

Task 6 - Налаштування реплікації та перевірка відмовостійкості MongoDB

Виконав: Потужний Богдан (ФІ-03)

Ознайомтесь з реплікацію даних в MongoDB

<http://docs.mongodb.org/manual/core/replication-introduction/>

Завдання:

- 1) Налаштувати реплікацію в конфігурації: Primary with Two Secondary Members (всі ноди можуть бути запущені як окремі процеси або у Docker контейнерах) - <http://docs.mongodb.org/manual/core/replica-set-architecture-three-members/>
 - Deploy a Replica Set for Testing and Development-
<http://docs.mongodb.org/manual/tutorial/deploy-replica-set-for-testing/>
 - <http://www.tugberkugurlu.com/archive/setting-up-a-mongodb-replica-set-with-docker-and-connecting-to-it-with-a-net-core-app>

Запускаємо 3 ноди використовуючи наступну команду

```
C:\Program Files\MongoDB\Server\7.0\bin>mongod.exe --port 27019 --dbpath "D:\kpi\high-load-systems\data\db2" --replSet rs0
{"t":{"$date":"2023-09-16T08:42:17.420+02:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0 to force-enable TLS 1.1 specify --sslDisabledProtocols 'none'"}

```

Під'єднуємося до головної ноди.

І в ній налаштовуємо сет реплік

```
rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "localhost:27018" },
    { _id: 1, host: "localhost:27019" },
    { _id: 2, host: "localhost:27020" }
  ]
})
```

У відповідь отримали ОК.

Подивимось на результат, ще за допомогою статусу

```
rs.status()
[
  {
    "$clusterTime": {
      "clusterTime": {"$timestamp": {"t": 1694846881, "i": 1}},
      "signature": {
        "hash": {"$binary": {"base64": "AAAAAAAAAAAAAAAAAAAAAAAAA=",
"subType": "00"}},
        "keyId": 0
      }
    },
    "date": {"$date": "2023-09-16T06:48:03.201Z"},
    "electionCandidateMetrics": {
      "lastElectionReason": "electionTimeout",
      "lastElectionDate": {"$date": "2023-09-16T06:44:31.497Z"},
      "electionTerm": 1,
      "lastCommittedOpTimeAtElection": {
        "ts": {"$timestamp": {"t": 1694846660, "i": 1}},
        "t": -1
      }
    }
  ]
}
```

```

    },
    "lastSeenOpTimeAtElection": {
      "ts": {"$timestamp": {"t": 1694846660, "i": 1}},
      "t": -1
    },
    "numVotesNeeded": 2,
    "priorityAtElection": 1,
    "electionTimeoutMillis": 10000,
    "numCatchUpOps": 0,
    "newTermStartDate": {"$date": "2023-09-16T06:44:31.563Z"},
    "wMajorityWriteAvailabilityDate": {"$date": "2023-09-
16T06:44:32.171Z"}
  },
  "heartbeatIntervalMillis": 2000,
  "lastStableRecoveryTimestamp": {"$timestamp": {"t": 1694846831,
"i": 1}},
  "majorityVoteCount": 2,
  "members": [
    {
      "_id": 0,
      "name": "localhost:27018",
      "health": 1,
      "state": 1,
      "stateStr": "PRIMARY",
      "uptime": 396,
      "optime": {
        "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
        "t": 1
      },
      "optimeDate": {"$date": "2023-09-16T06:48:01.000Z"},
      "lastAppliedWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
      "lastDurableWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
      "syncSourceHost": "",
      "syncSourceId": -1,
      "infoMessage": "",
      "electionTime": {"$timestamp": {"t": 1694846671, "i": 1}},
      "electionDate": {"$date": "2023-09-16T06:44:31.000Z"},
      "configVersion": 1,
      "configTerm": 1,
      "self": true,
      "lastHeartbeatMessage": ""
    },
    {
      "_id": 1,
      "name": "localhost:27019",
      "health": 1,
      "state": 2,
      "stateStr": "SECONDARY",
      "uptime": 222,
      "optime": {
        "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
        "t": 1
      },
      "optimeDurable": {
        "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
        "t": 1
      },
      "optimeDate": {"$date": "2023-09-16T06:48:01.000Z"},
      "optimeDurableDate": {"$date": "2023-09-16T06:48:01.000Z"},

