema = "database_name" ORDER BY (data_length + index_length) DESC;
```

### Reset Mot de passe Mysql:

```
ALTER USER 'mysqldump'@'localhost' IDENTIFIED BY 'dump';
```

# Dump/Restore

## Création de dump Mysql:

```
/usr/bin/mysqldump -u mysqldump -p name | gzip -c --best > /production/mysql_dump/Save_25.sql.gz
```

## Dezip d'un dump au format sql.gz :

```
gunzip /production/mysql_dump/Save_25.sql.gz /tmp
```

Attention cela remplace le fichier d'origine en .sql

## Affichage de tous les droits sur une BDD:

```
Use name;  
SELECT host,user,password,Grant_priv,Super_priv FROM mysql.user;
```

## Restauration d'un dump Mysql avec le dump au format .sql :

**show databases;** --> liste toutes les BDD

**Use nomdelabdd;** --> On se met sur la BDD qui nous intéresse

**Drop database nomdelabdd;** --> Suppression de la BDD actuelle

**Create database nomdelabdd;** --> Création de la BDD

**grant all privileges on \*.\* to mysqldump@localhost;** --> Dans le cas où le compte mysqldump n'a pas les droits de restauration de dump

**mysql -u mysqldump -p nomdelabdd < /production/mysql\_dump/nomdelabdd.sql** -->

Restauration du dump dans la BDD nouvellement créée

**ls -alsh /production/mysql/nomdelabdd/** --> Vérification de la bonne restauration du dump, voir s'il y a des éléments récents

Pour info, utiliser la commande **pv** lors de la restauration d'un dump afin de voir en live ce que fait la restauration:

# Mysql innodb cache info

Vérifier taille Buffer Pool Size:

mysql

```
SELECT FORMAT (BufferPoolPages*PageSize/POWER(1024,3),2) BufferPoolDataGB FROM (SELECT variable_value  
BufferPoolPages FROM information_schema.global_status WHERE variable_name  
='Innodb_buffer_pool_pages_total') A, (SELECT variable_value PageSize FROM information_schema.global_status  
WHERE variable_name ='Innodb_page_size') B;
```

```
[(none)] mysql> SELECT FORMAT(BufferPoolPages*PageSize/POWER(1024,3),2) BufferPoolDataGB FROM  
-> (SELECT variable_value BufferPoolPages FROM information_schema.global_status  
-> WHERE variable_name = 'Innodb_buffer_pool_pages_total') A,  
-> (SELECT variable_value PageSize FROM information_schema.global_status  
-> WHERE variable_name = 'Innodb_page_size') B;  
+-----+  
| BufferPoolDataGB |  
+-----+  
| 0.25            |  
+-----+  
1 row in set (0.003 sec)
```

Vérifier données en cache:

Mysql

```
SELECT FORMAT (BufferPoolPages*PageSize/POWER(1024,3),2) BufferPoolDataGB FROM (SELECT variable_value  
BufferPoolPages FROM information_schema.global_status WHERE variable_name  
='Innodb_buffer_pool_pages_data') A, (SELECT variable_value PageSize FROM information_schema.global_status  
WHERE variable_name ='Innodb_page_size') B;
```

```
[(none)] mysql> SELECT FORMAT(BufferPoolPages*PageSize/POWER(1024,3),2) BufferPoolDataGB FROM  
-> (SELECT variable_value BufferPoolPages FROM information_schema.global_status  
-> WHERE variable_name = 'Innodb_buffer_pool_pages_data') A,  
-> (SELECT variable_value PageSize FROM information_schema.global_status  
-> WHERE variable_name = 'Innodb_page_size') B;  
+-----+  
| BufferPoolDataGB |  
+-----+  
| 0.10            |  
+-----+  
1 row in set (0.003 sec)
```

Vérifier % d'utilisation en cache:

```
SELECT CONCAT (FORMAT (DataPages*100.0/TotalPages,2),' %') BufferPoolDataPercentage FROM (SELECT  
variable_value DataPages FROM information_schema.global_status WHERE variable_name  
='Innodb_buffer_pool_pages_data') A, (SELECT variable_value TotalPages FROM  
information_schema.global_status WHERE variable_name ='Innodb_buffer_pool_pages_total') B;
```

```
[(none)] mysql> SELECT CONCAT(FORMAT(DataPages*100.0/TotalPages,2),' %') BufferPoolDataPercentage FROM  
-> (SELECT variable_value DataPages FROM information_schema.global_status  
-> WHERE variable_name = 'Innodb_buffer_pool_pages_data') A,  
-> (SELECT variable_value TotalPages FROM information_schema.global_status  
-> WHERE variable_name = 'Innodb_buffer_pool_pages_total') B;  
+-----+  
| BufferPoolDataPercentage |  
+-----+  
| 40.72 % |  
+-----+  
1 row in set (0.003 sec)
```

Données supplémentaires sur le cache:

