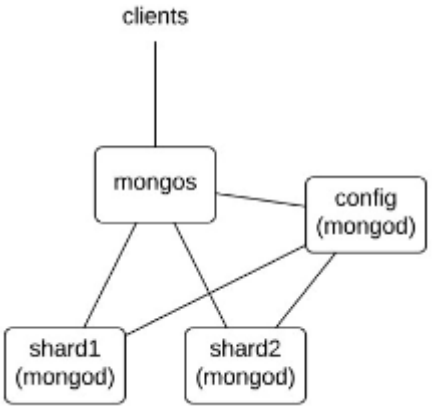
Assignment 3 MongoDB

1. **What is sharding in mongoDB?**

Sharding er en database architecture mønstre. Sharding arbejder med at adskille en tabels rækker i flere forskellige tabeller.

2. **What are the different components required to implement sharding?**



Mongo kommando er klienten, mens mongos er 'MongoDB Shard Utility'. Kommandoen mongod bruges til at starte MongoDB-serveren på din lokale maskine eller server. Derfra kan du oprette shards, configs og replica sets. Config kommunikere også med shards samt med mongos til at danne forbindelsen.

3. **Explain architecture of sharding in mongoDB?**

Sharding deler ens data op i to eller flere mindre bidder. Dette kaldes logiske shards. Database shards eksemplificerer en delt-intet-arkitektur. Det betyder, at shards er autonome. Da de er autonome deler de ikke af computer ressourcer. I nogle tilfælde kan det være fordelagtigt at replikere bestemte tabeller. Årsagen til dette er at tjene de pågældende shards som referencetabeller.

4. **Provide implementation of map and reduce function**

Map:

```
distinctName = function(tweets)
{
var number = twitter.components.tweets.user.screen_name + '',
  seen = []
  result = []
  i = number.length;

while(i -- )
```

```
{
seen[+number[i]]=1;
}

for(var i = 0; i<10; i++)
{
if(seen[i])
{
result[result.length]=i;
}
}
return result;
}



map = function()
{
 var names = distinctName(this);
 emit
 (
  {
   names: names,
   followers_count: this.components.tweets.user.followers_count
  },{
   count: 1
  }
 );
}
```

Reduce:

```
reduce = function(key, values) { var total = 0; for (var i = 0; i < values.length; i++) { total += values[i].count; } return { count : total }; }
```

5. **Provide execution command for running MapReduce**

```
results = db.runCommand
(
  {
    mapReduce: 'tweets',
    map: map,
    reduce:reduce,
    out: 'user.followers_count'
  }
)
```

Result:

```
result: 'tweets.user.followers_count',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1648048233, i: 2 }),
    signature: {
      hash: Binary(Buffer.from("0000000000000000000000000000000000000000",
"hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1648048233, i: 2 })
}
```

6. **Provide top 10 recorded out of the sorted result. (hint: use sort on the result returned by MapReduce)**

db.tweets.find().sort({'user.followers_count':-1},{followers_count:1}).limit(10)

Result:

1.
```
    name  'Barack Obama'
    followers_count  58672161
```
2.
```
    name  'Barack Obama'
    followers_count  58440499
```
3.
```
    name  'Barack Obama'
    followers_count  58434990
```
4.
```
    name  'Instagram'

    followers_count  38541311
```
5.
```
    name  'Shakira'
    followers_count  30724211
```
6.
```
    name  'Shakira'
    followers_count  30724211
```
7.
```
    name  'CNN Breaking News'
    followers_count  26405869
```
8.
```
    name  'CNN Breaking News'
    followers_count  26363618
```
9.
```
    name  'CNN Breaking News'
    followers_count  26244883
```

10.

```
name 'CNN Breaking News', [] } },
followers_count 26244883,
```