

Дмитренко ФБ-42мп

1)

```
test> rs.initiate({
: 0, host: "mongo1:27017" },
...   _id: "rs0",
" }
]
...   members: [
...     { _id: 0, host: "mongo1:27017" },
...     { _id: 1, host: "mongo2:27017" },
...     { _id: 2, host: "mongo3:27017" }
...   ]
... });
{ ok: 1 }
```

```
members: [
  {
    _id: 0,
    name: 'mongo1:27017',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 274,
    optime: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    optimeDate: ISODate('2024-12-12T19:42:24.000Z'),
    lastAppliedWallTime: ISODate('2024-12-12T19:42:24.744Z'),
    lastDurableWallTime: ISODate('2024-12-12T19:42:24.744Z'),
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: 'Could not find member to sync from',
    electionTime: Timestamp({ t: 1734032543, i: 1 }),
    electionDate: ISODate('2024-12-12T19:42:23.000Z'),
    configVersion: 1,
    configTerm: 1,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: 'mongo2:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 14,
    optime: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    optimeDurable: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    optimeDate: ISODate('2024-12-12T19:42:24.000Z'),
    optimeDurableDate: ISODate('2024-12-12T19:42:24.000Z'),
    lastAppliedWallTime: ISODate('2024-12-12T19:42:24.744Z'),
    lastDurableWallTime: ISODate('2024-12-12T19:42:24.744Z'),
    lastHeartbeat: ISODate('2024-12-12T19:42:25.374Z'),
    lastHeartbeatRecv: ISODate('2024-12-12T19:42:24.877Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: 'mongo1:27017',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 1,
    configTerm: 1
  },
  {
    _id: 2,
    name: 'mongo3:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 14,
    optime: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    optimeDurable: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    optimeDate: ISODate('2024-12-12T19:42:24.000Z'),
    optimeDurableDate: ISODate('2024-12-12T19:42:24.000Z'),
    lastAppliedWallTime: ISODate('2024-12-12T19:42:24.744Z'),
    lastDurableWallTime: ISODate('2024-12-12T19:42:24.744Z'),
    lastHeartbeat: ISODate('2024-12-12T19:42:25.374Z'),
    lastHeartbeatRecv: ISODate('2024-12-12T19:42:24.877Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: 'mongo1:27017',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 1,
    configTerm: 1
  }
]
```

```
rs0 [direct: primary] test> rs.isMaster();
{
  topologyVersion: {
    processId: ObjectId('675b3b90b8913eb958de0455'),
    counter: Long('6')
  },
  hosts: [ 'mongo1:27017', 'mongo2:27017', 'mongo3:27017' ],
  setName: 'rs0',
  setVersion: 1,
  ismaster: true,
  secondary: false,
  primary: 'mongo1:27017',
  me: 'mongo1:27017',
  electionId: ObjectId('7fffffff0000000000000001'),
  lastWrite: {
    opTime: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    lastWriteDate: ISODate('2024-12-12T19:42:24.000Z'),
    majorityOpTime: { ts: Timestamp({ t: 1734032544, i: 6 }), t: Long('1') },
    majorityWriteDate: ISODate('2024-12-12T19:42:24.000Z')
  },
  maxBsonObjectSize: 16777216,
  maxMessageSizeBytes: 48000000,
  maxWriteBatchSize: 100000,
  localTime: ISODate('2024-12-12T19:42:31.823Z'),
  logicalSessionTimeoutMinutes: 30,
  connectionId: 2,
  minWireVersion: 0,
  maxWireVersion: 17,
  readOnly: false,
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1734032544, i: 6 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1734032544, i: 6 }),
  isWritablePrimary: true
}
```

writeConcern = 3

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker stop mongo3
mongo3
```

```
rs0 [direct: primary] test> db.test.insertOne({ test: "timeout test" }, { writeConcern: { w: 3, wtimeout: 0 } });
[]
```

Нескінченно очікує поки підніметься нода

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker start mongo3
mongo3
```

```
rs0 [direct: primary] test> db.test.insertOne({ test: "timeout test" }, { writeConcern: { w: 3, wtimeout: 0 } });
{
  acknowledged: true,
  insertedId: ObjectId('675b3d96dc07d17111fe6911')
}
rs0 [direct: primary] test> []
```

```
rs0 [direct: primary] test> db.test.find();
[ { _id: ObjectId('675b3d96dc07d17111fe6911'), test: 'timeout test' } ]
rs0 [direct: primary] test> []
```

writeConcern = 3 + limited timeout

якщо всі ноди включені, або вимкнена нода встигла піднятися, то все працює

```
rs0 [direct: primary] test> db.test.find();
[ { _id: ObjectId('675b3d96dc07d17111fe6911'), test: 'timeout test' } ]
rs0 [direct: primary] test> db.test.insertOne({ test: "finite timeout test" }, { writeConcern: { w: 3, wtimeout: 5000 } });
{
  acknowledged: true,
  insertedId: ObjectId('675b3dfddc07d17111fe6912')
}
rs0 [direct: primary] test> db.test.find({ test: "finite timeout test" });
[
  {
    _id: ObjectId('675b3dfddc07d17111fe6912'),
    test: 'finite timeout test'
  }
]
```

