RÉPLICATION LOGIQUE ET POSTGRESQL

CAPITOLE DU LIBRE 2024

Philippe VIEGAS

LOXODATA

QUI SUIS-JE ?

Philippe Viegas

- PostgreSQL depuis 2008
- Rejoint LOXODATA en 2022
- Consultant et formateur PostgreSQL

LOXODATA

Entreprise disposant de 3 piliers d'expertises

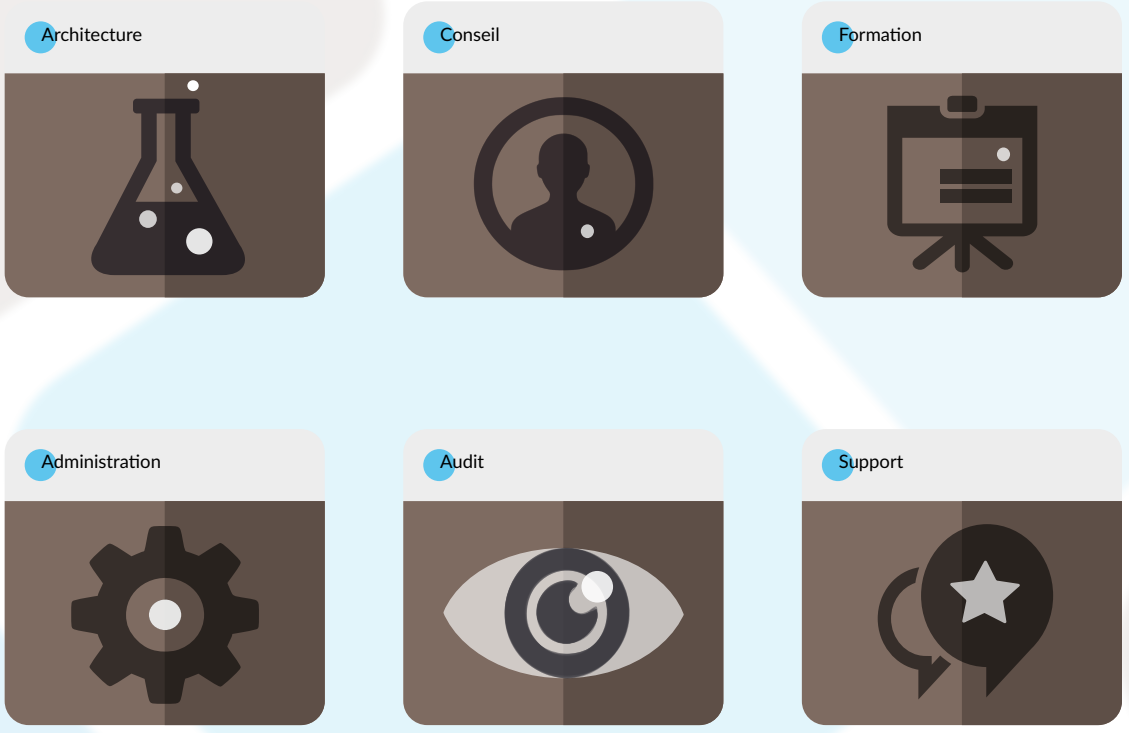
PostgreSQL

DevOps

Cloud

LOXODATA

Une large palette de services



POSTGRESQL

PRÉSENTATION PROJET

- Origine à l'université de Berkeley, initié par Michael Stonebraker
- Licence PostgreSQL, type BSD
- Projet communautaire porté par le PGDG

VERSIONS

- Versions majeurs maintenues pour cinq ans
- Versions mineurs tous les trimestres
- Versions actuelles : 17 à 13

HISTORIQUE

- PostgreSQL en 1996, avec support du SQL (version 6.0)
- Réplication (streaming) en version 9.0
- Décodage logique des WAL en version 9.4
- Réplication logique native en version 10

HISTORIQUE

- Réplication des tables partitionnées en version 13
- Ajout de la vue `pg_stat_replication_slots` en version 14
- Filtre sur colonne et lignes, et ajout de la vue `pg_stat_subscription_stats` en version 15
- Ajout du décodage logique depuis un serveur standby en version 16

FONCTIONNALITÉS

- MVCC
- ACID
- Respect standard SQL
- Moteur personnalisable
- Nombreuses API clientes
- Extensibilité
- Sécurité
- Feature matrix

RÉPLICATION

POURQUOI LA RÉPLICATION ?

