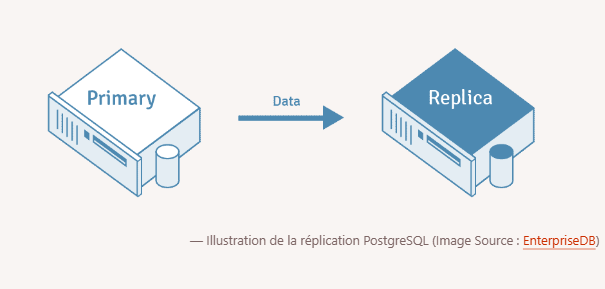
**Nom et prénoms : RANDRIANANDRASANA Lucas Eliade**

**N° d’inscription : 313I22**

**Réplication PostgreSQL**

**Qu’est-ce que la réplication PostgreSQL ?**



La réplication PostgreSQL est définie comme le processus de copie des données d’un serveur de base de données PostgreSQL vers un autre serveur. Le serveur de base de données source est également connu sous le nom de serveur « primaire », tandis que le serveur de base de données recevant les données copiées est connu sous le nom de serveur « réplique ».

La base de données PostgreSQL suit un modèle de réplication simple, où toutes les écritures vont vers un nœud primaire. Le nœud primaire peut ensuite appliquer ces modifications et les diffuser aux nœuds secondaires.

**Avantages de l’utilisation de la réplication PostgreSQL**

Voici quelques avantages clés de l’utilisation de la réplication PostgreSQL :

* Migration des données : Vous pouvez exploiter la réplication PostgreSQL pour la migration des données, soit par un changement de matériel du serveur de base de données, soit par le déploiement du système.
* Tolérance aux pannes : Si le serveur primaire tombe en panne, le serveur de secours peut faire office de serveur car les données contenues dans les serveurs primaire et de secours sont les mêmes.
* Performances du traitement transactionnel en ligne (OLTP) : Vous pouvez améliorer le temps de traitement des transactions et le temps de requête d’un système OLTP en supprimant la charge de requête de reporting. Le temps de traitement des transactions est la durée nécessaire à l’exécution d’une requête donnée avant qu’une transaction ne soit terminée.
* Test du système en parallèle : Lors de la mise à niveau d’un nouveau système, vous devez vous assurer que le système se comporte bien avec les données existantes, d’où la nécessité de tester avec une copie de la base de données de production avant le déploiement.

**Comment fonctionne la réplication PostgreSQL**

En général, les gens pensent que quand on s’amuse avec une architecture primaire et secondaire, il n’y a qu’une seule façon de configurer les sauvegardes et la réplication. Les déploiements PostgreSQL, cependant, peuvent suivre n’importe laquelle de ces trois méthodes :

* Réplication en continu : Réplique les données depuis le nœud primaire vers le secondaire, puis copie les données sur Amazon S3 ou Azure Blob pour le stockage de sauvegarde.
* Réplication au niveau du volume : Réplique les données au niveau de la couche de stockage, depuis le nœud primaire vers le nœud secondaire, puis copie les données vers Amazon S3 ou Azure Blob pour le stockage de sauvegarde.
* Sauvegardes incrémentielles : Réplique les données depuis le nœud primaire tout en construisant un nouveau nœud secondaire à partir du stockage Amazon S3 ou Azure Blob, ce qui permet le streaming directement depuis le nœud primaire.

**Les techniques de réplication sur PostgreSQL**

1. **Réplication en flux :**
   * Description : Permet de maintenir des copies synchronisées de la base de données en temps réel en envoyant les journaux de transactions (WAL) aux serveurs de secours (standby).
   * Avantages : Faible latence, haute disponibilité.
   * Inconvénients :
     + Latence : Même si la latence est faible, elle n'est pas nulle et peut affecter les performances des systèmes nécessitant une synchronisation en temps réel.
     + Complexité : La configuration et la gestion peuvent être complexes, surtout pour les déploiements de grande envergure.
     + Consommation de ressources : Peut consommer beaucoup de ressources réseau et CPU, surtout avec de nombreux serveurs standby.

Utilisation : Appropriée pour des systèmes nécessitant une haute disponibilité et une récupération rapide après sinistre.

