

DB Presentation:

Mongo DB data directory after running the command in the create_db.sh script:

```
WiredTiger                index-1-2152010350840750626.wt    log.cfg1
WiredTiger.lock           index-3-2152010350840750626.wt    log.cfg2
[WiredTiger.turtle        index-5-2152010350840750626.wt    log.diagnostic.data
[WiredTiger.wt            index-6-2152010350840750626.wt    log.mongos1
WiredTigerHS.wt           journal                           mongod.lock
_mdb_catalog.wt          log.a0                            shard-a0
collection-0-2152010350840750626.wt log.a0.2022-09-27T15-42-28        shard-a1
collection-2-2152010350840750626.wt log.a0.2022-09-27T15-45-38        shard-b0
collection-4-2152010350840750626.wt log.a0.2022-09-27T15-45-56        shard-b1
config-srv-0              log.a1                            sizeStorer.wt
config-srv-1              log.b0                            storage.bson
config-srv-2              log.b1
diagnostic.data           log.cfg0
```

Config server:

- ניתן לשים לב שעבור ה-config-srv קיימות כ-3 רפליקות (שכפולים) – 1 ראשי (PRIMARY) ו-2 משניות (SECONDARY).

1. Config server:

```
[config-srv [direct: primary] test> rs.status()
{
  set: 'config-srv',
  date: ISODate("2022-09-27T19:42:06.007Z"),
  myState: 1,
  term: Long("1"),
  syncSourceHost: '',
  syncSourceId: -1,
  configsvr: true,
  heartbeatIntervalMillis: Long("2000"),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1664307725, i: 1 }), t: Long("1") },
    lastCommittedWallTime: ISODate("2022-09-27T19:42:05.894Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1664307725, i: 1 }), t: Long("1") },
    appliedOpTime: { ts: Timestamp({ t: 1664307725, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1664307725, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2022-09-27T19:42:05.894Z"),
    lastDurableWallTime: ISODate("2022-09-27T19:42:05.894Z")
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1664307692, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate("2022-09-27T15:26:43.951Z"),
    electionTerm: Long("1"),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1664292403, i: 1 }), t: Long("-1") },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1664292403, i: 1 }), t: Long("-1") },
    numVotesNeeded: 1,
    priorityAtElection: 1,
    electionTimeoutMillis: Long("10000"),
    newTermStartDate: ISODate("2022-09-27T15:26:44.264Z"),
    wMajorityWriteAvailabilityDate: ISODate("2022-09-27T15:26:45.685Z")
  },
}
```

2. Config server:

```
members: [
  {
    _id: 0,
    name: 'localhost:27020',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 15655,
    optime: { ts: Timestamp({ t: 1664307725, i: 1 }), t: Long("1") },
    optimeDate: ISODate("2022-09-27T19:42:05.000Z"),
    lastAppliedWallTime: ISODate("2022-09-27T19:42:05.894Z"),
    lastDurableWallTime: ISODate("2022-09-27T19:42:05.894Z"),
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: '',
    electionTime: Timestamp({ t: 1664292403, i: 2 }),
    electionDate: ISODate("2022-09-27T15:26:43.000Z"),
    configVersion: 5,
    configTerm: 1,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: 'localhost:27021',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 15189,
    optime: { ts: Timestamp({ t: 1664307723, i: 1 }), t: Long("1") },
    optimeDurable: { ts: Timestamp({ t: 1664307723, i: 1 }), t: Long("1") },
    optimeDate: ISODate("2022-09-27T19:42:03.000Z"),
    optimeDurableDate: ISODate("2022-09-27T19:42:03.000Z"),
    lastAppliedWallTime: ISODate("2022-09-27T19:42:05.894Z"),
    lastDurableWallTime: ISODate("2022-09-27T19:42:05.894Z"),
    lastHeartbeat: ISODate("2022-09-27T19:42:04.756Z"),
    lastHeartbeatRecv: ISODate("2022-09-27T19:42:04.515Z"),
    pingMs: Long("0"),
    lastHeartbeatMessage: '',
    syncSourceHost: 'localhost:27020',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 5,
    configTerm: 1
  },
]
```

