

Налаштування реплікації та перевірка відмовостійкості MongoDB

Налаштувати реплікацію в конфігурації: Primary with Two Secondary Members

3 докера с монго дб

```
bakhmut@bakhmut-PC:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
e5c8f34b24cd   mongo    "docker-entrypoint.s..." 17 seconds ago Up 16 seconds 27017/tcp      mongo-n3
c6cf4ccf2ea3   mongo    "docker-entrypoint.s..." 23 seconds ago Up 22 seconds 27017/tcp      mongo-n2
299585fbb078   mongo    "docker-entrypoint.s..." 31 seconds ago Up 29 seconds 27017/tcp      mongo-n1
bakhmut@bakhmut-PC:~$ docker exec -it mongo-n1 mongo
rs0:PRIMARY> rs.status()
{
  "set" : "rs0",
  "date" : ISODate("2021-12-15T19:01:10.177Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "majorityVoteCount" : 2,
  "writeMajorityCount" : 2,
  "votingMembersCount" : 3,
  "writableVotingMembersCount" : 3,
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1639594863, 1),
      "t" : NumberLong(1)
    },
    "lastCommittedWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1639594863, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1639594863, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1639594863, 1),
      "t" : NumberLong(1)
    },
    "lastAppliedWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
    "lastDurableWallTime" : ISODate("2021-12-15T19:01:03.119Z")
  }
}
```

```

    "members" : [
      {
        "_id" : 0,
        "name" : "mongo-n1:27017",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 300,
        "optime" : {
          "ts" : Timestamp(1639594863, 1),
          "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2021-12-15T19:01:03Z"),
        "lastAppliedWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
        "lastDurableWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "electionTime" : Timestamp(1639594732, 1),
        "electionDate" : ISODate("2021-12-15T18:58:52Z"),
        "configVersion" : 1,
        "configTerm" : 1,
        "self" : true,
        "lastHeartbeatMessage" : ""
      },
    ],
    {
      "_id" : 1,
      "name" : "mongo-n2:27017",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 148,
      "optime" : {
        "ts" : Timestamp(1639594863, 1),
        "t" : NumberLong(1)
      },
      "optimeDurable" : {
        "ts" : Timestamp(1639594863, 1),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2021-12-15T19:01:03Z"),
      "optimeDurableDate" : ISODate("2021-12-15T19:01:03Z"),
      "lastAppliedWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
      "lastDurableWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
      "lastHeartbeat" : ISODate("2021-12-15T19:01:09.051Z"),
      "lastHeartbeatRecv" : ISODate("2021-12-15T19:01:08.569Z"),
      "pingMs" : NumberLong(0),
      "lastHeartbeatMessage" : "",
      "syncSourceHost" : "mongo-n1:27017",
      "syncSourceId" : 0,
      "infoMessage" : "",
      "configVersion" : 1,
      "configTerm" : 1
    },
  ],

```

```
{
  "_id" : 2,
  "name" : "mongo-n3:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 148,
  "optime" : {
    "ts" : Timestamp(1639594863, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1639594863, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2021-12-15T19:01:03Z"),
  "optimeDurableDate" : ISODate("2021-12-15T19:01:03Z"),
  "lastAppliedWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
  "lastDurableWallTime" : ISODate("2021-12-15T19:01:03.119Z"),
  "lastHeartbeat" : ISODate("2021-12-15T19:01:09.050Z"),
  "lastHeartbeatRecv" : ISODate("2021-12-15T19:01:08.569Z"),
  "pingMs" : NumberLong(0),
  "lastHeartbeatMessage" : "",
  "syncSourceHost" : "mongo-n1:27017",
  "syncSourceId" : 0,
  "infoMessage" : "",
  "configVersion" : 1,
  "configTerm" : 1
}
```

Продемонструвати запис даних на *primary node* з різними Write Concern Levels

- Unacknowledged w: 0
- Acknowledged w: 1
- Journaled j : true
- AcknowledgedReplica w: “majority”

```
rs0:PRIMARY> db.newdb.insert({name: "Ann", age: 33}, {writeConcern: {w:0}})
WriteResult({ })
rs0:PRIMARY> db.newdb.insert({name: "Dan", age: 28}, {writeConcern: {w:1}})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.newdb.insert({name: "Edd", age: 21}, {writeConcern: {j: true}})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.newdb.insert({name: "Penny", age: 18}, {writeConcern: {w:"majority"}})
WriteResult({ "nInserted" : 1 })
```