1. **Réplication logique :**
   * Description : Réplication des changements de niveau logique (tables, lignes, colonnes) en utilisant des publications et abonnements.
   * Avantages : Permet une réplication sélective et une flexibilité pour la migration et la synchronisation de bases de données.

* Inconvénients :
* Compatibilité : Ne supporte pas toutes les opérations (par exemple, certaines DDL ne sont pas entièrement répliquées).
* Performances : Peut-être plus lente que la réplication physique pour des volumes de données très importants.
* Configuration : Plus complexe à configurer que la réplication en flux, nécessitant une gestion fine des publications et abonnements.
  + Utilisation : Idéale pour la migration de données, la réplication de sous-ensembles de données, et les mises à jour entre versions différentes de PostgreSQL.

1. **Réplication physique :**
   * Description : Crée une copie exacte de la base de données principale en utilisant les fichiers de données physiques.
   * Avantages : Très rapide pour les restaurations et les sauvegardes.

* Inconvénients :
* Flexibilité : Moins flexible que la réplication logique, car elle ne permet pas de répliquer seulement des sous-ensembles de données.
* Espace disque : Nécessite plus d'espace disque car elle crée une copie exacte de la base de données principale.
* Downtime : Toute opération de restauration à partir d'une sauvegarde physique peut nécessiter un certain temps d'indisponibilité.
  + Utilisation : Souvent utilisée pour les sauvegardes de secours et la mise en place initiale d'un système de réplication en flux.

1. **Réplication en mode cascade :**
   * Description : Les serveurs standby peuvent eux-mêmes servir de source de réplication pour d'autres serveurs standby.
   * Avantages : Réduit la charge sur le serveur principal et permet une scalabilité horizontale.

* Inconvénients :
* Complexité accrue : La gestion et le dépannage peuvent être plus complexes en raison de la chaîne de dépendances entre les serveurs.
* Latence cumulative : La latence peut s'accumuler à chaque niveau de réplication, affectant les performances des serveurs secondaires éloignés.
  + Utilisation : Utile dans des architectures distribuées et de grande envergure.

1. **Système de haute disponibilité :**
   * Description : Combinaison de réplication en flux, gestion automatique des basculements (failover) et équilibrage de charge (load balancing).
   * Avantages : Offre une disponibilité continue et une tolérance aux pannes.

* Inconvénients :
* Coût : La mise en œuvre et la maintenance d'un système HA sont souvent coûteuses en termes de matériel, de logiciels et de ressources humaines.
* Complexité : La configuration, la gestion des basculements automatiques et l'équilibrage de charge ajoutent une couche de complexité à l'administration de la base de données.
* Risques de split-brain : Des situations de partitionnement réseau peuvent mener à des scénarios de "split-brain", où deux nœuds tentent de fonctionner comme principaux simultanément.
  + Utilisation : Essentiel pour les systèmes critiques nécessitant une disponibilité permanente.

1. **Réplication synchrone et asynchrone :**
   * Description :
     + Synchrone : Assure que les transactions sont répliquées sur les serveurs de secours avant d'être confirmées.
     + Asynchrone : Les transactions sont confirmées sur le serveur principal avant d'être répliquées.
   * Avantages : La réplication synchrone garantit une haute sécurité des données, tandis que l'asynchrone offre de meilleures performances.
   * Inconvénients :

* Réplication synchrone :
* Performances : Peut significativement ralentir les transactions car elles doivent être confirmées sur les serveurs standby avant d'être complétées.
* Dépendance réseau : La performance et la fiabilité dépendent fortement de la qualité et de la latence du réseau entre les serveurs.
* Réplication asynchrone :
* Risques de perte de données : En cas de panne du serveur principal avant que les transactions soient répliquées, il peut y avoir une perte de données.
* Consistance éventuelle : Les serveurs standby peuvent ne pas être à jour en temps réel, ce qui peut poser des problèmes pour certaines applications nécessitant une cohérence stricte.
  + Utilisation : La réplication synchrone est utilisée pour des scénarios nécessitant une garantie de non-perte de données, et l'asynchrone est préférée pour des systèmes nécessitant une latence réduite.