- Partage d'informations
- Garantir sécurité et disponibilité des données
- Haute-disponibilité
- Scalabilité

DIFFÉRENTS TYPES DE RÉPLICATION

- Réplication physique
- Réplication logique
- Symétrie
- Synchronisme

RÉPLICATION PHYSIQUE

- Réplication d'une instance au niveau fichiers de données
- Recopie initiale des données
- Envoi des fichiers WAL
- Restauration en continu
- Asynchrone/Synchrone
- Plusieurs typologies :
  - Warm standby
  - Hot standby
  - Streaming

RÉPLICATION LOGIQUE

- Réplication des données au niveau objet (table)
- Granularité plus fine (transaction)
- Modèle publication/souscription
- Décodage logique des WAL
- Asynchrone/Synchrone
- Commandes SQL intégrées

COMPARAISON

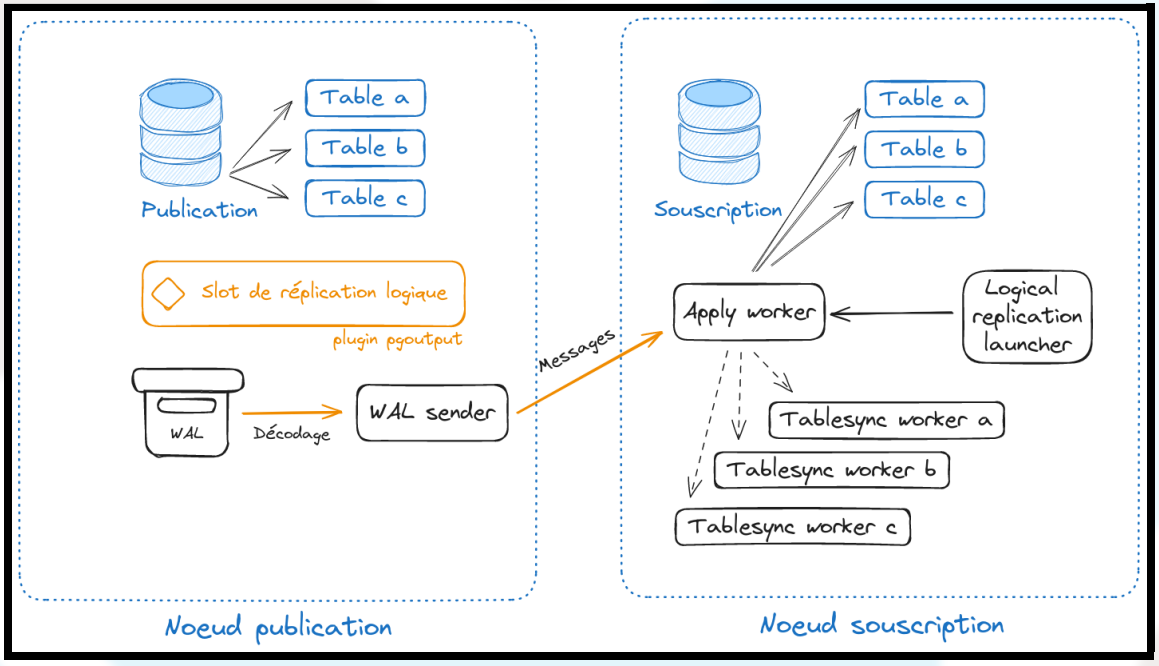
Type	Réplication physique	Réplication logique
Terminologie	Primaire/Standby	Publication/Souscription
Données échangées	Fichiers WAL	Messages de réplication
Synchro initiale	Restauration de sauvegarde	Automatique
Cible de restauration	Instance (cluster)	Base de données
Types d'opérations admises	Lecture seule	Lecture et écriture
Environnements	OS et versions majeures identiques	Hétérogènes

