MongoDB教育訓練-20211223-01

Hideden Replica Set Member 隱藏結點，不太使用（歷史，所有請求不會訪問）

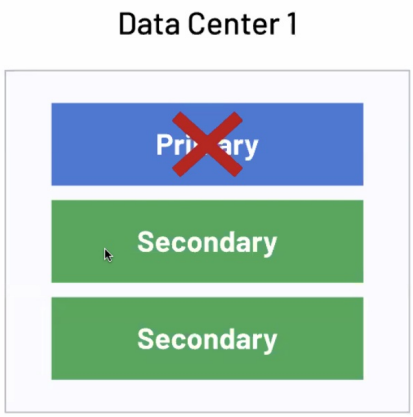
Delayed Replica Set Member 不太使用，先將資料取回不同步(延遲一段時間，保持舊資料，備援用)

Secondary Server

- A Secondary maintains a copy of the Primary's data set
- Type of Secondary
  - Priority 0 Replica Set Member
  - Hidden Replica Set Member
  - Delayed Replica Set Member

Note: Now Hidden and Delayed are considered Bad

3 Nodes, 1 data center

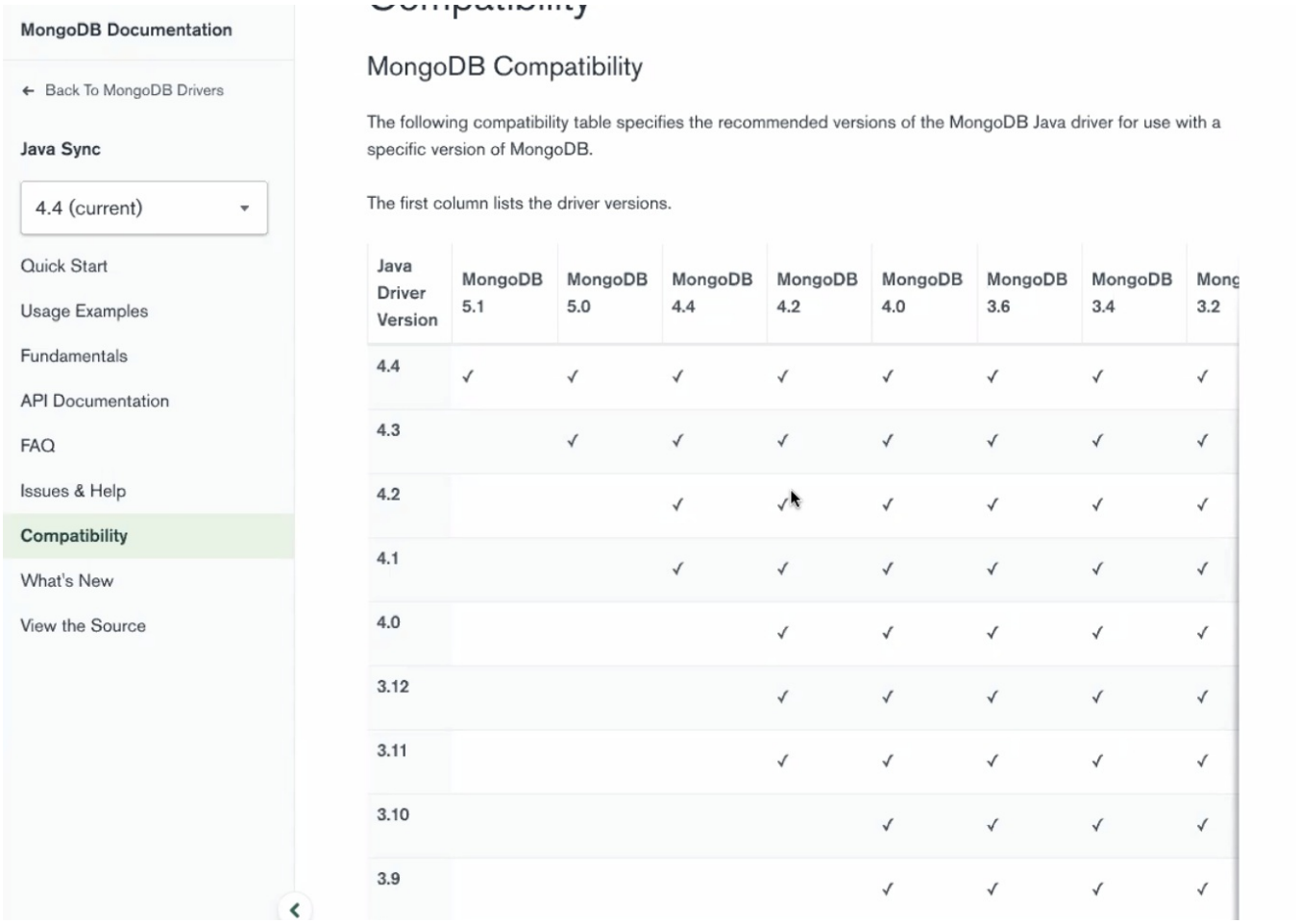


- 如何知道主結點？ Driver 告訴你
- Driver 幫你追蹤網路架構，不需知道主從節點
- 為截點點打上標籤，通過driver，告訴你

Drivers and Replica sets

- When coding with MongoDB, the Driver keeps track of the topology
- It knows where and how to route requests
- You don't need to know what is a primary or secondary at any point
- You can specify where you want a read to go logically.
- When upgrading the server, it is important to check driver compatibility.

Driver 版本匹配



MongoDB Documentation

← Back To MongoDB Drivers

Java Sync

4.4 (current)

Quick Start
Usage Examples
Fundamentals
API Documentation
FAQ
Issues & Help
Compatibility
What's New
View the Source

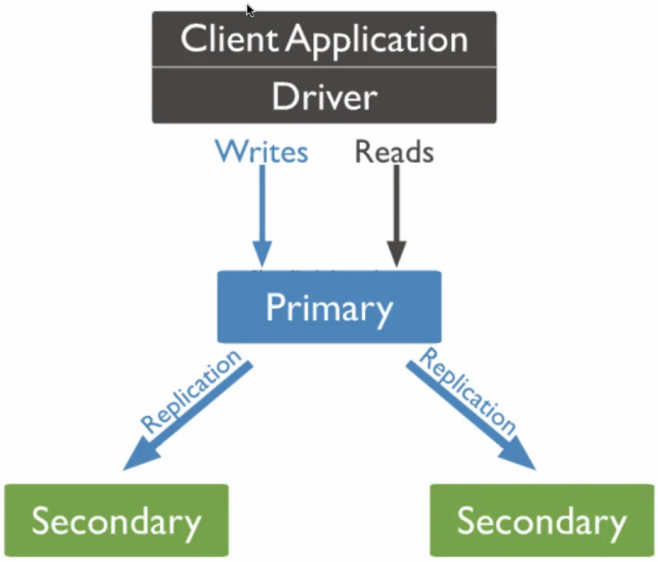
MongoDB Compatibility

The following compatibility table specifies the recommended versions of the MongoDB Java driver for use with a specific version of MongoDB.

