

## MongoDB Sharding Handleiding – Configuratie & Validatie

### 1. Voorbereiding van de directory-structuur

Om een sharded MongoDB-cluster te configureren, creëerden we de volgende mappen:

- C:/mongo/config/data
- C:/mongo/shards/shard1/data
- C:/mongo/shards/shard2/data
- C:/mongo/shards/shard3/data
- C:/mongo/logs/

Deze structuur scheidt configuratie- en shardnodes, zoals aanbevolen voor een productiecluster.

### 2. Starten van de Config Server ReplicaSet

We starten de config server op poort 27019:

```
mongod --replSet configReplSet --configsvr --port 27019 --dbpath C:/mongo/config/data --bind_ip 127.0.0.1
```

Daarna initialiseren we de replicaset via mongosh:

```
rs.initiate({  
  _id: "configReplSet",  
  configsvr: true,  
  members: [  
    { _id: 0, host: "localhost:27019" }  
  ]  
})
```

### 3. Starten van de drie shard servers

Shard 1:

```
mongod --replSet shard1 --shardsvr --port 27020 --dbpath C:/mongo/shards/shard1/data
```

Shard 2:

```
mongod --replSet shard2 --shardsvr --port 27021 --dbpath C:/mongo/shards/shard2/data
```

Shard 3:

```
mongod --replSet shard3 --shardsvr --port 27022 --dbpath C:/mongo/shards/shard3/data
```

Elke shard wordt geconfigureerd als een afzonderlijke replicaset:

```
rs.initiate({  
    _id: "shard1",  
    members: [{ _id: 0, host: "localhost:27020" }]  
})
```

en analoog voor shard2 en shard3.

#### 4. Starten van de Mongos Router

```
mongos --configdb configReplSet/localhost:27019 --port 27017 --bind_ip 127.0.0.1
```

Deze router vormt het toegangspunt voor alle shardingfunctionaliteit.

#### 5. Verbinden met de Mongos

```
mongosh "mongodb://localhost:27017/?directConnection=false"
```

Controle:

```
db.hello()
```

→ moet tonen: msg: "isdbgrid"

#### 6. Shards toevoegen aan het cluster

In mongosh:

```
sh.addShard("shard1/localhost:27020")
```

```
sh.addShard("shard2/localhost:27021")
```

```
sh.addShard("shard3/localhost:27022")
```

Controle:

```
sh.status()
```

7. Sharding inschakelen voor de database

```
sh.enableSharding("catchem")
```

Daarna wordt de collection geshard:

```
sh.shardCollection("catchem.treasures", { treasure_id: "hashed" })
```

8. Importeren van data in de sharded cluster

We importeerden het JSON-bestand in batches via een PowerShell-script.

De inserts werden uitgevoerd via mongosh, telkens gericht op de mongos router.

Batchimport werd uitgevoerd tot alle ~128.000 documenten geladen waren.

9. Verifiëren dat de data verdeeld is over shards

Belangrijk bewijs:

```
db.treasures.getShardDistribution()
```

Dit toont (voorbeeld):

Shard shard1:

33.36% data, 33.28% documenten

Shard shard2:

33.34% data, 33.45% documenten

Shard shard3:

33.28% data, 33.25% documenten

Dit bevestigt dat de cluster correct gebalanceerd is en dat sharding actief werkt.

## 10. Conclusie

We hebben succesvol:

- ✓ Een MongoDB cluster geconfigureerd met 3 shards
- ✓ Een configserver replicaset opgezet
- ✓ Een mongos-router geconfigureerd
- ✓ De database geshard op treasure\_id
- ✓ Data geïmporteerd en verdeling geverifieerd

Deze handleiding documenteert alle stappen, commando's en validaties die nodig zijn om sharding correct op te zetten en aan te tonen.