Продемонструвати Read Preference Modes: читання з *primary* і *secondary node*

```
rs0:PRIMARY> db.newdb.find({}).readPref("primary")
{ "_id" : ObjectId("61ba476d17059f485c307ee9"), "name" : "Ann", "age" : 33 }
{ "_id" : ObjectId("61ba478917059f485c307eea"), "name" : "Dan", "age" : 28 }
{ "_id" : ObjectId("61ba47b917059f485c307eeb"), "name" : "Edd", "age" : 21 }
{ "_id" : ObjectId("61ba489f17059f485c307eec"), "name" : "Penny", "age" : 18 }
rs0:PRIMARY> db.newdb.find({}).readPref("secondary")
{ "_id" : ObjectId("61ba476d17059f485c307ee9"), "name" : "Ann", "age" : 33 }
{ "_id" : ObjectId("61ba478917059f485c307eea"), "name" : "Dan", "age" : 28 }
{ "_id" : ObjectId("61ba47b917059f485c307eeb"), "name" : "Edd", "age" : 21 }
{ "_id" : ObjectId("61ba489f17059f485c307eec"), "name" : "Penny", "age" : 18 }
```


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

```
bakhmut@bakhmut-PC:~$ sudo docker ps
[sudo] password for bakhmut:
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
e5c8f34b24cd   mongo    "docker-entrypoint.s..." 2 hours ago   Up 2 hours   27017/tcp    mongo-n3
c6cf4ccf2ea3   mongo    "docker-entrypoint.s..." 2 hours ago   Up 2 hours   27017/tcp    mongo-n2
299585fbb078   mongo    "docker-entrypoint.s..." 2 hours ago   Up 2 hours   27017/tcp    mongo-n1
bakhmut@bakhmut-PC:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c6cf4ccf2ea3   mongo    "docker-entrypoint.s..." 2 hours ago   Up 2 hours   27017/tcp    mongo-n2
299585fbb078   mongo    "docker-entrypoint.s..." 2 hours ago   Up 2 hours   27017/tcp    mongo-n1
bakhmut@bakhmut-PC:~$ 
---
rs0:SECONDARY> use admin
switched to db admin
rs0:SECONDARY> db.shutdownServer()
bakhmut@bakhmut-PC:~$ 
rs0:PRIMARY> db.newdb.insert({name: "Sam", age: 44}, {writeConcern: {w:3, wtimeout: 0}})
rs0:PRIMARY> 
switched to db admin
rs0:SECONDARY> db.shutdownServer()
bakhmut@bakhmut-PC:~$ 
```

Доки нода не підключиться, запис не відбудеться, помилки також не буде через нескінченний таймаут

```
rs0:PRIMARY> db.newdb.insert({name: "Sam", age: 44}, {writeConcern: {w:3, wtimeout: 0}})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> 
bakhmut@bakhmut-PC:~$ sudo docker start mongo-n3
mongo-n3
```

Аналогічно попередньому пункту, але задати скінченний таймаут та дочекатись його закінчення. Перевірити чи данні записались і чи доступні на чиання з рівнем *readConcern*: “majority”

```
rs0:PRIMARY> db.newdb.insert({name: "ReadConcern"}, {writeConcern: {w:3, wtimeout: 5000}})
WriteResult({
  "nInserted" : 1,
  "writeConcernError" : {
    "code" : 64,
    "codeName" : "WriteConcernFailed",
    "errmsg" : "waiting for replication timed out",
    "errInfo" : {
      "wtimeout" : true,
      "writeConcern" : {
        "w" : 3,
        "wtimeout" : 5000,
        "provenance" : "clientSupplied"
      }
    }
  }
})
rs0:PRIMARY> db.newdb.find({}).readConcern("majority")
[ { "_id" : ObjectId("61ba5418624237bd0e1b714c"), "name" : "Sam", "age" : 44 },
  { "_id" : ObjectId("61ba5433624237bd0e1b714d"), "name" : "Sam", "age" : 44 },
  { "_id" : ObjectId("61ba56e8624237bd0e1b714e"), "name" : "Major", "age" : 32 },
  { "_id" : ObjectId("61ba592e624237bd0e1b714f"), "name" : "Majjkhwf", "age" : 42 },
  { "_id" : ObjectId("61ba5a44624237bd0e1b7150"), "name" : "Dan", "age" : 42 },
  { "_id" : ObjectId("61ba5a53624237bd0e1b7151"), "name" : "Dan", "age" : 42 },
  { "_id" : ObjectId("61ba5ac7624237bd0e1b7152"), "name" : "ReadConcern" } ]
```

```
rs0:PRIMARY> db.newdb.insert({name: "ReadConcern"}, {writeConcern: {w:3, wtimeout: 5000}})
WriteResult({
  "nInserted" : 1,
  "writeConcernError" : {
    "code" : 64,
    "codeName" : "WriteConcernFailed",
    "errmsg" : "waiting for replication timed out",
    "errInfo" : {
      "wtimeout" : true,
      "writeConcern" : {
        "w" : 3,
        "wtimeout" : 5000,
        "provenance" : "clientSupplied"
      }
    }
  }
})
rs0:PRIMARY> db.newdb.find({}).readConcern("majority")
{ "_id" : ObjectId("61ba5418624237bd0e1b714c"), "name" : "Sam", "age" : 44 }
{ "_id" : ObjectId("61ba5433624237bd0e1b714d"), "name" : "Sam", "age" : 44 }
{ "_id" : ObjectId("61ba56e8624237bd0e1b714e"), "name" : "Major", "age" : 32 }
{ "_id" : ObjectId("61ba592e624237bd0e1b714f"), "name" : "Majjkhwf", "age" : 42 }
{ "_id" : ObjectId("61ba5a44624237bd0e1b7150"), "name" : "Dan", "age" : 42 }
{ "_id" : ObjectId("61ba5a53624237bd0e1b7151"), "name" : "Dan", "age" : 42 }
{ "_id" : ObjectId("61ba5ac7624237bd0e1b7152"), "name" : "ReadConcern" }
```