The first column lists the driver versions.

Java Driver Version	MongoDB 5.1	MongoDB 5.0	MongoDB 4.4	MongoDB 4.2	MongoDB 4.0	MongoDB 3.6	MongoDB 3.4	Mong 3.2
4.4	✓	✓	✓	✓	✓	✓	✓	✓
4.3		✓	✓	✓	✓	✓	✓	✓
4.2			✓	✓	✓	✓	✓	✓
4.1			✓	✓	✓	✓	✓	✓
4.0				✓	✓	✓	✓	✓
3.12				✓	✓	✓	✓	✓
3.11				✓	✓	✓	✓	✓
3.10					✓	✓	✓	✓
3.9					✓	✓	✓	✓

The Replication process



The Replication process

1. Applications write all changes to the Primary
2. The Primary applies the change at time T and records this change in its Operation Log (Oplog)
3. The secondaries are observing this Oplog and read the changes up to time T
4. The secondaries apply new changes up to time T to themselves
5. The secondaries record them in their own oplogs
6. The secondaries request information after time T
7. The primary knows the latest seen T for each secondary. (This is important)

The Replication process

1. Applications write all changes to the Primary
2. The Primary applies the change at time T and records this change in its Operation Log (Oplog)
3. The secondaries are observing this Oplog and read the changes up to time T
4. The secondaries apply new changes up to time T to themselves
5. The secondaries record them in their own oplogs
6. The secondaries request information after time T
7. The primary knows the latest seen T for each secondary. (This is important)

