

Partitionnement de la Bdd (en local)

// lancement des serveurs de shards

// replica set pour le shard 1

```
mongod --shardsvr --replSet "rs0" --bind_ip 127.0.0.1 --dbpath <path-to>/db1  
--port 27021
```

```
mongod --shardsvr --replSet "rs0" --bind_ip 127.0.0.1 --dbpath <path-to>/dbrep11  
--port 27022
```

```
mongod --shardsvr --replSet "rs0" --bind_ip 127.0.0.1 --dbpath <path-to>/dbrep12  
--port 27023
```

// shard 2

```
mongod --shardsvr --bind_ip 127.0.0.1 --dbpath <path-to>/db2 --port 27031
```

// lancement d'un arbitre pour le shard1

```
mongod --dbpath /data/arb --replSet "rs0" --bind_ip 127.0.0.1 --port 27025
```

// initialisation du replica set pour le shard 1, sur le master du shard1 :

```
mongo --host 127.0.0.1 --port 27021  
rs.initiate(  
  {  
    _id: "rs0",  
    configsvr: false,  
    members: [  
      { _id : 0, host : "127.0.0.1:27021" },  
      { _id : 1, host : "127.0.0.1:27022" },  
      { _id : 2, host : "127.0.0.1:27023" }  
    ]  
  }  
)
```

// initialisation de l'arbitre pour le shard 1, sur le master du shard1 :

```
rs.addArb("127.0.0.1:27025")
```

```
rs.conf()
```

```
rs.status()
```

// création du serveur de config ; en prod, un replica set d'au moins 3 est recommandé

```
mongod --configsvr --replSet "rscfg" --bind_ip 127.0.0.1 --dbpath <path-  
to>/dbsvr --port 27040
```

// initialisation du replica set serveur (ici, on ne réplique pas...)

```
mongo --host 127.0.0.1 --port 27040  
rs.initiate(  
  {  
    _id: "rscfg",  
    configsvr: true,  
    members: [  
      { _id : 0, host : "127.0.0.1:27040" }  
    ]  
  }  
)
```



```

// création du serveur d'appli; en prod, au moins 2 sont recommandés
mongos --bind_ip 127.0.0.1 --port 27020 --configdb rscfg/127.0.0.1:27040

// déclaration des shards sur le serveur d'appli
// lancement de mongo sur le serveur
mongo --host 127.0.0.1 --port 27020 // si pas défini au-dessus, 27017/default
// verif status
sh.status();
// ajout des shards
sh.addShard("rs0/127.0.0.1:27021");
sh.addShard("127.0.0.1:27031");

// autorisation de découpage de la bdd
sh.enableSharding("<bdd>");
// critère de découpe par collection
sh.shardCollection("<bdd>.<collec1>", {"_id":"hashed"});
sh.shardCollection("<bdd>.<collec2>", {"_id":"hashed"});
...

// verif status
sh.status();

// ajout index pour requêtes géo-référencées sur coords géographiques
db.<collec1>.ensureIndex({<champ coord>:"2dsphere"});
db.<collec2>.ensureIndex({<champ coord>:"2dsphere"});
...

// si un nœud reste, pour le forcer à passer en primaire
mongo --host 127.0.0.1 --port <celui qui vit>
> rs.status()
> cfg = rs.conf()
> cfg.members = [cfg.members[<num de celui qui vit>]]
> rs.reconfig(cfg, {force : true})

```