RÉPLICATION LOGIQUE

CAS D'USAGE

- Migration de versions majeures
- Réplication entre plateformes différentes (Windows, linux)
- Accès à un ensemble de données pour des groupes/utilisateurs différents
- Partager un sous-ensemble d'une base vers une ou plusieurs bases
- Suivi incrémental des changements sur une base ou sous-ensemble (CDC)
- Auditing

ARCHITECTURE



CONFIGURATION

- Configuration `postgresql.conf`
- Publication :

```
wal_level = logical
max_wal_senders = 10
max_replication_slots = 10
```

- Souscription :

```
max_replication_slots = 10
max_worker_processes = 8
max_logical_replication_workers = 4
max_sync_workers_per_subscription = 2
max_parallel_apply_workers_per_subscription
```

CONFIGURATION

- Rôle de réplication

```
CREATE ROLE repli LOGIN REPLICATION PASSWORD <redacted>;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO repli;
```

CONFIGURATION

- Configuration `pg_hba.conf`

```
# TYPE   DATABASE     USER     ADDRESS          METHOD
host     dwhpoker     repli    127.0.0.1/32     scram-sha-256
```

PUBLICATION

- Commande CREATE PUBLICATION

```
CREATE PUBLICATION pub_dwh_poker
FOR TABLES users, bonus, tournament
```

- ALTER PUBLICATION

```
ALTER PUBLICATION pub_dwh_poker
ADD TABLE users_filtered (user_id, firstname);
```

PUBLICATION

- Publication depuis le primaire à répliquer
- Publication d'une table, groupe de tables ou schéma :

```
FOR ALL TABLES, FOR TABLE, FOR TABLES IN SCHEMA
```

- DML :

```
INSERT, UPDATE, DELETE, TRUNCATE
```

PUBLICATION

- Publication de colonnes seulement

```
CREATE PUBLICATION users_filtered
FOR TABLE users (user_id, firstname);
```

- Publication de lignes seulement

```
CREATE PUBLICATION users_filtered
FOR TABLE users WHERE (enabled IS TRUE);
```

SOUSCRIPTION

- Commande CREATE SUBSCRIPTION

```
CREATE SUBSCRIPTION subscr_to_poker
CONNECTION 'host=dwh01 port=5433 user=repli dbname=dwhpoker'
PUBLICATION pub_dwh_poker
WITH (disable_on_error = true);
```

- ALTER SUBSCRIPTION

```
ALTER SUBSCRIPTION subscr_to_poker ENABLE;
ALTER SUBSCRIPTION subscr_to_poker REFRESH PUBLICATION;
```

SOUSCRIPTION

- Une souscription pour une ou plusieurs publication
- Cascade de la réplication possible
- Un slot de réplication par souscription

SUPERVISION - PUBLICATION

- pg_stat_replication

```
dwhpoker=# SELECT * FROM pg_stat_replication ;
-[ RECORD 1 ]----+------------------------------
pid              | 775889
usesysid         | 220977
usename          | repli
application_name | subscr_to_poker
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 43516
backend_start    | 2024-11-16 15:34:16.861115+01
backend_xmin     |
state            | streaming
sent_lsn         | 3B/86148F10
write_lsn        | 3B/86148F10
flush_lsn        | 3B/86148F10
replay_lsn       | 3B/86148F10
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-11-16 15:55:36.472911+01
```

SUPERVISION - PUBLICATION

- pg_replication_slots

```
dwhpoker=# SELECT * FROM pg_replication_slots;
-[ RECORD 1 ]-------+-----------------------------------
slot_name           | subscr_to_poker
plugin              | pgoutput
slot_type           | logical
datoid              | 16437
database            | dwhpoker
temporary           | f
active              | t
active_pid          | 775889
xmin                |
catalog_xmin        | 98706640
restart_lsn         | 3B/861DF9F8
confirmed_flush_lsn | 3B/861DFA30
wal_status          | reserved
safe_wal_size       |
two_phase           | f
```

