

## **Bases de données Avancées**

### **Atelier 5 : Réplication dans MongoDB**

#### **Tolérance aux pannes**

Nous voulons ici tester le système de tolérance aux pannes de MongoDB. Pour cela, nous allons lancer des serveurs avec des ReplicaSets.

#### **Installation d'un replicaSet**

1. Le but est de créer un replicaSet composé de 5 serveurs, dont 1 primaire, 3 secondaires et un arbitre. À chaque serveur, un répertoire de données doit être créé :
  - /data/rs1
  - /data/rs2
  - /data/rs3
  - /data/rs4
  - /data/arb
2. Choisir un nom de réplication : RS1
3. Le premier serveur sera lancé sur le port 27017  
`mongod -replSet RS1 -dbpath /data/db -port 27017`
4. Ouvrir 3 nouvelles consoles, dans chaque console, lancez un répliquat avec un nouveau port de connexion : 27018, 27019, 27020
  - `mongod -replSet RS1 -dbpath /data/rs1 -port 27018`
  - `mongod -replSet RS1 -dbpath /data/rs2 -port 27019`
  - `mongod -replSet RS1 -dbpath /data/rs3 -port 27020`
5. Ouvrir une nouvelle console pour se connecter au serveur qui sera choisi comme serveur primaire. Utilisez le port 27017:  
`mongo -port 27017`
6. Lancer le mode de réplication :  
`rs.initiate();` Dans la console “mongo” apparaîtra RS1:PRIMARY> désignant le serveur primaire.
7. Rajouter chaque répliquat :
  - `rs.add("<host>:27018");`
  - `rs.add("<host>:27019");`
  - `rs.add("<host>:27020");`
8. Vérifier la configuration de l'ensemble des serveurs :  
`rs.conf();`
9. Lorsqu'il y a plusieurs répliquats, il est nécessaire de définir un arbitre dans le cas où le serveur primaire tombe. Ouvrir une nouvelle console sur le port 30000  
`mongod -replSet RS1 -dbpath /data/arb -port 30000`
10. Ajouter l'arbitre dans la console mongo RS1:PRIMARY>  
`rs.addArb("<host>:30000");`

### Test de réplication

1. Dans la console RS1:PRIMARY> vérifier le contenu d'une base de données.  
`use DBLP;db.publis.count();`
2. Ouvrir une nouvelle console sur un serveur répliquat  
`mongo -port 27018`

Dans la console mongo apparaîtra RS1:SECONDARY> désignant le serveur secondaire (répliquat).

3. Exécutez La commande : `use DBLP;db.publis.count();` Remarque ? Comment résoudre le problème ?  
retourne une erreur car l'utilisation d'un répliquat n'a pas été autorisé. Pour ce faire, taper la commande : `rs.slaveOk();`  
Refaire le test
4. Dans la console RS1:PRIMARY>, ajouter un document "{ test : 1}"
5. Dans la console RS1:SECONDARY>, vérifier son existence.  
`use DBLP;db.publis.find({test:1});`  
Cela peut prendre quelques secondes (souvent quelques milli-secondes)

### Test de tolérance aux pannes

1. Dans la console RS1:PRIMARY>, ajouter un document "{ test : 2}" à la collectionDBLP.publis
1. Tuer le processus mongod "primaire" (console ouvrant mongod avec le port 27017) avec "ctrl+C", Ou dans la console RS1:PRIMARY>, taper : `use admin; puis db.shutdownServer();`
2. Regarder le comportement des autres mongod, en particulier l'arbitre (port 30000).  
Sera alors désigné le serveur qui deviendra le primaire. Il est aussi possible de connaître le serveur primaire dans la console RS1:SECONDARY> : `rs.status();`
3. Dans la console RS1:PRIMARY>, taper 'exit'. Relancer "mongo" mais avec le port du nouveau serveur primaire
4. Vérifier l'existence de votre document dans la collectionDBLP.publis.  
`use DBLP;db.publis.find({test:2});`