

Réplication et tolérance aux pannes avec MongoDB

Nom/Prénom : GNOUG Omayma

Date : 02/12/2025

Sujet : Réplication et tolérance aux pannes avec MongoDB.

Partie 1 : Compréhension de base

1. Sur quels ports les trois serveurs du TP ont-ils été démarrés ? Un serveur sur le port 27018, un autre sur le port 27019 et le dernier sur le port 27020.
2. Quel serveur était le "Maître" (Primary) au début de la manipulation ? Le serveur écoutant sur le port 27018.
3. Par défaut, sur quel nœud s'effectuent les lectures et les écritures dans MongoDB ? Elles s'effectuent uniquement sur le nœud Primary (ici le 27018) pour assurer une cohérence forte.
4. Peut-on écrire des données directement sur un nœud Secondary (ex: 27019) ? Non, c'est impossible. Si on essaie (**insert**), le serveur renvoie une erreur indiquant qu'il n'est pas le maître.
5. Pourquoi la réplication est-elle qualifiée d'asynchrone dans ce TP ? Le serveur écrit dans son journal et envoie l'acquittement au client avant même que la donnée ne soit totalement répliquée sur les secondaires.
6. Quel est le risque de lire sur un nœud secondaire ? Récupérer une donnée incohérente (pas à jour) si la réplication n'a pas encore eu lieu entre le moment de l'écriture sur le primaire et la lecture sur le secondaire.
7. Quelle est la particularité d'un nœud Arbitre par rapport aux autres ? Il participe aux élections et à la grappe de serveurs, mais il ne sauvegarde pas les données (pas de stockage de la collection **person** par exemple).

Partie 2 : Commandes & configuration

8. Quelle commande a été utilisée pour recréer/initialiser le cluster ? `rs.initiate()` avec la configuration des membres (localhost:27018, etc.).
9. Quelle commande faut-il taper sur un nœud Secondary pour autoriser la lecture ? `rs.secondaryOk()` (ou `rs.secondary()` selon la version/dépréciation).
10. Quelle commande permet de vérifier si je suis connecté à un nœud Maître ou Esclave ? `db.isMaster()` ou simplement regarder le prompt du shell (qui indique souvent PRIMARY ou SECONDARY), ou tenter une commande comme `show collections` qui renvoie une erreur si on est sur un esclave non configuré.
11. Comment a-t-on créé la collection de test dans la base `demo1` ? `db.createCollection("person")`.
12. Quelle commande permet d'insérer un document JSON dans la collection ? `db.person.insert({ ... })`.
13. Comment a-t-on préparé l'ajout de l'arbitre au niveau système de fichiers ? En créant un répertoire spécifique via `mkdir arbitre1` pour séparer ses données (ou son absence de données) des autres nœuds.

Partie 3 : Résilience et tolérance aux pannes

14. Comment la panne du serveur Primary (27018) a-t-elle été simulée ? En arrêtant brutalement le processus avec `Ctrl + C` dans la fenêtre du terminal correspondant.
15. Que s'est-il passé immédiatement après l'arrêt du serveur 27018 ? Les deux autres nœuds (27019 et 27020) ont détecté la panne et ont lancé une élection ("campagne électorale") pour élire un nouveau maître.
16. Quel serveur a été élu nouveau Primary après la panne ? Le serveur écoutant sur le port 27019.
17. Est-il possible de se connecter au serveur 27018 juste après l'avoir arrêté ? Non, la connexion échoue ("Connection refused") car le processus n'est plus en cours d'exécution.
18. Si le serveur 27018 redémarre et rejoint le cluster, quel sera son rôle ? Il ne sera plus Primary (car le 27019 a pris la place), il deviendra un Secondary.

Partie 4 — Scénarios pratiques

19. Vous insérez "Dupont" sur le 27018. Vous vous connectez immédiatement sur le 27019. La donnée est-elle visible ? Oui, elle est visible (via `db.person.find()`), mais seulement après avoir activé la lecture sur le secondaire (`rs.secondaryOk()`), prouvant que la réplication fonctionne.

20. Vous essayez de faire `show collections` sur le 27019 sans configuration préalable. Que se passe-t-il ? Le serveur retourne une erreur (message "not master") et n'affiche rien.

21. Un client insère une donnée sur le 27019 (qui est devenu Primary après la panne). Est-ce que le 27020 la verra ? Oui, car le 27020 est un secondaire du même Replica Set et il répliquera la donnée depuis le nouveau maître (27019).

22. Pourquoi définit-on un répertoire spécifique lors du lancement du nœud arbitre (ex: `arbitre1`) ? Même si l'arbitre ne stocke pas de données applicatives, il a besoin d'un répertoire pour stocker sa configuration et ses logs de fonctionnement.

23. Dans le TP, l'arbitre a été configuré sur quel port ? Sur le port 27021 (indiqué comme `021` dans la transcription).

24. Si je veux une cohérence forte absolue sans risque de lire une vieille donnée, que dois-je faire ? Ne jamais forcer la lecture sur les secondaires (`rs.secondaryOk()`) et lire uniquement sur le Primary (comportement par défaut).