Lab 7: Sharding

In this lab we will set up a cluster with two shards and shard a collection.

1. For this lab please create a separate working directory (e.g. /mongodb_sharding) and copy the file `lab_07.js` into this directory.

2. Apart from the two shards we also need a config server, thus we create the following three folders:

   ```
   > mkdir shard1
   > mkdir shard2
   > mkdir config
   ```

3. First we start one node as a standalone server (originating from our working directory):

   ```
   > mongod --dbpath shard1 --port 27010 --smallfiles --oplogSize 50
   ```

4. Now we start an additional Mongo Shell as following:

   ```
   > mongo --port 27010 --shell lab_07.js
   ```

   In this Mongo Shell we call the following function to populate the data base:

   ```
   > init()
   ```

   The initiation of the data base might take some seconds as 500.001 documents will be created. To verify everything worked as expected, please do the following:

   ```
   > use lab_07
   > db.trades.count()
   500001
   ```

5. Now we convert this standalone server into a single shard. To do this, we stop the `mongod` process and start it again, but this time with the option `--shardsvr`:

   ```
   > mongod --shardsvr --dbpath shard1 --port 27010 --smallfiles --
   oplogSize 50
   ```

6. Additionally we need to start a config server:

```
> mongod --configsvr --dbpath config --port 10000 --smallfiles --
oplogSize 50
```

7. Furthermore we need to start the routing service `mongos`:

```
> mongos --configdb localhost:10000
```

8. The first shard needs to be configured via a Mongo Shell connecting to `mongos` on the default port 27017:

```
> mongo
> sh.addShard("localhost:27010")
```

9. Next we check that the collection created in step 4 is available via the routing server `mongos`. This should be the case even though we haven't sharded the data base and collection yet:

```
> use lab_07
> db.trades.find()
```

10. Now we enable the sharding for the data base:

```
> sh.enableSharding("lab_07")
```

11. Before we are able to shard the collection on a compound shard key consisting of `ticker` and `time`, we need to create an index on that shard key first:

```
> db.trades.ensureIndex( { ticker:1, time:1 } )
```

12. After the index has been created successfully, we are now able to shard the collection:

```
> sh.shardCollection("lab_07.trades", { ticker:1, time:1 })
```

13. After the collection has been shared, we can take a closer look at the chunks of the sharded collection:

```
> use config
> db.chunks.find()
```

14. Aren't we still missing something? Exactly, we are still missing the second shard which we start as following:

```
> mongod --shardsvr --dbpath shard2 --port 27020 --smallfiles --
oplogSize 50
```

15. This shard needs to be added to the config first:

```
> sh.addShard( "localhost:27020" )
```

16. The balancer is now automatically distributing the chunks on the two shard. The progress can be checked by periodically calling the following aggregation:

```
> use config
> db.chunks.aggregate( [
        { $match : { ns : "lab_07.trades" } } ,
        { $group : { _id : "$shard", n : { $sum : 1 } } }
  ] )
```