

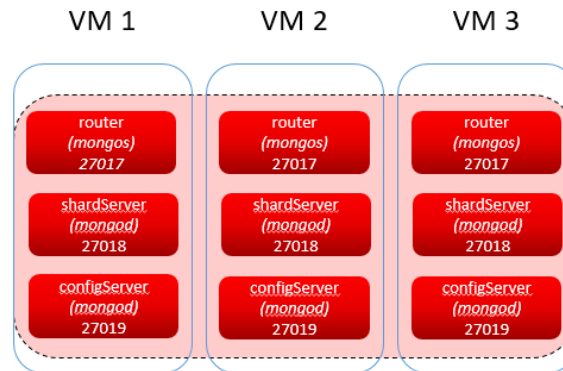
Créer une architecture de sharding dans MongoDB

Table des matières

Avant de commencer	2
1. Installation de MongoDB.....	3
1.1. Création des Dossiers	3
1.2. Configuration des hosts.....	3
1.3. Création d'une clé d'authentification.....	3
2. Création des ConfigServers.....	4
2.1. Création du fichier de configuration des ConfigServers.....	4
2.2. Création du service systemd pour lancer les ConfigServers.....	4
2.3. Mise en place de la réplication sur les ConfigServers	5
3. Création des ShardServers.....	6
3.1. Création du fichier de configuration des ShardServers.....	6
3.2. Création du service systemd pour lancer les ShardServers.....	6
3.3. Mise en place de la réplication sur les shardServers.....	7
4. Mise en place du sharding.....	8
4.1. Création des utilisateurs.....	8
4.2. Mise en place de la clé secrète.....	8
4.3. Vérifier le bon fonctionnement de la clé.....	8
5. Création des routers.....	9
5.1. Création du fichier de configuration des routers	9
5.2. Création du service systemd pour lancer les routers	9
5.3. Création d'un shard	10
6. Exemple de mise en place du sharding sur une base de données	11
Conclusion	11

Avant de commencer

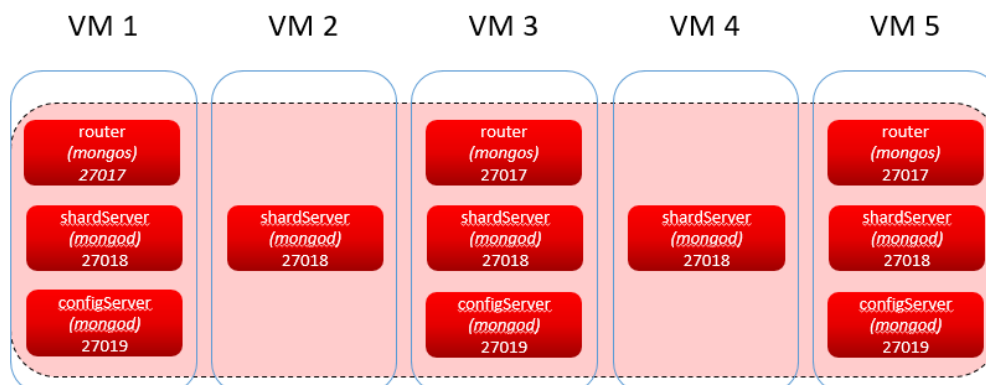
Dans cette documentation, je vais montrer comment monter un cluster mongodb de 3 nœuds selon l'architecture ci-dessous :



Dans cette documentation, j'adopterai la notation suivante pour désigner les vm : (à adapter à vos cas d'usage)

- VM1 : mongodb-1
- VM2 : mongodb-2
- VM3 : mongodb-3

L'exemple développé ci-dessous est un exemple d'architecture de sharding à 3 nœuds. On retrouve donc sur chaque instance toutes les briques de mongodb. Une architecture à 5 nœuds pourrait être la suivante (en règle général, toujours choisir un nombre impair de nœud)



1. Installation de MongoDB

Sur chaque nœud :

- Créer un groupe mongodb : `sudo groupadd mongodb`
- Créer un utilisateur mongodb : `sudo useradd --create-home --home /home/mongodb --shell /bin/bash -g gid mongodb mongodb`
- Télécharger la version .tar.gz de MongoDB ici : <https://www.mongodb.com/try/download/community>
- Installer mongodb dans le dossier des applications : `sudo -u mongodb tar xzvf mongodb-shell-linux-x86_64-debian10-4.4.6.tgz -C /apps`

1.1. Création des Dossiers

Sur chaque nœud du cluster :

