

COMPTE RENDU TP5 YASSINE MHIRSI

1.

```
root@DESKTOP-AEPIKF1:/home/top# docker network ls
NETWORK ID          NAME                DRIVER             SCOPE
0b77e8838558        bridge             bridge             local
161bc07e2962        host               host               local
e74b96247032        none               null               local
root@DESKTOP-AEPIKF1:/home/top# docker network create mongo_cluster
2e2dac8094b4a63de407f64ff783b084823fc452798239850938dc088ceefafa
root@DESKTOP-AEPIKF1:/home/top# docker network ls
NETWORK ID          NAME                DRIVER             SCOPE
0b77e8838558        bridge             bridge             local
161bc07e2962        host               host               local
2e2dac8094b4        mongo_cluster      bridge             local
e74b96247032        none               null               local
root@DESKTOP-AEPIKF1:/home/top#
```

2.

```
root@DESKTOP-AEPIKF1:/home/top# docker run -p 27022:27017 --name mongo1 --net mongo_cluster -d mongo mongod --replSet retp5
30bcd73f3be6b1ccf3fe3cd96de7489529f0e44fd51fe3b407c6a728c68f460
root@DESKTOP-AEPIKF1:/home/top# docker run -p 27023:27017 --name mongo2 --net mongo_cluster -d mongo mongod --replSet retp5
c3595cadee5a1a9388afa5df5a55e88a7120521be520409f351ce24966d98286
root@DESKTOP-AEPIKF1:/home/top# docker run -p 27024:27017 --name mongo3 --net mongo_cluster -d mongo mongod --replSet retp5
9c39967313b28e559e6ea80955cb72356c876cfd52675c06455808bb66cbb43
```

3.

```
test> rs.initiate({_id: "reptp5",version:1,members:[{_id:0,host:"mongo1:27017"},{_id:1,host:"mongo2:27017"},{_id:2,host:"mongo3:27017"}]})
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1730450476, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1730450476, i: 1 })
}
reptp5 [direct: secondary] test> rs.status()
{
  set: 'reptp5',
  date: ISODate('2024-11-01T08:41:25.054Z'),
  myState: 2,
  term: Long('0'),
  syncSourceHost: '',
  syncSourceId: -1,
  reptp5 [direct: secondary] test>
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
    lastCommittedWallTime: ISODate('2024-11-01T08:41:16.505Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
    appliedOpTime: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
    durableOpTime: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
    writtenOpTime: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
    lastAppliedWallTime: ISODate('2024-11-01T08:41:16.505Z'),
    lastDurableWallTime: ISODate('2024-11-01T08:41:16.505Z'),
    lastWrittenWallTime: ISODate('2024-11-01T08:41:16.505Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1730450476, i: 1 }),
  members: [
    {
      _id: 0,
      name: 'mongo1:27017',
      health: 1,
      state: 2,
      stateStr: 'SECONDARY',
      uptime: 853,
      optime: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
      optimeDate: ISODate('2024-11-01T08:41:16.000Z'),
      optimeWritten: { ts: Timestamp({ t: 1730450476, i: 1 }), t: Long('-1') },
      optimeWrittenDate: ISODate('2024-11-01T08:41:16.000Z'),
```

Question 1 : Quel nœud a été élu le maitre ?

-le nœud qui a élu le maitre est mongo1

Question 2 : Quels sont les nœuds esclaves ?