Продемонстрував перевибори primary node в відключивши поточний primary (Replica Set Elections)

```
rs0:SECONDARY>
rs0:PRIMARY>
rs0:PRIMARY>
rs0:PRIMARY> db.col2.insert({name:"While shutdown"})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>
```

Тепер mongo-n2 головна

```
rs0:SECONDARY> db.col2.find()
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
rs0:SECONDARY>
```

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

```
mongo-n1
bakhmut@bakhmut-PC:~$ sudo docker stop mongo-n1
[sudo] password for bakhmut:
mongo-n1
bakhmut@bakhmut-PC:~$ sudo docker start mongo-n1
mongo-n1
bakhmut@bakhmut-PC:~$ sudo docker exec -it mongo-node1 mongo
```

```
"_id" : 0,
"name" : "mongo-n1:27017",
"health" : 1,
"state" : 2,
"stateStr" : "SECONDARY",
"uptime" : 823.
```

```
rs0:SECONDARY> rs.secondaryOk()
rs0:SECONDARY> use ndb
switched to db ndb
rs0:SECONDARY> db.col2.find()
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
rs0:SECONDARY>
```

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

- відключити останню *secondary node* та протягом 5 сек. на мастері записати значення (з w:1) і перевірити, що воно записалось

```
bakhmut@bakhmut-PC:~$ sudo docker stop mongo-n1
mongo-n1
```

```
rs0:PRIMARY> db.col2.insert({name:" cluster incon2"}, {writeConcern:{w:1}})
WriteResult({ "nInserted" : 1 })
```

- спробувати зчитати це значення з різними рівнями *read concern* - readConcern: {level: <"majority"|"local"|"linearizable">}


```

rs0:PRIMARY> db.col2.insert({name:"hi"},{writeConcern: {w:1}})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>
rs0:PRIMARY>
rs0:PRIMARY> db.col2.find({}).readPref("primary").readConcern('local')
Error: error: {
  "topologyVersion" : {
    "processId" : ObjectId("61bb1d68f756270d8dfc69f5"),
    "counter" : NumberLong(11)
  },
  "ok" : 0,
  "errmsg" : "not master and slaveOk=false",
  "code" : 13435,
  "codeName" : "NotPrimaryNoSecondaryOk",
  "$clusterTime" : {
    "clusterTime" : Timestamp(1639656936, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1639656936, 1)
}
rs0:SECONDARY> db.col2.find({}).readPref("secondary").readConcern('local')
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
rs0:SECONDARY> db.col2.find({}).readPref("secondary").readConcern('majority')
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }

rs0:SECONDARY> db.col2.find({}).readPref("secondary").readConcern('linearizable')
Error: error: {
  "topologyVersion" : {
    "processId" : ObjectId("61bb1d68f756270d8dfc69f5"),
    "counter" : NumberLong(11)
  },
  "ok" : 0,
  "errmsg" : "cannot satisfy linearizable read concern on non-primary node",
  "code" : 10107,
  "codeName" : "NotWritablePrimary",
  "$clusterTime" : {
    "clusterTime" : Timestamp(1639656936, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1639656936, 1)
}