- Créer les dossiers suivants :
 - o Pour la configuration : `sudo -u mongodb mkdir /home/mongodb/conf`
 - o Pour les bases de données : `sudo -u mongodb mkdir /home/mongodb/data`
 - o Pour le configServer : `sudo -u mongodb mkdir /home/mongodb/data/configsvr`
 - o Pour le shardServer : `sudo -u mongodb mkdir /home/mongodb/data/shardsvr`
 - o Pour les logs : `sudo -u mongodb mkdir /home/mongodb/logs`
 - o Pour les process ids : `sudo -u mongodb mkdir /home/mongodb/run`
 - o Pour les clés ssl : `sudo -u mongodb mkdir /home/mongodb/ssl`
- Créer les liens symboliques dans /apps/mongodb :
 - o Vers la base de données : `sudo ln -s /home/mongodb/data/ /apps/mongodb/data`
 - o Vers les logs : `sudo ln -s /home/mongodb/logs/ /apps/mongodb/logs`
 - o Vers la configuration : `sudo ln -s /home/mongodb/conf/ /apps/mongodb/conf`

1.2. Configuration des hosts

Sur chaque nœud du cluster :

- o Editer le fichier /etc/hosts
- o Ajouter et/ou modifier les lignes suivantes :

```
10.10.10.xxx    mongodb-1
10.10.10.yyy    mongodb-2
10.10.10.zzz    mongodb-3
```
- o Sauvegarder le fichier

1.3. Création d'une clé d'authentification

Sur la VM mongodb-1:

- `openssl rand -base64 756 > /tmp/mongodb.key`
- `sudo chown mongodb:mongodb /tmp/mongodb.key`
- `sudo chmod 400 /tmp/mongodb.key`
- `sudo mv /tmp/mongodb.key /home/mongodb/ssl/mongodb.key`
- `sudo -u mongodb scp /home/mongodb/ssl/mongodb.key mongodb@mongodb-2:/home/mongodb/ssl/mongodb.key`
- `sudo -u mongodb scp /home/mongodb/ssl/mongodb.key mongodb@mongodb-3:/home/mongodb/ssl/mongodb.key`

2. Création des ConfigServers

2.1. Création du fichier de configuration des ConfigServers

Sur chaque nœud :

- Editer le fichier conf du configServer : `sudo -u mongodb nano /home/mongodb/conf/configsvr.conf`

```
net:
  port: 27019
  bindIp: 127.0.0.1,mongodb-x

processManagement:
  fork: true
  pidFilePath: /home/mongodb/run/mongodb-configsvr.pid
  timeZoneInfo: /usr/share/zoneinfo

replication:
  replSetName: configReplSet

sharding:
  clusterRole: configsvr

#security:
  #keyFile: /home/mongodb/ssl/mongodb.key

storage:
  dbPath: /home/mongodb/data/configsvr
  journal:
    enabled: true

systemLog:
  destination: file
  logAppend: true
  path: /home/mongodb/logs/configsvr.log
```

2.2. Création du service systemd pour lancer les ConfigServers

- Editer le fichier de service du configsvr : `sudo nano /etc/systemd/system/mongodb-configsvr.service`

```
[Unit]
Description=MongoDB ConfigServer
Documentation=https://docs.mongodb.org/manual
After=network.target

[Service]
User=mongodb
Group=mongodb
ExecStart=/apps/mongodb/bin/mongod --config=/home/mongodb/conf/configsvr.conf
ExecStartPre=/usr/bin/chown -R mongodb:mongodb /apps/mongodb
ExecStartPre=/usr/bin/chown -R mongodb:mongodb /home/mongodb
ExecStopPost=/usr/bin/rm -f /home/mongodb/run/mongodb-27019.sock
ExecStopPost=/usr/bin/rm -f /home/mongodb/run/mongodb-configsvr.pid
PermissionsStartOnly=true
Type=forking
# file size
LimitFSIZE=infinity
# cpu time
LimitCPU=infinity
# virtual memory size
```

```
LimitAS=infinity
# open files
LimitNOFILE=64000
# processes/threads
LimitNPROC=64000
# locked memory
LimitMEMLOCK=infinity
# total threads (user+kernel)
TasksMax=infinity
TasksAccounting=false
# Recommended limits for mongod as specified in
# https://docs.mongodb.com/manual/reference/ulimit/#recommended-ulimit-settings

[Install]
WantedBy=multi-user.target
```

- Lancer le service sur chaque nœud :
 - o `sudo systemctl enable mongod-configsvr.service`
 - o `sudo systemctl start mongod-configsvr.service`

2.3. Mise en place de la réplication sur les ConfigServers

Sur un seul nœud (ex : mongod-1) :