```
SHOW GLOBAL STATUS LIKE 'Innodb_buffer_pool%';
```

```
[(none)] mysql> SHOW GLOBAL STATUS LIKE 'Innodb_buffer_pool%';  
+-----+  
| Variable_name | Value |  
+-----+  
| Innodb_buffer_pool_dump_status | |  
| Innodb_buffer_pool_load_status | Buffer pool(s) load completed at 220905 17:35:17 |  
| Innodb_buffer_pool_resize_status | |  
| Innodb_buffer_pool_load_incomplete | OFF |  
| Innodb_buffer_pool_pages_data | 6671 |  
| Innodb_buffer_pool_bytes_data | 109297664 |  
| Innodb_buffer_pool_pages_dirty | 0 |  
| Innodb_buffer_pool_bytes_dirty | 0 |  
| Innodb_buffer_pool_pages_flushed | 131 |  
| Innodb_buffer_pool_pages_free | 9699 |  
| Innodb_buffer_pool_pages_misc | 14 |  
| Innodb_buffer_pool_pages_total | 16384 |  
| Innodb_buffer_pool_read_ahead_rnd | 0 |  
| Innodb_buffer_pool_read_ahead | 0 |  
| Innodb_buffer_pool_read_ahead_evicted | 0 |  
| Innodb_buffer_pool_read_requests | 25646 |  
| Innodb_buffer_pool_reads | 6541 |  
| Innodb_buffer_pool_wait_free | 0 |  
| Innodb_buffer_pool_write_requests | 843 |  
+-----+  
19 rows in set (0.001 sec)
```

Statut Innodb:

```
SHOW ENGINE INNODB STATUS;
```

Buffer hit ratio:

```
SELECT round ((P2.variable_value / P1.variable_value),4), P2.variable_value, P1.variable_value FROM
information_schema.GLOBAL_STATUS P1, information_schema.GLOBAL_STATUS P2 WHERE P1. variable_name
='innodb_buffer_pool_read_requests'ANDP2. variable_name ='innodb_buffer_pool_reads';
```

```
[(none)] mysql> SELECT round ((P2.variable_value / P1.variable_value),4),
-> P2.variable_value, P1.variable_value
-> FROM information_schema.GLOBAL_STATUS P1,
-> information_schema.GLOBAL_STATUS P2
-> WHERE P1. variable_name = 'innodb_buffer_pool_read_requests'
-> AND P2. variable_name = 'innodb_buffer_pool_reads';
+-----+-----+-----+
| round ((P2.variable_value / P1.variable_value),4) | variable_value | variable_value |
+-----+-----+-----+
| 0.2550 | 6541 | 25646 |
+-----+-----+-----+
1 row in set (0.003 sec)
```

# Modification du compte mysql\_monitor

```
mysql
```

```
SHOW DATABASES;
```

```
select user, host from mysql.users;
```

```
UPDATE user SET password=PASSWORD('PASSWORD') WHERE user='mysql_monitor';
```

si la commande ci-dessous ne fonctionne pas

```
ALTER USER 'mysql_monitor' IDENTIFIED BY '*****';
```

```
FLUSH PRIVILEGES;
```

# POSTGRES

# TimescaleDB Update

```
[root@intpostgresql ~]# sudo -u [redacted] -i psql
psql (14.6, server 12.13)
Type "help" for help.

[redacted]=# \c [redacted]
psql (14.6, server 12.13)
You are now connected to database "[redacted]" as user "[redacted]".

[redacted]=# \dx

              List of installed extensions
  Name      | Version | Schema  | Description
-----+-----+-----+-----
plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
timescaledb | 2.8.0   | public   | Enables scalable inserts and complex queries for time-series data
```

Ajouter le proxy avec la commande export

## Yum update

```
Package Arch Version Repository Size
-----+-----+-----+-----+-----
Updating:
timescaledb-2-loader-postgresql-12 x86_64 2.11.2-0.el7 timescale_timescaledb 85 k
timescaledb-2-loader-postgresql-14 x86_64 2.11.2-0.el7 timescale_timescaledb 88 k
timescaledb-2-postgresql-12 x86_64 2.11.2-0.el7 timescale_timescaledb 2.7 M
timescaledb-2-postgresql-14 x86_64 2.11.2-0.el7 timescale_timescaledb 2.8 M
timescaledb-tools x86_64 0.14.3-0.el7 timescale_timescaledb 2.8 M

Transaction Summary
Upgrade 5 Packages

Total download size: 8.5 M
Is this ok [y/d/N]: n
```

Installation à la mano:

yum update timescaledb-2-postgresql-12

systemctl restart postgresql-12

timescaledb-tune --pg-config=/usr/pgsql-12/bin/pg\_config