3. Config server:

```
{
  _id: 2,
  name: 'localhost:27022',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 15167,
  optime: { ts: Timestamp({ t: 1664307724, i: 1 }), t: Long("1") },
  optimeDurable: { ts: Timestamp({ t: 1664307724, i: 1 }), t: Long("1") },
  optimeDate: ISODate("2022-09-27T19:42:04.000Z"),
  optimeDurableDate: ISODate("2022-09-27T19:42:04.000Z"),
  lastAppliedWallTime: ISODate("2022-09-27T19:42:05.894Z"),
  lastDurableWallTime: ISODate("2022-09-27T19:42:05.894Z"),
  lastHeartbeat: ISODate("2022-09-27T19:42:05.210Z"),
  lastHeartbeatRecv: ISODate("2022-09-27T19:42:05.006Z"),
  pingMs: Long("0"),
  lastHeartbeatMessage: '',
  syncSourceHost: 'localhost:27021',
  syncSourceId: 1,
  infoMessage: '',
  configVersion: 5,
  configTerm: 1
}
],
ok: 1,
lastCommittedOpTime: Timestamp({ t: 1664307725, i: 1 }),
'$clusterTime': {
  clusterTime: Timestamp({ t: 1664307725, i: 1 }),
  signature: {
    hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
    keyId: Long("0")
  }
},
operationTime: Timestamp({ t: 1664307725, i: 1 })
}
```

Sharding:

- ניתן לראות בחלק זה כי טבלת qa_data מחולקת ל-2 מחיצות a, b (shards) אשר לכל מחיצה כזאת קיימות כ-2 רפליקות, אחת מרכזית (PRIMARY) ואחת משנית (SECONDARY).
- ניתן לראות את כמות הנתונים (93625 שאלות) ואת פילוגם למחיצות ביחס של: (a – 50.26%, b – 49.73%).

```
[[direct: mongos] QAEngine> db.qa_data.getShardDistribution()
Shard b at b/localhost:27012,localhost:27013
{
  data: '8.51MiB',
  docs: 46475,
  chunks: 2,
  'estimated data per chunk': '4.25MiB',
  'estimated docs per chunk': 23237
}
---
Shard a at a/localhost:27010,localhost:27011
{
  data: '8.6MiB',
  docs: 47150,
  chunks: 2,
  'estimated data per chunk': '4.3MiB',
  'estimated docs per chunk': 23575
}
---
Totals
{
  data: '17.12MiB',
  docs: 93625,
  chunks: 4,
  'Shard b': [
    '49.73 % data',
    '49.63 % docs in cluster',
    '192B avg obj size on shard'
  ],
  'Shard a': [
    '50.26 % data',
    '50.36 % docs in cluster',
    '191B avg obj size on shard'
  ]
}
```

- בנוסף, ניתן לראות כי פונקציית החלוקה למחיצות (shards) קורת על-פי פונקציית hash על השדה .qa_id

```
collections: {
  'QAEngine.qa_data': {
    shardKey: { qa_id: 'hashed' },
    unique: false,
    balancing: true,
    chunkMetadata: [ { shard: 'a', nChunks: 2 }, { shard: 'b', nChunks: 2 } ]
```

```
chunks: [
  { min: { qa_id: MinKey() }, max: { qa_id: Long("-4611686018427387902") }, 'on shard': 'a',
  { min: { qa_id: Long("-4611686018427387902") }, max: { qa_id: Long("0") }, 'on shard': 'a',
  { min: { qa_id: Long("0") }, max: { qa_id: Long("4611686018427387902") }, 'on shard': 'b',
  { min: { qa_id: Long("4611686018427387902") }, max: { qa_id: MaxKey() }, 'on shard': 'b',
],
```

```
[direct: mongos] config> db.shards.find()
[
  {
    _id: 'a',
    host: 'a/localhost:27010,localhost:27011',
    state: 1,
    topologyTime: Timestamp({ t: 1664294711, i: 4 })
  },
  {
    _id: 'b',
    host: 'b/localhost:27012,localhost:27013',
    state: 1,
    topologyTime: Timestamp({ t: 1664294749, i: 4 })
  }
]
```