

POSTGRES

- [TimescaleDB Update](#)
- [Création user en lecture seul](#)
- [Modification du compte Monitor](#)
- [Vacuum](#)
- [Splitbrain](#)
- [Commande Postgresql](#)

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
-----+-----+-----+-----+-----
Updates:
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/N]: y
```

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
```

Nom	Version	Schéma	Liste des extensions installées 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`

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 POSTGRESQ11.repli -U repmgr -d repmgr standby clone -F
```

Ici POSTGRESQ11 est le node master

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

```
[root@... :/production/pgsql]
# sudo su postgres
bash-4.2$ pwd
/production/pgsql
bash-4.2$ /usr/pgsql-12/bin/repmgr -h I... -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=f... .repli user=... dbname=r...
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 I... -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

Commande Postgresql