```
[root@intpostgresql ~]# timescaledb-tune --pg-config=/usr/pgsql-12/bin/pg_config
Using postgresql.conf at this path:
/var/lib/postgres/data/postgresql.conf

Is this correct? [(y)es/(n)o]: y
Writing backup to:
/tmp/timescaledb_tune.backup202309071132

success: shared_preload_libraries is set correctly

Tune memory/parallelism/WAL and other settings? [(y)es/(n)o]: y
Recommendations based on 15.51 GB of available memory and 8 CPUs for PostgreSQL 12

Memory settings recommendations
Current:
work_mem = 10167kB
Recommended:
work_mem = 5083kB
Is this okay? [(y)es/(s)kip/(q)uit]: s
warning: memory settings left alone, but still need tuning

Parallelism settings recommendations
Current:
timescaledb.max_background_workers = 8
max_worker_processes = 15
max_parallel_workers_per_gather = 2
missing: max_parallel_workers
Recommended:
timescaledb.max_background_workers = 16
max_worker_processes = 27
max_parallel_workers_per_gather = 4
max_parallel_workers = 8
Is this okay? [(y)es/(s)kip/(q)uit]: y
success: parallelism settings will be updated

WAL settings recommendations
Current:
Recommended:
Is this okay? [(y)es/(s)kip/(q)uit]: s
warning: WAL settings left alone, but still need tuning

Background writer settings recommendations
Current:
Recommended:
Is this okay? [(y)es/(s)kip/(q)uit]: s
warning: background writer settings left alone, but still need tuning

Miscellaneous settings recommendations
success: miscellaneous settings are already tuned
Saving changes to: /var/lib/postgres/data/postgresql.conf
```

sudo -u "user" -i psql

\c database

\dx

ALTER EXTENSION timescaledb UPDATE;

```
:=# \dx

          List of installed extensions
  Name      | Version | Schema | Description
-----+-----+-----+-----
 plpgsql    | 1.0     | pg_catalog | PL/pgSQL procedural language
 timescaledb | 2.11.2  | public   | Enables scalable inserts and complex queries for time-series data
(2 rows)
```

\q

Infos supplémentaire:

Je vois bien la dernière version de timescaledb dans pg\_available\_extensions, mais la version installée sur la BDD reste toujours la 2.8.1.

```
mass=> select * from pg_available_extensions;
```

name	default_version	installed_version	comment
plpgsql	1.0	1.0	PL/pgSQL procedural language
timescaledb	2.11.2	2.8.1	Enables scalable inserts and complex queries for time-series data (Apache 2 Edition)

```
mass=> \dx
```

Liste des extensions installées			
Nom	Version	Schéma	Description
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language
timescaledb	2.8.1	public	Enables scalable inserts and complex queries for time-series data

Il faudrait donc mettre à jour la version de timescaledb sur la BDD MASS.

Exemple :

- **ALTER EXTENSION timescaledb UPDATE TO '2.11.2';**

<https://www.postgresql.org/docs/12/sql-alterextension.html>

- Ou si ça ne fonctionne pas, désinstaller puis réinstaller :

**DROP EXTENSION timescaledb CASCADE;**

**CREATE EXTENSION IF NOT EXISTS timescaledb;**

# Création user en lecture seul

```
sudo -u postgres -i
```

```
CREATE USER servicenav LOGIN PASSWORD 'Password';  
GRANT CONNECT ON DATABASE contact TO servicenav;  
GRANT CONNECT ON DATABASE side TO servicenav;  
GRANT CONNECT ON DATABASE vac TO servicenav;
```

```
GRANT USAGE ON SCHEMA public TO servicenav;  
GRANT SELECT ON ALL TABLES IN schema public TO servicenav;
```

# Modification du compte Monitor

```
sudo -u postgres -i
cd /production/pgsql/data/
```

Modification :

```
vim /production/pgsql/data/pg_hba.conf
```

```
#host all monitor 10.0.0.0/8 password
host all monitor 10.1.1.85/32 password
```

Modification mot de passe compte postgresql monitor :

```
psql test
```

```
test=# \du
```

List of roles

Role name	Attributes	Member of
test		{ }
monitor	Superuser	{ }
postgres	Superuser, Create role, Create DB, Replication, Bypass RLS	{ }
postgresdump	Superuser	{ }

```
ALTER USER monitor WITH PASSWORD 'password';
```

```
systemctl reload postgresql-12
```

Nb : pour avoir le nom exact du service : `systemctl --type=service | grep postgre`

POSTGRES

# Vacuum

