Chapter 3: Sharding

Lab - Shard a Collection

Problem:

At this point, your cluster is now configured for sharding. You should already have a CSRS, mongos, and primary shard.

In this lab, you will do the following with your cluster:

1. Add a second shard
2. Import a dataset onto your primary shard
3. Choose a shard key and shard your collection

1. Adding a Second Shard

Your second shard `m103-repl-2` will be a three-node replica set, just like your primary shard.

You can create the data paths for each node in `m103-repl-2` with the following command:

```
mkdir /var/mongodb/db/{4,5,6}
```

COPY

Here are the config files for this replica set:

Node 4:

COPY

```
storage:
  dbPath: /var/mongodb/db/4
  wiredTiger:
     engineConfig:
        cacheSizeGB: .1
net:
  bindIp: 192.168.103.100,localhost
  port: 27004
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/4/mongod.log
  logAppend: true
processManagement:
  fork: true
operationProfiling:
  slowOpThresholdMs: 50
replication:
```

```
  replSetName: m103-repl-2
sharding:
  clusterRole: shardsvr
```

Node 5:

COPY

```
storage:
  dbPath: /var/mongodb/db/5
  wiredTiger:
     engineConfig:
        cacheSizeGB: .1
net:
  bindIp: 192.168.103.100,localhost
  port: 27005
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/5/mongod.log
  logAppend: true
processManagement:
  fork: true
operationProfiling:
  slowOpThresholdMs: 50
replication:
  replSetName: m103-repl-2
sharding:
  clusterRole: shardsvr
```

Node 6:

COPY

```
storage:
  dbPath: /var/mongodb/db/6
  wiredTiger:
     engineConfig:
        cacheSizeGB: .1
net:
  bindIp: 192.168.103.100,localhost
  port: 27006
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/6/mongod.log
  logAppend: true
processManagement:
  fork: true
operationProfiling:
  slowOpThresholdMs: 50
replication:
  replSetName: m103-repl-2
```

```
sharding:
  clusterRole: shardsvr
```

We can now initialize `m103-repl-2` as a normal replica set.

Now exit the mongo shell and connect to mongos. We can add `m103-repl-2` as a shard with the following command:

```
sh.addShard("m103-repl-2/192.168.103.100:27004")
```

COPY

The output of `sh.status()` should now reflect the new shard.

2. Importing Data onto the Primary Shard

The dataset `products.json` is contained in your Vagrant box, in the `/dataset/` directory. You can import this dataset into mongos with the following command:

COPY

```
mongoimport --drop /dataset/products.json --port 26000 -u "m103-
admin" \
-p "m103-pass" --authenticationDatabase "admin" \
--db m103 --collection products
```

You can verify that the entire dataset was imported by querying the `m103` database on mongos. The collection `products` should contain exactly 516784 documents.

3. Sharding the Collection

Before you can shard your new `products` collection, you must enable sharding on the `m103`

Correct! SEE DETAILED ANSWER ×

```
sh.enableSharding("m103")
```

COPY

Once you've done this, it's time to choose a shard key on a single field in the `products` collection. To do this, you should review the qualities of a good shard key in the docs and the following information about the `products` collection:

- `_id` is a serial number for each product in this collection, rarely used in queries but important for internal MongoDB usage
- `sku` (Stock Keeping Unit) is a randomly generated integer unique to each product - this is commonly used to refer to specific products when updating stock quantities
- `name` is the name of the product as it appears in the store and on the website
- `type` is the type of product, with the possible values "Bundle", "Movie", "Music" and "Software"
- `regularPrice` is the regular price of the product, when there is no sale - this price changes every season
- `salePrice` is the price of a product during a sale - this price changes arbitrarily based on when sales occur
- `shippingWeight` is the weight of the product in kilograms, ranging between 0.01 and 1.00 - this value is not known for every product in the collection

If you want a better understanding of the distribution and the nature of the fields and their types, you can do so using Compass.

Once you've chosen a shard key, you must create an index on the shard key field:

```
```

```
db.products.createIndex({"<shard_key>": 1})
```

COPY

Once the index is created, shard the collection with the following command:

```
db.adminCommand( { shardCollection: "m103.products", key:
```

COPY

Choosing the Correct Shard Key

Now run the validation script in your vagrant and outside the mongo shell and enter the validation key you receive below:

```
vagrant@m103:~$ validate_lab_shard_collection
```

COPY

If you chose the wrong shard key, the validation script will give you an error. However, if you already imported the dataset, you must drop the collection and reimport it in order to choose a different shard key.

Attempts Remaining: Correct Answer

Enter answer here:

```
5a621149d083824c6d889865
```

Correct!

See detailed answer

Proceed to next section

Assignment is Due

18d:02hr:16m
Dec 17, 17:00 UTC

Your Grade

PASS/FAIL
Submitted