- Se connecter à un configServer :
 - o `sudo -u mongod /apps/mongod/bin/mongo --host=mongod-1 --port=27019`
- Initialiser la réplication ainsi :

```
rs.initiate(
  {
    _id: "configReplSet",
    configsvr: true,
    members: [
      { _id : 0, host : "mongod-1:27019", priority : 3 },
      { _id : 1, host : "mongod-2:27019", priority : 1 },
      { _id : 2, host : "mongod-3:27019", priority : 2 }
    ]
  }
)
```

- Vérifier la bonne mise en place de la réplication ainsi :
 - o `rs.status();`

3. Création des ShardServers

3.1. Création du fichier de configuration des ShardServers

Sur chaque nœud :

- Editer le fichier conf du shardServer : `sudo -u mongodb nano /home/mongodb/conf/shardsvr.conf`

```
net:
  port: 27018
  bindIp: mongodb-x

processManagement:
  fork: true
  pidFilePath: /home/mongodb/run/mongodb-shardsvr.pid
  timeZoneInfo: /usr/share/zoneinfo

replication:
  replSetName: shardReplSet

#security:
  #keyFile: /home/mongodb/ssl/mongodb.key

sharding:
  clusterRole: shardsvr

storage:
  dbPath: /home/mongodb/data/shardsvr
  journal:
    enabled: true

systemLog:
  destination: file
  logAppend: true
  path: /home/mongodb/logs/shardsvr.log
```

3.2. Création du service systemd pour lancer les ShardServers

- Editer le fichier de service du shardsvr : `sudo nano /etc/systemd/system/mongodb-shardsrv.service`

```
[Unit]
Description=MongoDB ShardServer
Documentation=https://docs.mongodb.org/manual
After=network.target

[Service]
User=mongodb
Group=mongodb
ExecStart=/apps/mongodb/bin/mongod --config=/home/mongodb/conf/shardsvr.conf
ExecStartPre=/usr/bin/chown -R mongodb:mongodb /apps/mongodb
ExecStartPre=/usr/bin/chown -R mongodb:mongodb /home/mongodb
ExecStopPost=/usr/bin/rm -f /home/mongodb/run/mongodb-27018.sock
ExecStopPost=/usr/bin/rm -f /home/mongodb/run/mongodb-shardsvr.pid
PermissionsStartOnly=true
Type=forking
# file size
LimitFSIZE=infinity
# cpu time
LimitCPU=infinity
# virtual memory size
LimitAS=infinity
```

```
# open files
LimitNOFILE=64000
# processes/threads
LimitNPROC=64000
# locked memory
LimitMEMLOCK=infinity
# total threads (user+kernel)
TasksMax=infinity
TasksAccounting=false
# Recommended limits for mongod as specified in
# https://docs.mongodb.com/manual/reference/ulimit/#recommended-ulimit-settings

[Install]
WantedBy=multi-user.target
```

- Lancer le service sur chaque nœud :
 - o `sudo systemctl enable mongodb-shardsvr.service`
 - o `sudo systemctl start mongodb-shardsvr.service`

3.3. Mise en place de la réplication sur les shardServers

Sur un seul nœud (ex : mongodb-1) :

- Se connecter à un shardServer :
 - o `sudo -u mongodb /apps/mongodb/bin/mongo --host=mongodb-1 --port=27018`
- Initialiser la réplication ainsi :

```
rs.initiate(
  {
    _id: "shardReplSet",
    members: [
      { _id : 0, host : "mongodb-0:27018", priority : 3 },
      { _id : 1, host : "mongodb-1:27018", priority : 1 },
      { _id : 2, host : "mongodb-2:27018", priority : 2 }
    ]
  }
)
```

- Vérifier la bonne mise en place de la réplication ainsi :
 - o `rs.status();`

4. Mise en place du sharding

4.1. Création des utilisateurs

Note : La création des utilisateurs s'effectue pour une base associée. Dans notre cas ici, ce sera la database admin (créée par défaut).