```
sudo -u postgres -i psql
```

```
postgres=# \c "nom_de_la_bdd"
```

```
nom_de_la_bdd=# VACUUM VERBOSE ANALYZE;
```

```
nom_de_la_bdd=# \q
```

# Splitbrain

## Reconstruction du cluster Postgresql sous Centos

Premièrement sauvegarder la VM via un quick backup du côté du serveur VEEAM ou autre

Deuxièmement : vérifier qu'il n'existe pas de job de dump sql sur les serveurs membres du cluster

UNIQUEMENT QUAND CECI EST FAIT désactiver le service PostgreSQL sur le Node qui sera le slave/standby

```
Service postgres12 stop
```

Dans le cas où il risque d'y avoir des cron de sauvegarde ou autre désactiver aussi le service crond

```
Service crond stop
```

Puis, toujours sur le node standby lancez cette commande en se positionnant sur un dossier qui est "possédé" par le user qui a accès à la/aux base/s ici c'est postgres et on effectue à partir du dossier **/production/pgsql**

```
/usr/pgsql-12/bin/repmgr -h POSTGRESQL1.repli -U repmgr -d repmgr standby clone -F
```

Ici POSTGRESQL1 est le node master

Afin de forcer la synchronisation des données entre le nœud master et standby

```
[root@xxxxxxxxxxxxx :/production/pgsql]
# sudo su postgres
bash-4.2$ pwd
/production/pgsql
bash-4.2$ /usr/pgsql-12/bin/repmgr -h IXXXXXXXXXXXXXXXXXXXXX -U repmgr -d repmgr standby clone -F
NOTICE: destination directory "/production/pgsql/12/data" provided
INFO: connecting to source node
DETAIL: connection string is: host=fXXXXXXXXXXXXXXXXXXXXX .repli user=IXXXXXXXXXXXXX dbname=rXXXXXXXXXXXXX
DETAIL: current installation size is 1093 GB
INFO: replication slot usage not requested; no replication slot will be set up for this standby
NOTICE: checking for available walsenders on the source node (2 required)
NOTICE: checking replication connections can be made to the source server (2 required)
WARNING: data checksums are not enabled and "wal_log_hints" is "off"
DETAIL: pg_rewind requires "wal_log_hints" to be enabled
INFO: checking and correcting permissions on existing directory "/production/pgsql/12/data"
NOTICE: starting backup (using pg_basebackup)...
HINT: this may take some time; consider using the -c/--fast-checkpoint option
INFO: executing:
/usr/pgsql-12/bin/pg_basebackup -l "repmgr base backup" -D /production/pgsql/12/data -h IXXXXXXXXXXXXXXXXXXXXX -p 5432 -U repmgr -X stream
```

Astuce : utilisez **watch -n1 df -h** côté nœud standby pour regarder l'évolution du remplissage du disque et pour vérifier qu'entre le nœud principal et secondaire une même quantité de données est utilisée.

Une fois cette opération qui peut prendre du temps est terminée  
Relancer le service postgresql

```
Service postgresql12 start
```

Vérifier son état

```
Service postgresql12 status
```

Maintenant lancer cette commande sur le nœud qui est censé être le standby

```
repmgr standby register --force
```

```
bash-4.2$ ./repmgr standby register --force
INFO: connecting to local node "PRDPOSTGRESQL2.repli" (ID: 2)
INFO: connecting to primary database
INFO: standby registration complete
NOTICE: standby node "PRDPOSTGRESQL2.repli" (ID: 2) successfully registered
bash-4.2$
```

Ensuite vérifier que les rôles sont de nouveau assignés normalement

```
repmgr cluster show --verbose
```

```
bash-4.2$ ./repmgr cluster show --verbose
INFO: checking for package configuration file "/etc/repmgr/12/repmgr.conf"
INFO: configuration file found at: "/etc/repmgr/12/repmgr.conf"
INFO: connecting to database
```

ID	Name	Role	Status	Upstream	Location	Priority	Timeline	Connection string
1		primary	* running		default	100	19	host=
2		standby	running		default	100	19	host=

```
.repli user=repmgr dbname=repmgr connect_timeout=2
.repli user=repmgr dbname=repmgr connect_timeout=2
```

Astuce si vous avez dû fermer le service Cron pensez à le redémarrer

```
Service crond start
```

Vérifier

```
Service crond status
```

Et regarder si des jobs qui devaient se lancer n'ont pas été rater et dans le cas où si c'est le cas

demander au client si vous pouvez les lancer ou bien vous le signaler au client pour qu'il fasse ses vérifications lui-même



POSTGRES

# Commande Postgresql

# SQLPLUS

# Notes Général

POUR SQLITE 3

Les commandes deletes s'exécutent sans \* comme dans l'exemple suivant :

```
DELETE from users WHERE id IN (3, 9);
```