```

```

        "lastAppliedWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
        "lastDurableWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
        "lastHeartbeat": {"$date": "2023-09-16T06:48:02.431Z"},
        "lastHeartbeatRecv": {"$date": "2023-09-16T06:48:03.091Z"},
        "pingMs": 0,
        "lastHeartbeatMessage": "",
        "syncSourceHost": "localhost:27018",
        "syncSourceId": 0,
        "infoMessage": "",
        "configVersion": 1,
        "configTerm": 1
    },
    {
        "_id": 2,
        "name": "localhost:27020",
        "health": 1,
        "state": 2,
        "stateStr": "SECONDARY",
        "uptime": 222,
        "optime": {
            "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
            "t": 1
        },
        "optimeDurable": {
            "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
            "t": 1
        },
        "optimeDate": {"$date": "2023-09-16T06:48:01.000Z"},
        "optimeDurableDate": {"$date": "2023-09-16T06:48:01.000Z"},
        "lastAppliedWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
        "lastDurableWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
        "lastHeartbeat": {"$date": "2023-09-16T06:48:02.431Z"},
        "lastHeartbeatRecv": {"$date": "2023-09-16T06:48:01.559Z"},
        "pingMs": 0,
        "lastHeartbeatMessage": "",
        "syncSourceHost": "localhost:27018",
        "syncSourceId": 0,
        "infoMessage": "",
        "configVersion": 1,
        "configTerm": 1
    }
],
"myState": 1,
"ok": 1,
"operationTime": {"$timestamp": {"t": 1694846881, "i": 1}},
"optimes": {
    "lastCommittedOpTime": {
        "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
        "t": 1
    },
    "lastCommittedWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
    "readConcernMajorityOpTime": {
        "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
        "t": 1
    },
    "appliedOpTime": {
        "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
        "t": 1
    }
},

```

```

    "durableOpTime": {
      "ts": {"$timestamp": {"t": 1694846881, "i": 1}},
      "t": 1
    },
    "lastAppliedWallTime": {"$date": "2023-09-16T06:48:01.729Z"},
    "lastDurableWallTime": {"$date": "2023-09-16T06:48:01.729Z"}
  },
  "set": "rs0",
  "syncSourceHost": "",
  "syncSourceId": -1,
  "term": 1,
  "votingMembersCount": 3,
  "writableVotingMembersCount": 3,
  "writeMajorityCount": 2
}
]

```

Бачимо, що усі ноди успішно підключені перейдемо до наступних завдань.

- 2) Продемонструвати Read Preference Modes: читання з *primary* і *secondary* node (<http://docs.mongodb.org/manual/core/read-preference/>)

```

db.collection.insertOne({data: "Some data"})

db.collection.find().readPref('primary')
db.collection.find().readPref('primaryPreferred')
db.collection.find().readPref('secondary')
db.collection.find().readPref('secondaryPreferred')
db.collection.find().readPref('nearest')

```

	Result 26-2	Result 26-3	Result 26-4	Result 26-5	Result 26-6 ×
	{ _id		{ data		
1	6505558d217371775860e952		Some data		

- 3) Спробувати зробити запис з однією відключеною ногою та *write concern* рівнім 3 та нескінченим таймаутом. Спробувати під час таймаута включити відключену ноду

```

db.collection.insertOne({data: " Some data node 3 off"}, {writeConcern:
{w: 3, wtimeout: 0}})

```

```

2
3 ✨ db.collection.insertOne({data: "Some data node 3 off"}, {writeConcern: {w: 3, wtimeout: 0}}) 4 s

```

	{ _id		{ data	
1	6505558d217371775860e952		Some data	
2	650557d7217371775860e958		Some data node 3 off	

- 4) Аналогічно попередньому пункту, але задати скінченний таймаут та дочекатись його закінчення. Перевірити чи данні записались і чи доступні на читання з рівнем *readConcern: "majority"*

```
db.collection.insertOne({data: "Some data node 3 off with timeout"},  
{writeConcern: {w: 3, wtimeout: 5000}})
```

waiting for replication timed out

Під'єднаємося до нашої 3-ої ноди і перевіримо чи там є ця інформація

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'config', 'local', and 'test'. Under 'test', there is a 'collection' folder. The main panel displays 'test.collection' with three documents:

- `{ "_id": ObjectId('6505558d217371775860e952'), "data": "Some data" }`
- `{ "_id": ObjectId('650557d7217371775860e958'), "data": "Some data node 3 off" }`
- `{ "_id": ObjectId('6505584b217371775860e95a'), "data": "Some data node 3 off with timeout" }`

Так вона була додана, тепер доєднаємося до головної ноди за наступним `mongodb://localhost:27018/collection?readConcernLevel=majority` доступом і спробуємо проглянути данні

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'config', 'local', and 'test'. Under 'test', there is a 'collection' folder. The main panel displays 'test.collection' with three documents:

- `{ "_id": ObjectId('6505558d217371775860e952'), "data": "Some data" }`
- `{ "_id": ObjectId('650557d7217371775860e958'), "data": "Some data node 3 off" }`
- `{ "_id": ObjectId('6505584b217371775860e95a'), "data": "Some data node 3 off with timeout" }`

Тут також усі дані відображаються

- 5) Продемонстрував перевибори primary node в відключивши поточний primary (Replica Set Elections) - <http://docs.mongodb.org/manual/core/replica-set-elections/>

Re-elections пройшли успішно, і був встановлений наступний порядок

```
[
  {
    "$clusterTime": {
      "clusterTime": {"$timestamp": {"t": 1694853417, "i": 1}},
      "signature": {
        "hash": {"$binary": {"base64": "AAAAAAAAAAAAAAAAAAAAAAAAAAAA=",
"subType": "00"}},
        "keyId": 0
      }
    },
    "date": {"$date": "2023-09-16T08:37:06.678Z"},
    "electionCandidateMetrics": {
      "lastElectionReason": "stepUpRequestSkipDryRun",
      "lastElectionDate": {"$date": "2023-09-16T08:34:05.800Z"},
      "electionTerm": 2,
      "lastCommittedOpTimeAtElection": {
        "ts": {"$timestamp": {"t": 1694853237, "i": 1}},
        "t": 1
      },
      "lastSeenOpTimeAtElection": {
        "ts": {"$timestamp": {"t": 1694853237, "i": 1}},
        "t": 1
      },
      "numVotesNeeded": 2,
      "priorityAtElection": 1,
      "electionTimeoutMillis": 10000,
      "priorPrimaryMemberId": 0,
      "numCatchUpOps": 0,
      "newTermStartDate": {"$date": "2023-09-16T08:34:05.885Z"},
      "wMajorityWriteAvailabilityDate": {"$date": "2023-09-16T08:34:05.926Z"}
    },
    "electionParticipantMetrics": {
      "votedForCandidate": true,
      "electionTerm": 1,
      "lastVoteDate": {"$date": "2023-09-16T06:44:31.501Z"},
      "electionCandidateMemberId": 0,
      "voteReason": "",
      "lastAppliedOpTimeAtElection": {
        "ts": {"$timestamp": {"t": 1694846660, "i": 1}},
        "t": -1
      },
      "maxAppliedOpTimeInSet": {
        "ts": {"$timestamp": {"t": 1694846660, "i": 1}},
        "t": -1
      },
      "priorityAtElection": 1
    },
    "heartbeatIntervalMillis": 2000,
    "lastStableRecoveryTimestamp": {"$timestamp": {"t": 1694853396, "i": 1}},
  }
]
```

```

"majorityVoteCount": 2,
"members": [
  {
    "_id": 0,
    "name": "localhost:27018",
    "health": 0,
    "state": 8,
    "stateStr": "(not reachable/healthy)",
    "uptime": 0,
    "optime": {
      "ts": {"$timestamp": {"t": 0, "i": 0}},
      "t": -1
    },
    "optimeDurable": {
      "ts": {"$timestamp": {"t": 0, "i": 0}},
      "t": -1
    },
    "optimeDate": {"$date": "1970-01-01T00:00:00.000Z"},
    "optimeDurableDate": {"$date": "1970-01-01T00:00:00.000Z"},
    "lastAppliedWallTime": {"$date": "2023-09-16T08:34:15.926Z"},
    "lastDurableWallTime": {"$date": "2023-09-16T08:34:15.926Z"},
    "lastHeartbeat": {"$date": "2023-09-16T08:37:04.430Z"},
    "lastHeartbeatRecv": {"$date": "2023-09-16T08:34:20.564Z"},
    "pingMs": 0,
    "lastHeartbeatMessage": "Error connecting to localhost:27018
(127.0.0.1:27018) :: caused by :: onInvoke :: caused by :: No connection
could be made because the target machine actively refused it.",
    "syncSourceHost": "",
    "syncSourceId": -1,
    "infoMessage": "",
    "configVersion": 1,
    "configTerm": 2
  },
  {
    "_id": 1,
    "name": "localhost:27019",
    "health": 1,
    "state": 1,
    "stateStr": "PRIMARY",
    "uptime": 6891,
    "optime": {
      "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
      "t": 2
    },
    "optimeDate": {"$date": "2023-09-16T08:36:57.000Z"},
    "lastAppliedWallTime": {"$date": "2023-09-16T08:36:57.013Z"},
    "lastDurableWallTime": {"$date": "2023-09-16T08:36:57.013Z"},
    "syncSourceHost": "",
    "syncSourceId": -1,
    "infoMessage": "",
    "electionTime": {"$timestamp": {"t": 1694853245, "i": 1}},
    "electionDate": {"$date": "2023-09-16T08:34:05.000Z"},
    "configVersion": 1,
    "configTerm": 2,
    "self": true,
    "lastHeartbeatMessage": ""
  },
  {
    "_id": 2,