- **Se connecter au shardServer primaire** : `sudo -u mongodb /apps/mongodb/bin/mongo --host=mongodb-1 --port=27018`
 - o `use admin;`
 - o `db.createUser({user: "root", pwd: "root",roles: [{ role: "root", db: "admin" }]})`
 - o `db.createUser({user: "admin", pwd: "admin",roles: [{ role: "userAdminAnyDatabase", db: "admin" }]})`

- **Se connecter au configServer primaire** : `sudo -u mongodb /apps/mongodb/bin/mongo --host=mongodb-1 --port=27019`
 - o `use admin;`
 - o `db.createUser({user: "root", pwd: "root",roles: [{ role: "root", db: "admin" }]})`
 - o `db.createUser({user: "admin", pwd: "admin",roles: [{ role: "userAdminAnyDatabase", db: "admin" }]})`

4.2. Mise en place de la clé secrète

- Dé-commenter la partie **security** dans tous les fichiers `configsvr.conf` et `shardsvr.conf`
- Redémarrer tous les services `mongodb` :
 - o `sudo systemctl restart mongodb_configsvr.service`
 - o `sudo systemctl restart mongodb_shardsvr.service`

4.3. Vérifier le bon fonctionnement de la clé

- **Se connecter à un configServer** : `/apps/mongodb/bin/mongo --host=mongodb-1 --port=27019 -u "admin" -p "admin"`
 - o `show dbs;`

5. Création des routers

5.1. Création du fichier de configuration des routers

Sur chaque nœud :

- **Editer le fichier conf du router** : `sudo -u mongodb nano /home/mongodb/conf/router.conf`

```
net:
  port: 27017
  bindIpAll: true

processManagement:
  fork: true
  pidFilePath: /home/mongodb/run/mongodb-router.pid
  timeZoneInfo: /usr/share/zoneinfo

sharding:
  configDB: configReplSet/mongodb-1:27019,mongodb-2:27019,mongodb-3:27019

security:
  keyFile: /home/mongodb/ssl/mongodb.key

systemLog:
  destination: file
  logAppend: true
  path: /home/mongodb/logs/router.log
```

5.2. Création du service systemd pour lancer les routers

- **Editer le fichier de service du router** : `sudo nano /etc/systemd/system/mongodb-router.service`

```
[Unit]
Description=MongoDB Router
Documentation=https://docs.mongodb.org/manual
After=network.target

[Service]
User=mongodb
Group=mongodb
ExecStart=/apps/mongodb/bin/mongos --config=/home/mongodb/conf/router.conf
ExecStartPre=/usr/bin/chown -R mongodb:mongodb /apps/mongodb
ExecStartPre=/usr/bin/chown -R mongodb:mongodb /home/mongodb
ExecStopPost=/usr/bin/rm -f /home/mongodb/run/mongodb-27017.sock
ExecStopPost=/usr/bin/rm -f /home/mongodb/run/mongodb-router.pid
PermissionsStartOnly=true
Type=forking
# file size
LimitFSIZE=infinity
# cpu time
LimitCPU=infinity
# virtual memory size
LimitAS=infinity
# open files
LimitNOFILE=64000
# processes/threads
LimitNPROC=64000
# locked memory
LimitMEMLOCK=infinity
# total threads (user+kernel)
```

```
TasksMax=infinity
TasksAccounting=false
# Recommended limits for mongod as specified in
# https://docs.mongodb.com/manual/reference/ulimit/#recommended-ulimit-settings

[Install]
WantedBy=multi-user.target
```

- Lancer le service sur chaque nœud :
 - o `sudo systemctl enable mongodb-router.service`
 - o `sudo systemctl start mongodb-router.service`

5.3. Création d'un shard

Sur un seul nœud (ex : mongodb-1) :

- Se connecter au routeur primaire (*et uniquement au routeur primaire*)
 - o `sudo -u mongodb /apps/mongodb/bin/mongo --host=mongodb-1 --port=27017 -u "root" -p "root"`
- Créer le shard :
 - o `sh.addShard("shardRep1Set/mongodb-1:27018,mongodb-2:27018,mongodb-3:27018");`
- Vérifier la bonne mise en place de la réplication ainsi :
 - o `sh.status();`

6. Exemple de mise en place du sharding sur une base de données

Sur un seul nœud (ex : mongodb-1) :

- Se connecter au routeur primaire (*et uniquement au routeur primaire*)
 - o `sudo -u mongodb /apps/mongodb/bin/mongo --host=mongodb-1 --port=27017 -u "root" -p "root"`
- Créer une nouvelle base de données
 - o `use dbtest ;`
- Activer le sharding :
 - o `sh.enableSharding("testdb");`
- Créer une collection
 - o `db.createCollection("animal");`
 - o `db.animal.createIndex({"_id":1});`
- Sharder la collection
 - o `sh.shardCollection("testdb.animal",{"_id":1});`

Conclusion

Ainsi se termine le montage de notre architecture de sharding. Il ne vous reste plus qu'à utiliser votre cluster mongodb pour y créer vos collections et y exécuter vos requêtes. Bonne continuation.