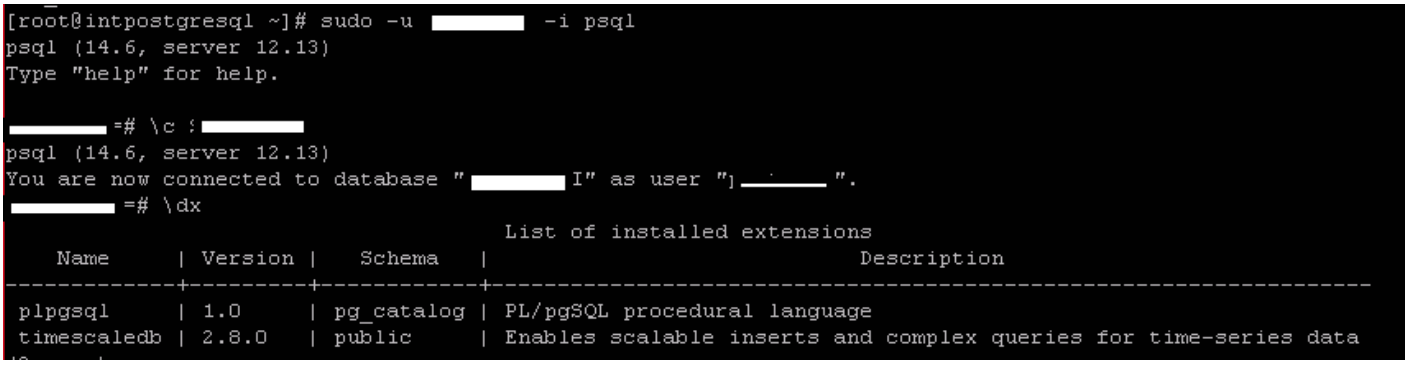
POSTGRES

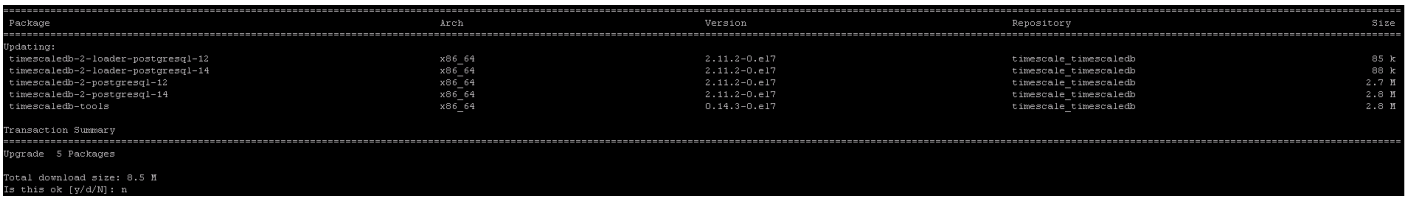
- TimescaleDB Update
- Création user en lecture seul
- Modification du compte Monitor
- Vacuum
- Splitbrain
- Commande Postgresql

TimescaleDB Update



Ajouter le proxy avec la commande export

Yum update



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;

```
[redacted]:=# \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

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
```

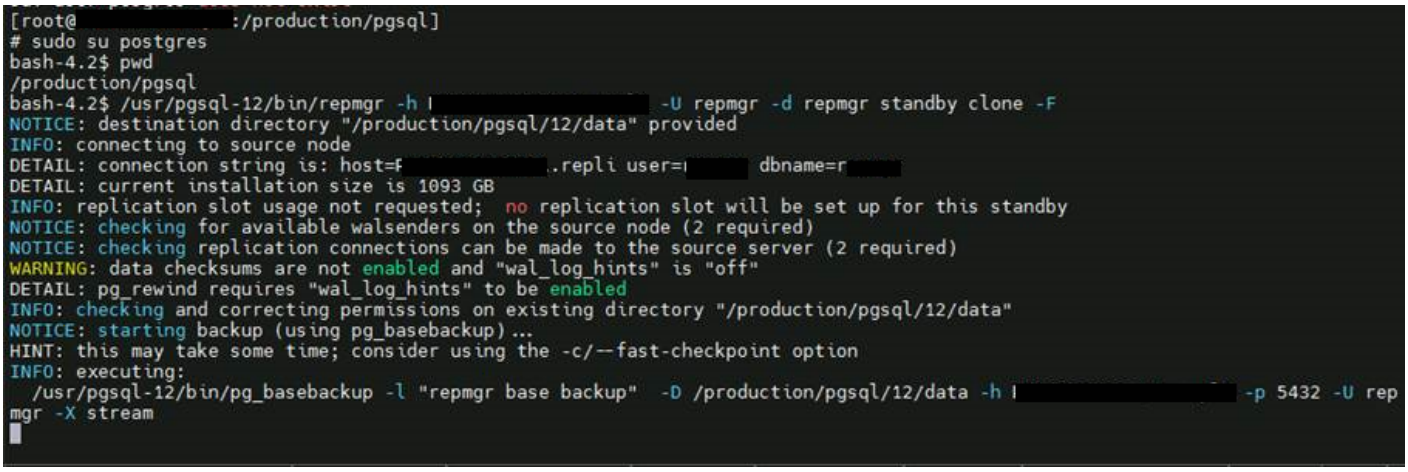
Dans le cas où il risque d'y avoir des cron de sauvegarde ou autre désactiver aussi le service crond

```
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



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 temp est terminée
Relancer le service postgresql

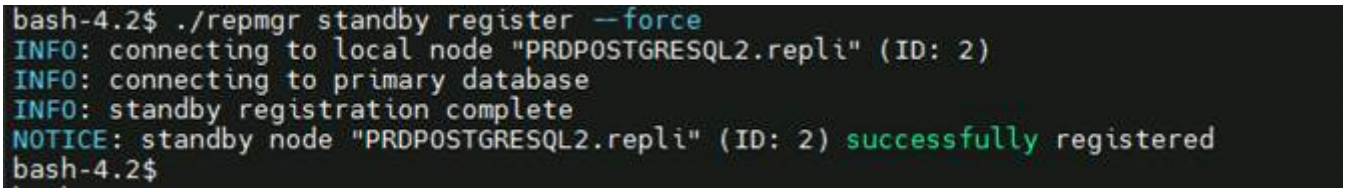
```
Service postgresql12 start
```

Vérifier son état

```
Service postgresql12 status
```

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

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



Ensuite vérifier que les rôles sont de nouveau assignés normalement

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



Astuce si vous avez dût 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