

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@xxxxxxxxxxxxx :/production/pgsql]
# sudo su postgres
bash-4.2$ pwd
/production/pgsql
bash-4.2$ /usr/pgsql-12/bin/repmgr -h [REDACTED] -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[REDACTED]..repli user=[REDACTED] dbname=r[REDACTED]
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 [REDACTED] -p 5432 -U rep
mgr -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
```

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

Revision #2

Created 31 October 2024 22:52:05 by Cavallone

Updated 25 November 2024 18:31:41 by Cavallone