Administration PostgreSQL Server

MODULE 2 : MODES DE RÉCUPÉRATION ET STRATÉGIES DE BACKUP

Objectifs du Module

• Comprendre l'importance des modes de journalisation pour la récupération des données.

• Identifier les types de sauvegardes dans PostgreSQL.

• Apprendre à configurer une stratégie de sauvegarde adaptée.

Plan du Module

1. ARCHITECTURE DU JOURNAL WAL (WRITE-AHEAD LOGGING)

2. MODES DE RÉCUPÉRATION DISPONIBLES DANS POSTGRESQL

3. TYPES DE SAUVEGARDES DANS POSTGRESQL

4. STRATÉGIES DE SAUVEGARDE : CHOIX ET PLANIFICATION

Architecture du Journal WAL

- Le journal WAL consigne toutes les modifications avant qu'elles ne soient écrites sur disque.
- Objectif : garantir l'intégrité des données en cas de panne.
- Emplacement par défaut des fichiers WAL : pg_wal/.
- Exemple de paramètres dans postgresql.conf :

- wal_level = replica

- archive_mode = on

Modes de Récupération dans PostgreSQL

1. RÉCUPÉRATION STANDARD : RESTAURATION DEPUIS LE DERNIER CHECKPOINT.

2. RÉCUPÉRATION POINT-IN-TIME (PITR) : RESTAURATION À UN MOMENT PRÉCIS.

3. RÉCUPÉRATION VIA UN REPLICA : BASCULEMENT VERS UN SERVEUR RÉPLIQUÉ.

Types de Sauvegardes dans PostgreSQL

1. Sauvegarde logique : pg_dump, pg_dumpall (format SQL ou archive).

• Utilisation pour migrations ou exports partiels.

2. Sauvegarde physique : pg_basebackup (copie complète des fichiers binaires).

• Utilisation pour des sauvegardes complètes.

Exemple de commande pg_dump :

```
pg_dump -U postgres -d ma_base -F c -f sauvegarde.dump
```

Différences entre Sauvegarde Logique et Physique

• Sauvegarde logique (pg_dump, pg_dumpall) : SQL ou archive binaire, export partiel possible.

• Sauvegarde physique (pg_basebackup) : fichiers binaires complets.

• Sauvegarde physique plus rapide pour la restauration complète.

- Checkpoints : enregistrent l'état des pages modifiées sur disque.
- Segments WAL : stockent les modifications avant le disque.
- Commandes associées :
  - SELECT pg_current_wal_lsn();
  - pg_switch_wal();

Journalisation des Transactions

Planification d'une Stratégie de Sauvegarde

- Fréquence de mise à jour des données : volume des transactions.
- Temps de restauration acceptable (RTO) et perte de données acceptable (RPO).
- Recommandations :

- Système critique : sauvegarde physique + WAL archivés (quotidienne).

- Système non critique : pg_dump (hebdomadaire).

Configuration de l'Archivage des WAL

- postgresql.conf :
  - archive_mode = on
  - archive_command = 'cp %p /path/to/archive/%f'.
- Utilisation pour Point-In-Time Recovery (PITR).

Exemple de Processus de Sauvegarde avec PITR

1. Sauvegarde complète initiale : pg_basebackup -D /backup_dir -Ft -z -X fetch.

2. Archivage WAL activé : archive_mode = on.

3. Restauration à une date donnée : recovery_target_time.

4. Relance de PostgreSQL après restauration.

Bonnes Pratiques de Sauvegarde

• Automatiser les sauvegardes avec cron sous Linux.

• Stocker les sauvegardes sur un stockage distant.

• Tester régulièrement les sauvegardes et les restaurations.

• Activer la compression des sauvegardes.

Exercices Pratiques

1. Effectuer une sauvegarde logique avec pg_dump.

2. Restaurer une base à partir de la sauvegarde.

3. Configurer un archivage WAL et réaliser un PITR.

Conclusion et Questions

- Résumé des concepts clés : sauvegardes logiques, physiques, journal WAL.
- Importance de la stratégie de sauvegarde adaptée.
- Questions des participants.
- Aperçu du prochain module : Backup et Restauration avancés.