```

```

    "name": "localhost:27020",
    "health": 1,
    "state": 2,
    "stateStr": "SECONDARY",
    "uptime": 4268,
    "optime": {
      "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
      "t": 2
    },
    "optimeDurable": {
      "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
      "t": 2
    },
    "optimeDate": {"$date": "2023-09-16T08:36:57.000Z"},
    "optimeDurableDate": {"$date": "2023-09-16T08:36:57.000Z"},
    "lastAppliedWallTime": {"$date": "2023-09-16T08:36:57.013Z"},
    "lastDurableWallTime": {"$date": "2023-09-16T08:36:57.013Z"},
    "lastHeartbeat": {"$date": "2023-09-16T08:37:05.645Z"},
    "lastHeartbeatRecv": {"$date": "2023-09-16T08:37:06.102Z"},
    "pingMs": 0,
    "lastHeartbeatMessage": "",
    "syncSourceHost": "localhost:27019",
    "syncSourceId": 1,
    "infoMessage": "",
    "configVersion": 1,
    "configTerm": 2
  }
],
"myState": 1,
"ok": 1,
"operationTime": {"$timestamp": {"t": 1694853417, "i": 1}},
"optimes": {
  "lastCommittedOpTime": {
    "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
    "t": 2
  },
  "lastCommittedWallTime": {"$date": "2023-09-16T08:36:57.013Z"},
  "readConcernMajorityOpTime": {
    "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
    "t": 2
  },
  "appliedOpTime": {
    "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
    "t": 2
  },
  "durableOpTime": {
    "ts": {"$timestamp": {"t": 1694853417, "i": 1}},
    "t": 2
  },
  "lastAppliedWallTime": {"$date": "2023-09-16T08:36:57.013Z"},
  "lastDurableWallTime": {"$date": "2023-09-16T08:36:57.013Z"}
},
"set": "rs0",
"syncSourceHost": "",
"syncSourceId": -1,
"term": 2,
"votingMembersCount": 3,
"writableVotingMembersCount": 3,
"writeMajorityCount": 2

```


}
]

- і що після відновлення роботи старої primary на неї реплікуються нові дані, які з'явилися під час її простою

```
✓ db.collection.insertOne({data: "Some data when primary(node1) off"});  
✓ db.collection.insertOne({data: "Some data when primary(node1) off 2"});  
✓ db.collection.insertOne({data: "Some data when primary(node1) off 3"});
```