-les nœuds esclaves sont mongo2 et mongo3

```

reptp5 [direct: primary] test> rs.isMaster()
{
  topologyVersion: {
    processId: ObjectId('672490e0990ed09be17d1809'),
    counter: Long('6')
  },
  hosts: [ 'mongo1:27017', 'mongo2:27017', 'mongo3:27017' ],
  setName: 'reptp5',
  setVersion: 1,
  ismaster: true,
  secondary: false,
  primary: 'mongo1:27017',
  me: 'mongo1:27017',
  electionId: ObjectId('7fffffff0000000000000001'),
  lastWrite: {
    opTime: { ts: Timestamp({ t: 1730451307, i: 1 }), t: Long('1') },
    lastWriteDate: ISODate('2024-11-01T08:55:07.000Z'),
    majorityOpTime: { ts: Timestamp({ t: 1730451307, i: 1 }), t: Long('1') },
    majorityWriteDate: ISODate('2024-11-01T08:55:07.000Z')
  },
  maxBsonObjectSize: 16777216,
  maxMessageSizeBytes: 48000000,
  maxWriteBatchSize: 100000,
  localTime: ISODate('2024-11-01T08:55:14.308Z'),
  logicalSessionTimeoutMinutes: 30,
  connectionId: 8,
  minWireVersion: 0,
  maxWireVersion: 25,
  readOnly: false,
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1730451307, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1730451307, i: 1 }),
  isWritablePrimary: true
}
reptp5 [direct: primary] test> 

```

4. Réplication :

```
reptp5 [direct: primary] tp5> db.rep.insertMany([ { "Name": "Alice Johnson", "Title": "Data Analyst", "Year": 2019, "Department": "Data Science", "Location": "San Francisco" }, { "Name": "Bob Smith", "Title": "Product Manager", "Year": 2021, "Department": "Product", "Location": "Seattle" }, { "Name": "Charlie Brown", "Title": "UI/UX Designer", "Year": 2018, "Department": "Design", "Location": "Chicago" }, { "Name": "Diana Lee", "Title": "Marketing Specialist", "Year": 2022, "Department": "Marketing", "Location": "Los Angeles" }, { "Name": "Evan Rogers", "Title": "Financial Analyst", "Year": 2020, "Department": "Finance", "Location": "Boston" } ] );
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672499f4a768f1557f964038'),
    '1': ObjectId('672499f4a768f1557f964039'),
    '2': ObjectId('672499f4a768f1557f96403a'),
    '3': ObjectId('672499f4a768f1557f96403b'),
    '4': ObjectId('672499f4a768f1557f96403c')
  }
}
reptp5 [direct: primary] tp5> db.rep.find()
[
  {
    _id: ObjectId('672499f4a768f1557f964038'),
    Name: 'Alice Johnson',
    Title: 'Data Analyst',
    Year: 2019,
    Department: 'Data Science',
    Location: 'San Francisco'
  },
  {
    _id: ObjectId('672499f4a768f1557f964039'),
    Name: 'Bob Smith',
    Title: 'Product Manager',
    Year: 2021,
    Department: 'Product',
    Location: 'Seattle'
  },
  {
    _id: ObjectId('672499f4a768f1557f96403a'),
    Name: 'Charlie Brown',
    Title: 'UI/UX Designer',
    Year: 2018,
    Department: 'Design',
    Location: 'Chicago'
  },
  {
    _id: ObjectId('672499f4a768f1557f96403b'),
    Name: 'Diana Lee',
    Title: 'Marketing Specialist',
    Year: 2022,
    Department: 'Marketing',
    Location: 'Los Angeles'
  },
  {
    _id: ObjectId('672499f4a768f1557f96403c'),
    Name: 'Evan Rogers',
    Title: 'Financial Analyst',
    Year: 2020,
    Department: 'Finance',
    Location: 'Boston'
  }
]
```

Se connecter sur l'un des secondaires

rs.secondaryOk()

```
reptp5 [direct: secondary] tp5> rs.secondaryOk()
DeprecationWarning: .setSecondaryOk() is deprecated. Use .setReadPref("primaryPreferred") instead
Setting read preference from "primary" to "primaryPreferred"
```

Est-ce qu'on peut lire dans les deux secondaires? Expliquer pourquoi

Oui, on peut lire depuis les secondaires, car les secondaires contiennent des copies des données du primaire.

Est-ce qu'on peut insérer dans les deux secondaires ? Expliquer pourquoi.

Non, on ne peut pas écrire dans les secondaires, car seules les écritures sur le primaire sont répliquées vers les secondaires.

5. Réplication et reprise en cas de panne

>db.shutdownServer()

Question : Qui devient le maître?

le nouveau maitre est le premier secondaire mongo2

Question : Qui est encore le maître à la fin?

Mongo2 est le maitre a la fin