Oplog 會紀錄所有變更，增刪改查

The Oplog

- The Oplog is a read-only capped collection of BSON documents.
- Each time a write changes a document, it's recorded in the Oplog.
- The Oplog records changes like dropping a collection or creating an index.
- It is independent of the database's binary form.
- The oplog is in a database called 'local'; the collection is 'oplog.rs'

Oplog - one entry per change

```
db.foo.deleteMany({ age : 30 })
```

This will be represented in the oplog with records such as the following:

```
{ "ts" : Timestamp(1407159845, 5), "h" : NumberLong("-704612487691926908"), "v" :
2, "op" : "d", "ns" : "bar.foo", "b" : true, "o" : { "_id" : 65 } }
```

```
 { "ts" : Timestamp(1407159845, 1), "h" : NumberLong("6014126345225019794"), "v"
: 2, "op" : "d", "ns" : "bar.foo", "b" : true, "o" : { "_id" : 333 } }
```

```
 { "ts" : Timestamp(1407159845, 4), "h" : NumberLong("8178791764238465439"), "v"
: 2, "op" : "d", "ns" : "bar.foo", "b" : true, "o" : { "_id" : 447 } }
```

Oplog entries are Idempotent

Operations in Oplog can be played multiple times as they do not depend on the previous state

`{$inc:{a: 2}}` becomes `{$set:{ a: 5}}` assuming a was previously equal to 3

`{$push:{b: "cheese"}}` becomes `{$set:{ "b.8" : "cheese" }}`

Oplog and Replication: Exercise

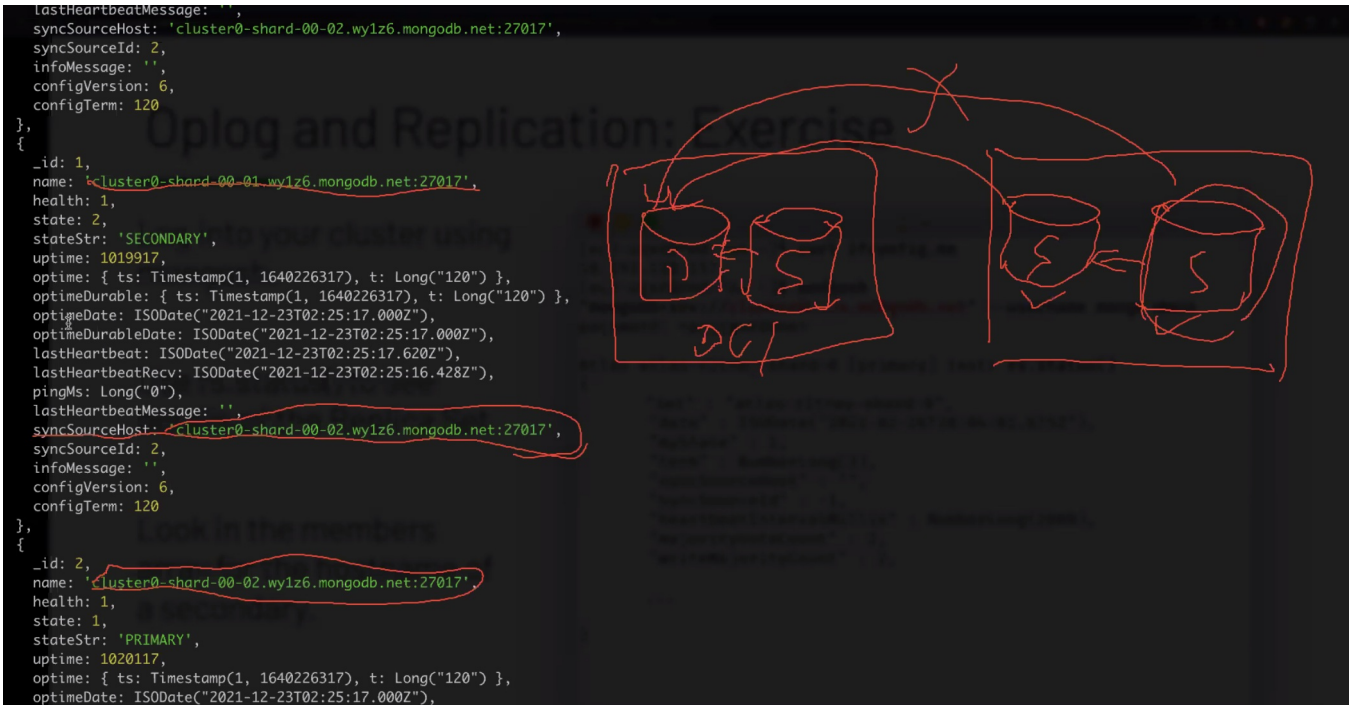
Log into your cluster using mongosh.