Якщо нода не встигла піднятися

```
rs0 [direct: primary] test> db.test.insertOne({ test: "finite timeout test" }, { writeConcern: { w: 3, wtimeout: 5000 } });
Uncaught:
MongoWriteConcernError[WriteConcernFailed]: waiting for replication timed out
Additional information: {
  wtimeout: true,
  writeConcern: { w: 3, wtimeout: 5000, provenance: 'clientSupplied' }
}
Result: {
  n: 1,
  electionId: ObjectId('7fffffff0000000000000001'),
  opTime: { ts: Timestamp({ t: 1734033510, i: 1 }), t: Long('1') },
  writeConcernError: {
    code: 64,
    codeName: 'WriteConcernFailed',
    errmsg: 'waiting for replication timed out',
    errInfo: {
      wtimeout: true,
      writeConcern: { w: 3, wtimeout: 5000, provenance: 'clientSupplied' }
    }
  },
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1734033510, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1734033510, i: 1 })
}
rs0 [direct: primary] test> |
```

Змінило Primary, для цього вимкнув і увімкнув Primary ноду:

```
● PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker stop mongo1
mongo1
● PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker start mongo1
mongo1
```

```

members: [
  {
    _id: 0,
    name: 'mongo1:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 8,
    optime: { ts: Timestamp({ t: 1734033902, i: 1 }), t: Long('2') },
    optimeDate: ISODate('2024-12-12T20:05:02.000Z'),
    lastAppliedWallTime: ISODate('2024-12-12T20:05:02.614Z'),
    lastDurableWallTime: ISODate('2024-12-12T20:05:02.614Z'),
    syncSourceHost: 'mongo3:27017',
    syncSourceId: 2,
    infoMessage: '',
    configVersion: 1,
    configTerm: 2,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: 'mongo2:27017',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 7,
    optime: { ts: Timestamp({ t: 1734033902, i: 1 }), t: Long('2') },
    optimeDurable: { ts: Timestamp({ t: 1734033902, i: 1 }), t: Long('2') },
    optimeDate: ISODate('2024-12-12T20:05:02.000Z'),
    optimeDurableDate: ISODate('2024-12-12T20:05:02.000Z'),
    lastAppliedWallTime: ISODate('2024-12-12T20:05:02.614Z'),
    lastDurableWallTime: ISODate('2024-12-12T20:05:02.614Z'),
    lastHeartbeat: ISODate('2024-12-12T20:05:07.155Z'),
    lastHeartbeatRecv: ISODate('2024-12-12T20:05:06.622Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: '',
    electionTime: Timestamp({ t: 1734033872, i: 1 }),
    electionDate: ISODate('2024-12-12T20:04:32.000Z'),
    configVersion: 1,
    configTerm: 2
  },
  {
    _id: 2,
    name: 'mongo3:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',

```

Як бачимо тепер інша нода стала Primary

Далі виникли трабли з докером, тому поставив монго на хост, в принципі ніякої різниці немає, але надалі тестую на хості все.

```
rs0 [direct: primary] test> db.likes.insertOne({ counter: 0 });
{
  acknowledged: true,
  insertedId: ObjectId('675b4515cfea086ef7fe6911')
}
rs0 [direct: primary] test> []
```

Результати лічильників:

```
(whoisn3kk@ MacBook-Air-Daniil) - [~/Desktop/github/pvs_labs]
o $ /Library/Frameworks/Python.framework/Versions/3.10/bin/python3 /Users/whoisn3kk/Desktop/github/pvs_labs/lab4/lab4.py
Колекція створена. Поточний лічильник: 0
Тест з writeConcern=1 (без відключень)
Фінальний лічильник (writeConcern=1): 100000
Time: 21.878232955932617 s.

Лічильник оновлено!
Тест з writeConcern=majority (без відключень)
Фінальний лічильник (writeConcern=1): 100000
Time: 8.175148010253906 s.

Лічильник оновлено!
Тест з writeConcern=1 (з відключенням Primary)
Вимикаємо Primary на порту 27017...
Нода на порту 27017 вимкнена (PID: 61081).
Фінальний лічильник (writeConcern=1): 99997
Time: 26.521278142929077 s.

Очікуємо поки буде увімкнено вимкнену ноду на порту 27017...
Користувач увімкнув ноду!

Лічильник оновлено!
Тест з writeConcern=majority (з відключенням Primary)
Вимикаємо Primary на порту 27018...
Нода на порту 27018 вимкнена (PID: 61357).
Фінальний лічильник (writeConcern=1): 100000
Time: 37.96386504173279 s.
```