```

- включити дві інші ноди таким чином, щоб вони не бачили попереднього мастера (його можна відключити) і дочекатись поки вони оберуть нового мастера

```

bakhmut@bakhmut-PC:~$ sudo docker stop mongo-n1
mongo-n1
bakhmut@bakhmut-PC:~$ sudo docker start mongo-n3
mongo-n3
bakhmut@bakhmut-PC:~$ sudo docker start mongo-n2
mongo-n2

```

```

  "_id" : 2,
  "name" : "mongo-n3:27017",
  "health" : 1,
  "state" : 1,
  "stateStr" : "PRIMARY",
  "uptime" : 34,
  "optime" : {

```

- підключити (включити) попередню primary-ноду до кластеру і подивитись, що сталося зі значенням яке було на неї записано

```

rs0:PRIMARY> use ndb
switched to db ndb
rs0:PRIMARY> db.col2.find()
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
rs0:PRIMARY>

switched to db ndb
rs0:SECONDARY> db.col2.find({}).readConcern("local")
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
rs0:SECONDARY>

```

Земулювати eventual consistency за допомогою установки затримки реплікації для репліки


```

rs0:PRIMARY> cfg.members[0].priority = 0
0
rs0:PRIMARY> cfg.members[0].hidden = true
true
rs0:PRIMARY> cfg.members[0].secondaryDelaySecs = 100
100
rs0:PRIMARY> db.col2.remove({name: "some changes"})
WriteResult({ "nRemoved" : 1 })
rs0:PRIMARY> db.col2.remove({name: "some more changes"})
WriteResult({ "nRemoved" : 1 })
rs0:PRIMARY> rs.reconfig(cfg)
{
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1639658493, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1639658493, 1)
}
rs0:PRIMARY> db.col2.insert({name: "some changes"})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.col2.remove({name: "some more changes"})
WriteResult({ "nRemoved" : 0 })
rs0:PRIMARY> db.col2.insert({name: "some more changes"})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>

```

Звичайний нод

```

rs0:SECONDARY> db.col2.find()
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
{ "_id" : ObjectId("61bb3181c61579027976d85a"), "name" : "some changes" }
{ "_id" : ObjectId("61bb31d1c61579027976d85b"), "name" : "some more changes" }
rs0:SECONDARY>

```

Нод, який «затримали»

```

rs0:SECONDARY> db.col2.find()
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
rs0:SECONDARY> db.col2.find()

```

Можна побачити, що записи додаються через 100 секунд

```

rs0:PRIMARY> db.col2.insert({name: "some after 3 min"})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>
{ "_id" : ObjectId("61bb345bc61579027976d85e"), "name" : "some more changes" }
{ "_id" : ObjectId("61bb351ec61579027976d85f"), "name" : "some after 3 min" }
rs0:SECONDARY>
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
{ "_id" : ObjectId("61bb3419c61579027976d85d"), "name" : "some changes" }
{ "_id" : ObjectId("61bb345bc61579027976d85e"), "name" : "some more changes" }
rs0:SECONDARY>

```

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

```

rs0:PRIMARY> db.col2.insert({name: "some text"})
{ "_id" : ObjectId("61bb3419c61579027976d85d"), "name" : "some changes" }
{ "_id" : ObjectId("61bb345bc61579027976d85e"), "name" : "some more changes" }
{ "_id" : ObjectId("61bb351ec61579027976d85f"), "name" : "some after 3 min" }
rs0:SECONDARY>

```

Виникає затримка на запис

```

rs0:PRIMARY> db.col2.find({}).readConcern("linearizable")
{ "_id" : ObjectId("61bb1357ef76d559542492c0"), "name" : "regegr" }
{ "_id" : ObjectId("61bb14ac764d513d7f521692"), "name" : "While shutdown" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }
{ "_id" : ObjectId("61bb1d24764d513d7f521694"), "name" : " cluster incon2" }
{ "_id" : ObjectId("61bb2b91e7b69183ed570a57"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2b9ee7b69183ed570a58"), "name" : "cluster in" }
{ "_id" : ObjectId("61bb2ddae7b69183ed570a59"), "name" : "hi" }
{ "_id" : ObjectId("61bb3419c61579027976d85d"), "name" : "some changes" }
{ "_id" : ObjectId("61bb345bc61579027976d85e"), "name" : "some more changes" }
{ "_id" : ObjectId("61bb351ec61579027976d85f"), "name" : "some after 3 min" }
{ "_id" : ObjectId("61bb373ac61579027976d860"), "name" : "some text" }
{ "_id" : ObjectId("61bb1aaf764d513d7f521693"), "name" : " cluster incon" }

```

Та на зчитування