Use rs.status() to see details of the Replica Set

Look in the members array for the hostname of a secondary.

```
[ec2-user@red-fox ~]$ curl ifconfig.me
18.193.118.157
[ec2-user@red-fox ~]$ mongosh
"mongodb+srv://cluster0.xxx.mongodb.net" --username mongoadmin
password: <passwordone>

Atlas atlas-r2tney-shard-0 [primary] test> rs.status()
{
        "set" : "atlas-r2tney-shard-0",
        "date" : ISODate("2021-02-26T10:04:01.675Z"),
        "myState" : 1,
        "term" : NumberLong(3),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,

        ...

}
```

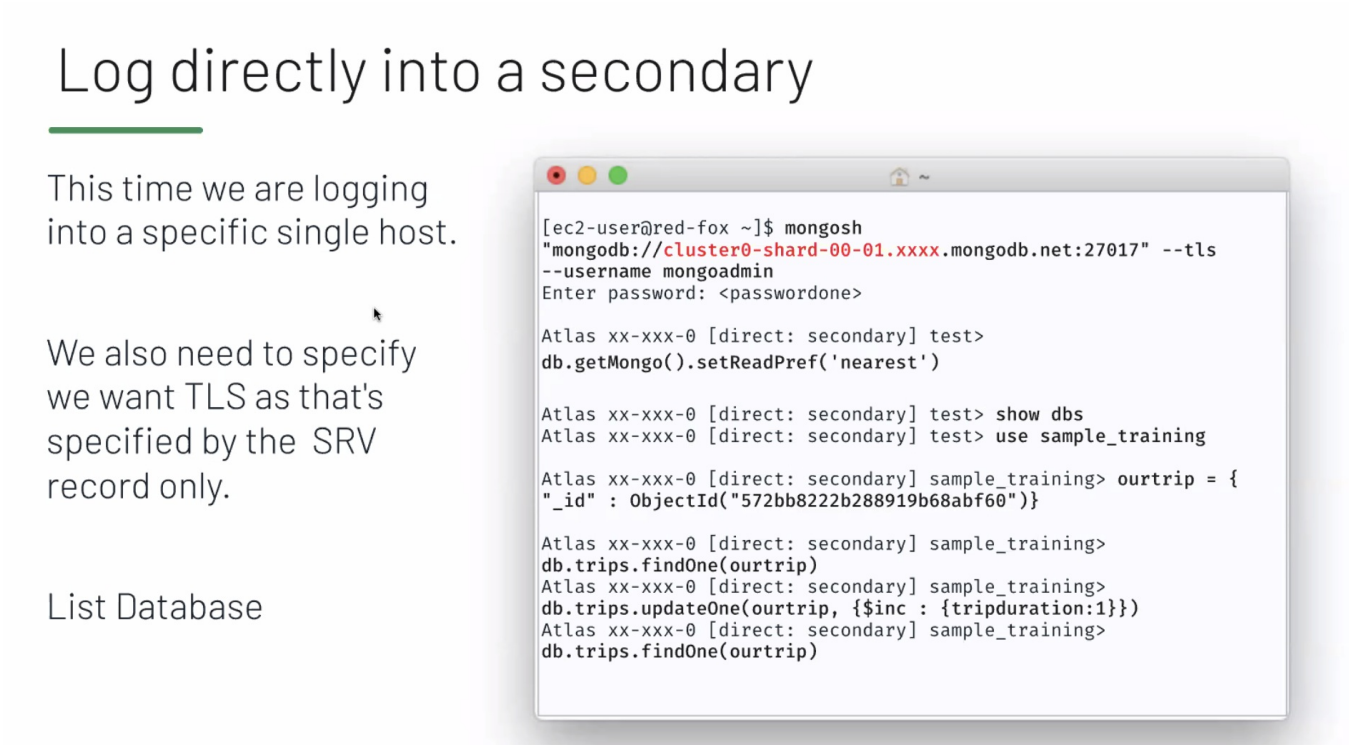
節點從最近節點複製



我們可以直接連線至從節點

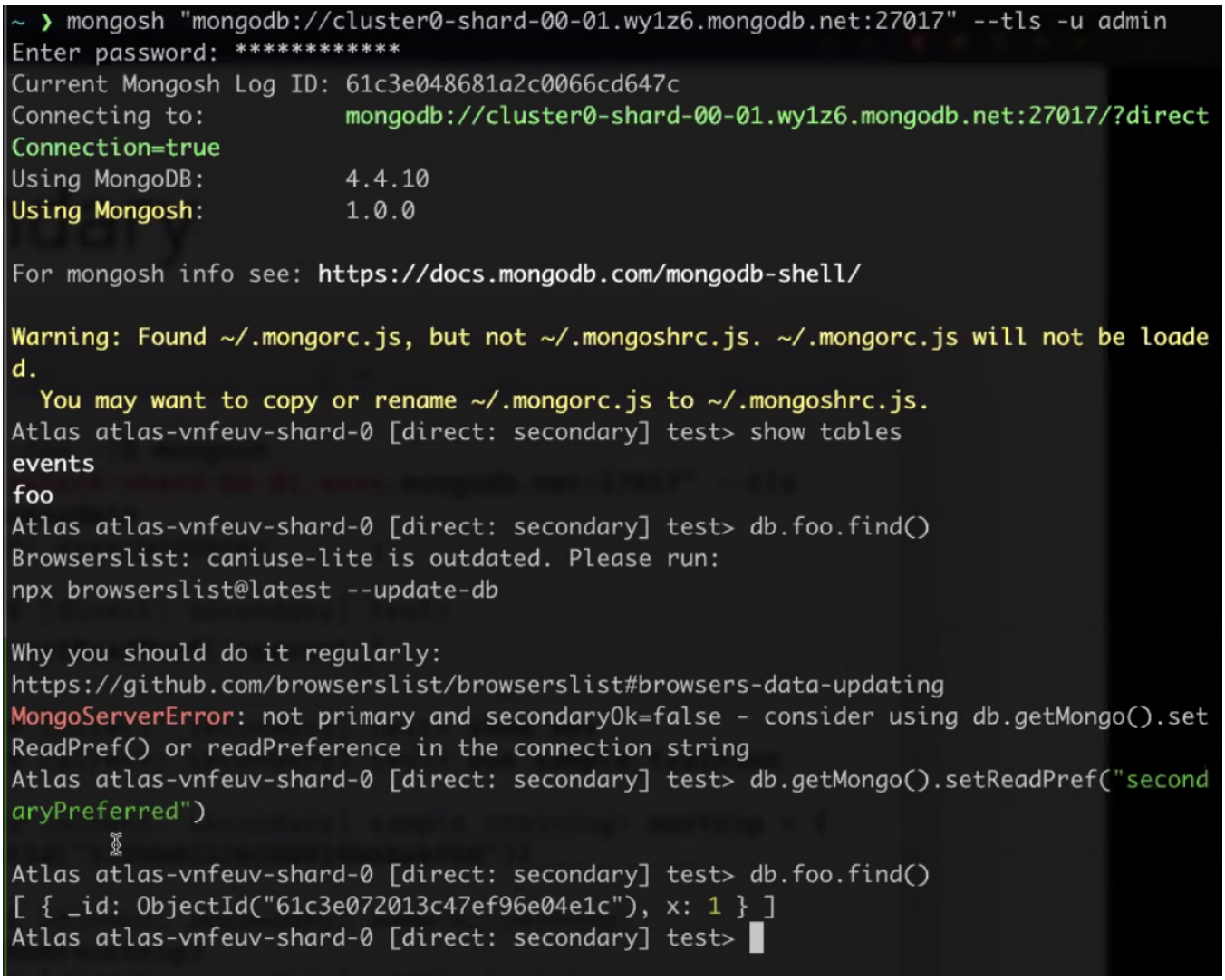
```
Enter password: ************
Current Mongosh Log ID: 61c3df914eeb6d2bd3b539da
Connecting to:          mongodb://cluster0-shard-00-01.wy1z6.mongodb.net:27017/?directConnection=true
Using MongoDB:          4.4.10
Using Mongosh:          1.0.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
  You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
Atlas atlas-vnfeuv-shard-0 [direct: secondary] test>
```

```
Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
  You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
Atlas atlas-vnfeuv-shard-0 [direct: secondary] test> show dbs
sample_airbnb       54.5 MB
sample_analytics    9.38 MB
sample_geospatial    1.7 MB
sample_mflix        47.8 MB
sample_restaurants  7.31 MB
sample_supplies     1.02 MB
sample_training     56.2 MB
sample_weatherdata  3.65 MB
test                65.5 kB
admin                340 kB
local               4.41 GB
```