SUPERVISION - PUBLICATION

- pg_stat_replication_slots

```
dwhpoker=# SELECT * FROM pg_stat_replication_slots;
-[ RECORD 1 ]+-------------------------------------
slot_name    | subscr_to_poker
spill_txns   | 0
spill_count  | 0
spill_bytes  | 0
stream_txns  | 0
stream_count | 0
stream_bytes | 0
total_txns   | 246
total_bytes  | 77381
stats_reset  |
```

SUPERVISION - SOUSCRIPTION

- pg_stat_subscription_stats
- pg_stat_subscription

```
dwh=# SELECT * FROM pg_stat_subscription ;
-[ RECORD 1 ]---------+------------------------------
subid                 | 51545
subname               | subscr_to_poker
pid                   | 1574896
relid                 |
received_lsn          | 3B/861E5D48
last_msg_send_time    | 2024-11-16 16:04:02.009662+01
last_msg_receipt_time | 2024-11-16 16:04:02.015951+01
latest_end_lsn        | 3B/861E5D48
latest_end_time       | 2024-11-16 16:04:02.009662+01
```

RESTRICTIONS

- Schémas et commandes DDL
- Séquences
- Tables et tables partitionnées seulement
- TRUNCATE et clés étrangères
- Large Objects
- Clés primaires ou `REPLICA IDENTITY` requis

NOUVEAUTÉS DE LA 17

- `pg_upgrade` et migration de slots
- `pg_createsubscriber`
- Failover des slots de réplication

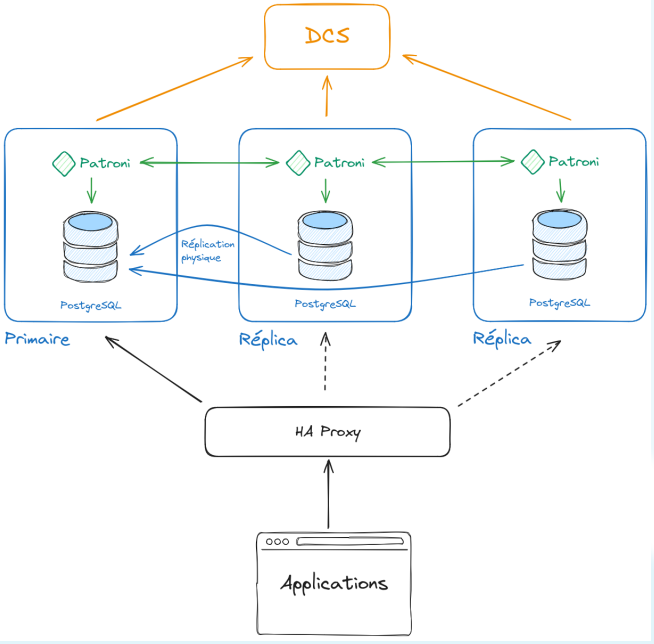
```
CREATE SUBSCRIPTION sub CONNECTION '...'
PUBLICATION pub WITH (failover = 'true');
```

HAUTE DISPONIBILITÉ

SLOTS DE RÉPLICATION ET BASCULE

- Slots de réplication créés sur le primaire
- Bascule d'un primaire bloque la réplication logique
- Pas de solution native jusqu'à la version 17
- Avant, utiliser pg_failover_slots ou sinon avec Patroni

PATRONI


DCS
Patroni
Patroni
Patroni
Réplication physique
PostgreSQL
PostgreSQL
PostgreSQL
Primaire
Réplica
Réplica
HA Proxy
Applications

PATRONI

- Framework de gestion de cluster PostgreSQL
- Haute disponibilité de service
- Configuration centralisée
- Bascule automatique