Відновлюємо роботу попередньої головної ноди і перевіряємо чи відбувся запис

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays 'localhost:27018' and a list of databases: admin, config, local, and test. Under the 'test' database, the 'collection' is selected. The main panel shows the 'test.collection' documents. There are 6 documents listed, each with an '_id' and a 'data' field. The documents are: 1. '_id: ObjectId('6505558d217371775860e952'), data: 'Some data' 2. '_id: ObjectId('650557d7217371775860e958'), data: 'Some data node 3 off' 3. '_id: ObjectId('6505584b217371775860e95a'), data: 'Some data node 3 off with timeout' 4. '_id: ObjectId('650569b49f0e040b9fa20829'), data: 'Some data when primary(node1) off' 5. '_id: ObjectId('650569b59f0e040b9fa2082a'), data: 'Some data when primary(node1) off 2' 6. '_id: ObjectId('650569b79f0e040b9fa2082b'), data: 'Some data when primary(node1) off 3'

Усі данні було додано до попередньої головної ноди

- 6) Привести кластер до неконсистентного стану користуючись моментом часу коли *primary node* не відразу помічає відсутність *secondary node*

- відключивши дві *secondary node* протягом 5 сек. на мастер записати значення (з w:1) і перевірити, що воно записалось

Перевірили що головна нода знаходиться на порту 27018, а отже вимикаємо 27019 та 27020

```
db.collection.insertOne({data: "2 nodes turned of"},  
{writeConcern: {w: 1}})
```

```
db.collection.find()
```

Але, оскільки ми вимикаємо ноди, відразу відбувається процес перерозподілення головної ноди, і коли у нас залишається тільки одна нода, більшість відсутня а тому обрати головну ноду не можливо, що призводить до того що запис не можливий і усі інші підпункти б) не можуть бути виконані.

```
Command failed with error 10107 (NotWritablePrimary): 'not primary' on server localhost:27019. The full response is {"errorLabels":["RetryableWriteError"], "topologyVersion": {"processId": {"$oid": "65054e495b7d083b82425f38"}, "counter": 10}, "ok": 0.0, "errmsg": "not primary", "code": 10107, "codeName": "NotWritablePrimary", "$clusterTime": {"clusterTime": {"timestamp": {"t": 1694857141, "i": 2}}, "signature": {"hash": {"$binary": {"base64": "AAAAAAAAAAAAAAAAAAAAAAAAAAAA", "subType": "00"}}, "keyId": 0}}, "operationTime": {"timestamp": {"t": 1694857141, "i": 2}}}
```

- спробувати зчитати це значення з різними рівнями *read concern* -
`readConcern: {level: <"majority" | "local" | "linearizable">}`
- включити дві інші ноди таким чином, щоб вони не бачили попереднього мастера (його можна відключити) і дочекатись поки вони оберуть нового мастера
- підключити (включити) попередню primary-ноду до кластеру і подивитись, що сталося зі значенням яке було на неї записано

7) Земмулювати eventual consistency за допомогою установки затримки реплікації для репліки <https://docs.mongodb.com/manual/tutorial/configure-a-delayed-replica-set-member/>

```
{
  _id: 1,
  host: 'localhost:27019',
  arbiterOnly: false,
  buildIndexes: true,
  hidden: false,
  priority: 0,
  tags: {},
  secondaryDelaySecs: 120,
  votes: 1
},
{
```

```
> rs.reconfig(cfg)
< {
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1694883169, i: 1 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1694883169, i: 1 })
}
```

- 8) Лишити *primary* та *secondary* для якої налаштована затримка реплікації.
Записати декілька значень. Спробувати прочитати значення з `readConcern: {level: "linearizable"}`

Має бути затримка поки значення не реплікуються на більшість нод

```
> db.collection.find().readConcern(level = "linearizable")  
✓ db.collection.find().readConcern(level = "linearizable")
```

```
{  
  _id: ObjectId("6505e06781040feab9f67fc4"),  
  data: 'delayed1'  
}
```

Опис додаткових команд Replication Reference -

<http://docs.mongodb.org/manual/reference/replication/>

Вимогу до оформлення протоколу:

Завдання здається особисто без протоколу, або надсилається протокол який має містити:

- команди та результати їх виконання