從節點查詢資料會出現告警，需下以下指令

db.getMongo().setReadPref("secondaryPreferred")



The Oplog Window

- Oplogs are capped collections.
- Capped collections are fixed-size in bytes.
- They guarantee the preservation of insertion order.
- They support high-throughput operations.
- Like circular buffers, once a collection fills its allocated space:
  - It makes room for new documents.
  - By deleting the oldest documents in the collection.

Sizing the Oplog

- The oplog should be sized to account for latency among members. +48h
- The default size oplog is usually sufficient.
- But you want to make sure that your oplog is large enough:
  - So that the oplog window is large enough to support replication
  - To give you a large enough history for any diagnostics you might wish to run.

Elections - The Simple Version

- A secondary determines:
  - It has not heard from the Primary recently.
  - There may be a primary running, but it's not getting through
  - It can contact a majority of secondaries, including itself
  - It can vote and is not specifically ineligible.

- The secondary will then propose an election - with itself as Primary
  - It will state its latest transaction time; it will also state the election term
  - The term goes up by one every election.
  - It will vote for itself by default.

When does a primary step down?

- A Primary will become a secondary when:
  - It sees an election happening that's later than the one it was elected in.
  - You explicitly tell it to step down - it does a good handover
  - It can no longer see a majority of the secondaries

In Summary

- There are multiple servers with the same data.
- Data is durable when it's on a majority of voting servers.
- Data takes time to propagate (normally milliseconds).
- There can be non-voting servers for extra tasks, not HA.

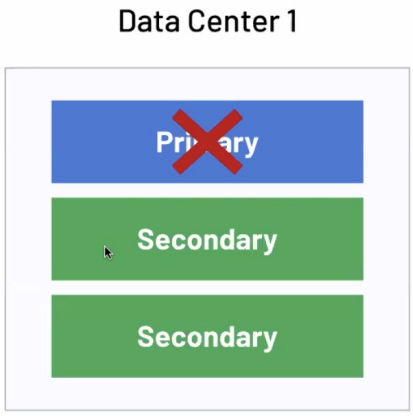
Exercise

As a class - we are going to look at what happens when networks break.

For each of the next five slides we will see which host, if any will be the new primary as well as asking some other questions.