CONFIGURATION

- `wal_level` à logical (restart)
- Commande `patronictl`

```
postgres@pgdeb01:~$ patronictl -c /etc/patroni/config.yml edit-config
---
+++
@@ -4,6 +4,7 @@
   parameters:
     archive_command: pgbackrest --stanza=loxodemo archive-push %p
     archive_mode: 'on'
+    wal_level: 'logical'
   recovery_conf:
     restore_command: pgbackrest --stanza=loxodemo archive-get %f %p
   use_pg_rewind: false

Apply these changes? [y/N]: y
Configuration changed
```

CONFIGURATION

- Lister les noeuds

```
postgres@pgdeb01:~$ patronictl -c /etc/patroni/config.yml list
+ Cluster: loxodemo (7382147555638198668) ----+----+-----------+-----------------+-----------
| Member   | Host         | Role    | State     | TL | Lag in MB | Pending restart | Pending re
+----------+--------------+---------+-----------+----+-----------+-----------------+-----------
| pgdeb01  | 10.200.0.11  | Leader  | running   |  1 |           | *               | wal_level:
| pgdeb02  | 10.200.0.12  | Replica | streaming |  1 |         0 | *               | wal_level:
| pgdeb03  | 10.200.0.13  | Replica | streaming |  1 |         0 | *               | wal_level:
+----------+--------------+---------+-----------+----+-----------+-----------------+-----------
```

CONFIGURATION

- Redémarrer le cluster

```
postgres@pgdeb01:~$ patronictl -c /etc/patroni/config.yml restart --force loxodemo
+ Cluster: loxodemo (7382147555638198668) ----+----+-----------+-----------------+-----------
| Member   | Host         | Role    | State     | TL | Lag in MB | Pending restart | Pending re
+----------+--------------+---------+-----------+----+-----------+-----------------+-----------
| pgdeb01 | 10.200.0.11 | Leader  | running   |  1 |           | *               | wal_level:
| pgdeb02 | 10.200.0.12 | Replica | streaming |  1 |         0 | *               | wal_level:
| pgdeb03 | 10.200.0.13 | Replica | streaming |  1 |         0 | *               | wal_level:
+---------+-------------+---------+-----------+----+-----------+-----------------+-----------
Success: restart on member pgdeb01
Success: restart on member pgdeb02
Success: restart on member pgdeb03
```

CONFIGURATION

- Création des slots de réplication

```
postgres@pgdeb01:~$ patronictl -c /etc/patroni/config.yml edit-config
---
+++
@@ -8,6 +8,11 @@
   recovery_conf:
     restore_command: pgbackrest --stanza=loxodemo archive-get %f %p
   use_pg_rewind: false
-  use_slots: false
+  use_slots: true
 retry_timeout: 10
 ttl: 30
+slots:
+  logical_slot_emp:
+    database: employees
+    plugin: pgoutput
+    type: logical

Apply these changes? [y/N]: y
Configuration changed
```

CONFIGURATION

- use_slots
- slots
  - logical_slot_emp
  - database
  - plugin
  - type

SUPERVISION

- Un exemple avec pg_replication_slots

```
SELECT slot_name,
       active,
       confirmed_flush_lsn,
       pg_current_wal_lsn(),
       pg_size_pretty(pg_wal_lsn_diff(pg_current_wal_lsn(), restart_lsn)) AS retained_walsize,
       pg_size_pretty(pg_wal_lsn_diff(pg_current_wal_lsn(), confirmed_flush_lsn)) AS subscriber_
FROM pg_replication_slots;

    slot_name                | active | confirmed_flush_lsn | pg_current_wal_lsn | retained_wal
-----------------------------+--------+---------------------+--------------------
 logical_slot_emp            | t      | 0/72B96B50          | 0/72B96B50
```

CONCLUSION

EN RÉSUMÉ

- Réplication logique native depuis la version 10
- Améliorations constantes au gré des versions
- Manque quelques fonctionnalités
- Permet de s'intégrer dans diverses topologies

DES QUESTIONS ?

LOXODATA
p.viegas@loxodata.com