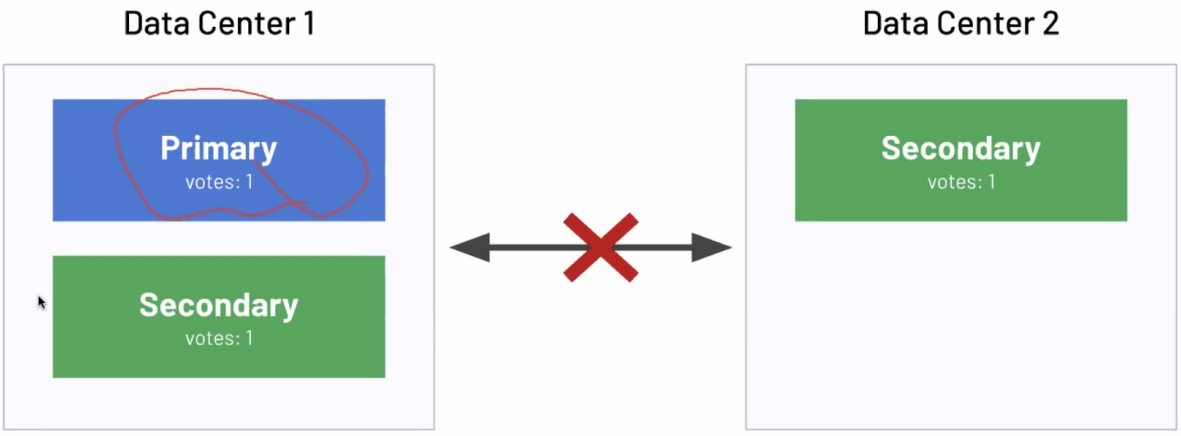
總節點是奇數個才能選出主節點，剩下最新的會變成主節點

3 Nodes, 1 data center



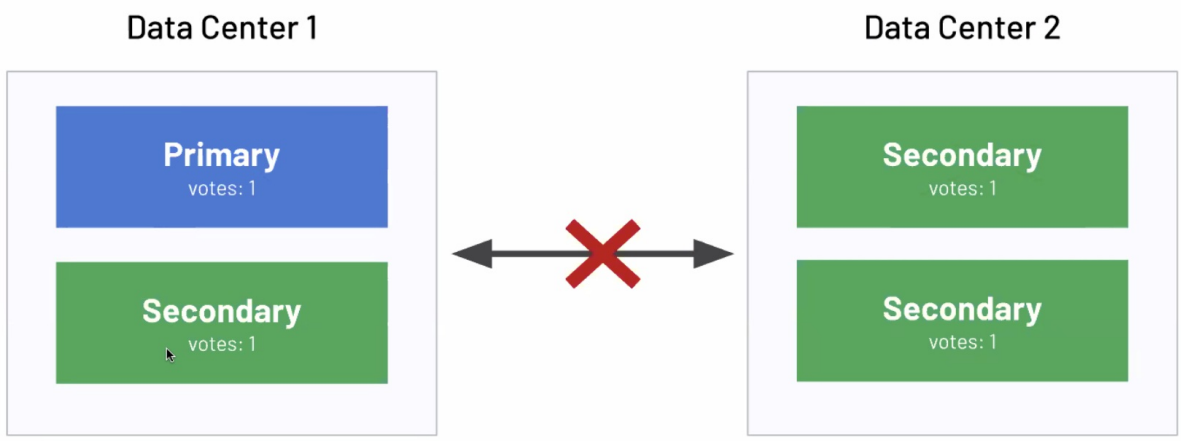
網路斷開主節點還是存在

3 Nodes, 2 data centers



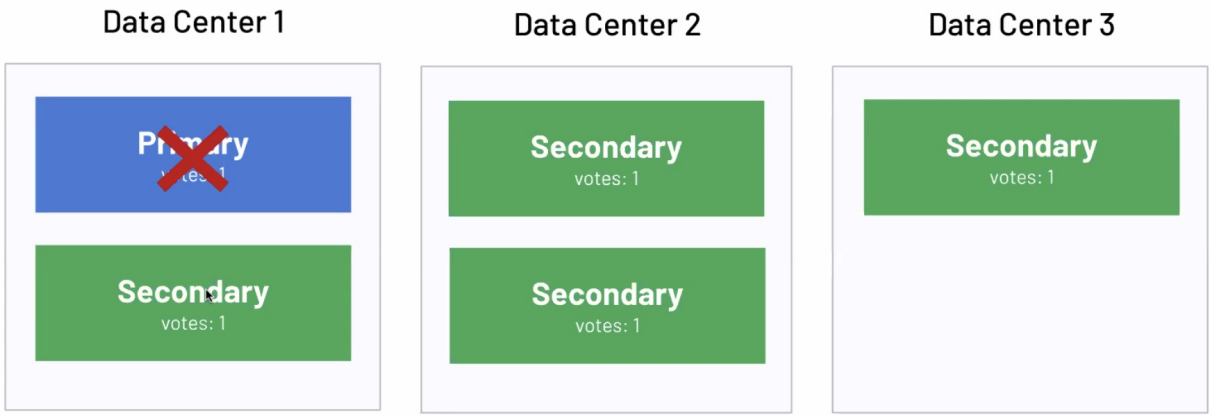
需要有大多數才能投票，此主節點失效

4 Nodes, 2 data centers



會可以投票主節點，DC1有機會成為主節點(資料較新？)

5 Nodes, 3 data centers



Developers have responsibilities too

- Replication doesn't seem like a developer issue
  - Developers need to understand the implications
  - As the software must support it correctly.
  - And make decisions with business owners about speed versus durability.
  - In any multi-server system, safety, speed, and correctness must be considered.

★由 treeman 建立於 23 ◆Q◆G◆◆ 2021 17:56:23
✎由 treeman 更新於 5 ◆◆◆◆ 2023 10:14:59