MongoDB Replication

Derick Rethans

derick@mongodb.com – @derickr

https://joind.in/13409

Replication

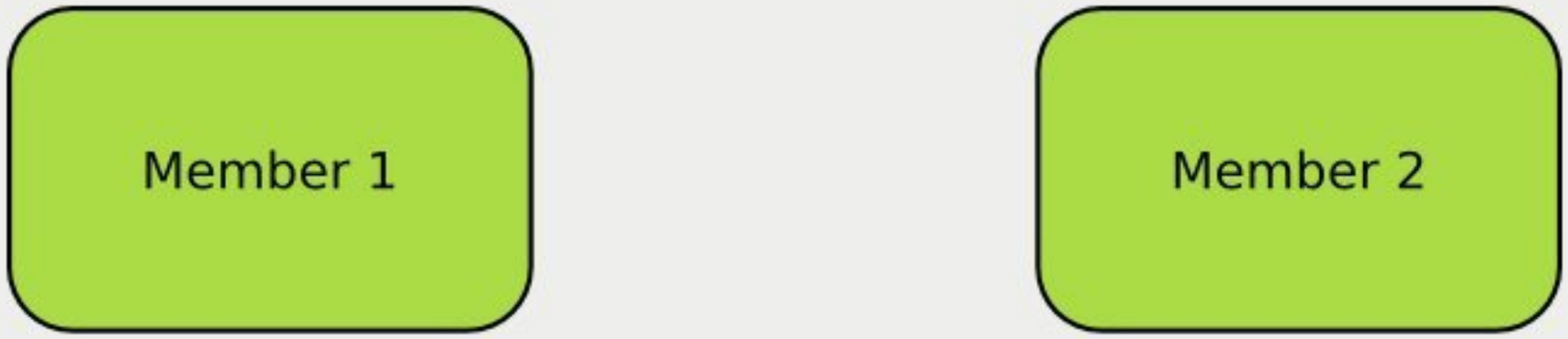
Use cases

- High Availability (auto-failover)
- Read Scaling (extra copies to read from)
- Backups
  - Online, Delayed Copy (fat finger)
  - Point in Time (PiT) backups
- Use (hidden) replica for secondary workload
  - Analytics
  - Data-processing
  - Integration with external systems (f.e. Solr)

MongoDB Replicaset features

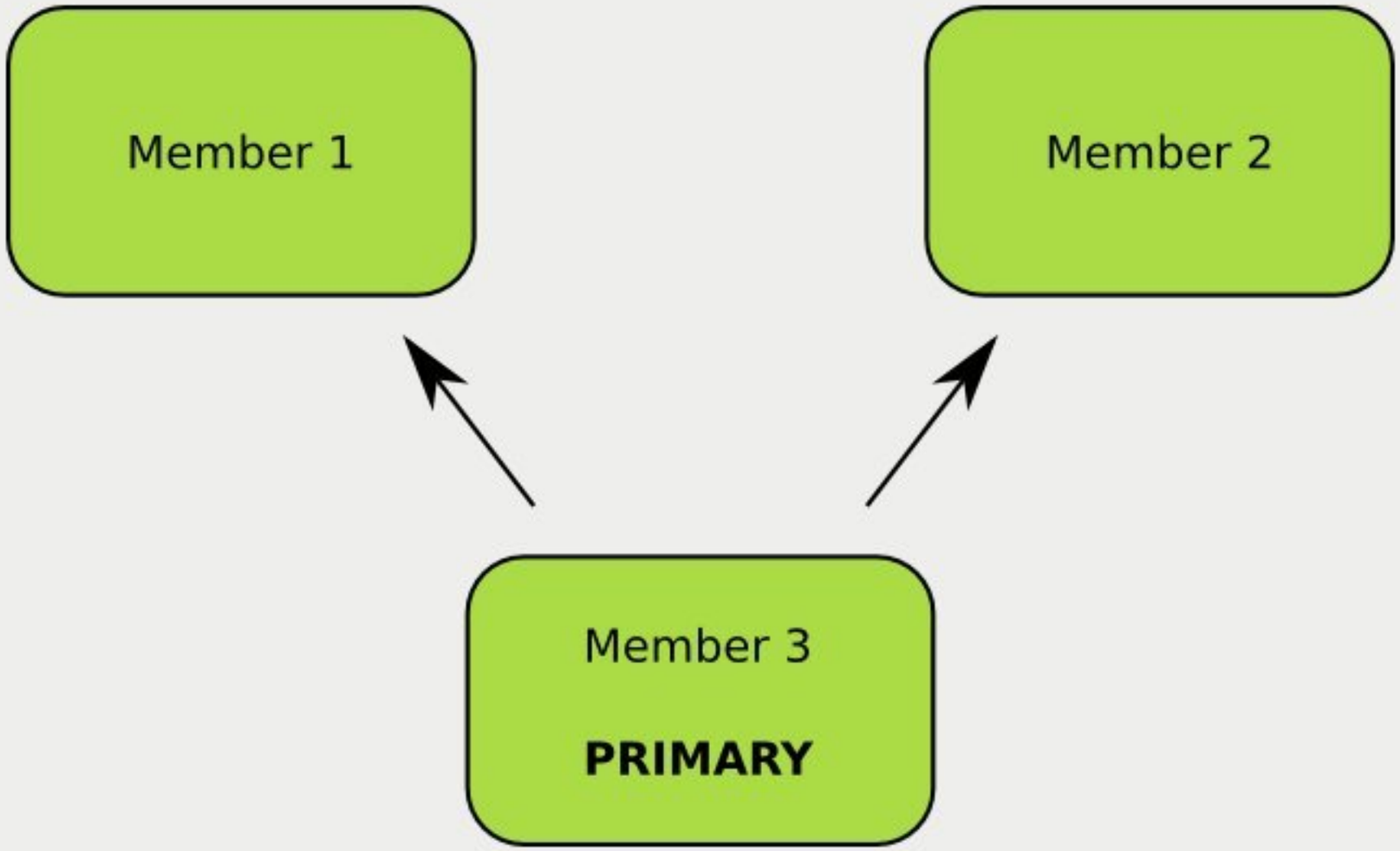
- A cluster of N servers
- All writes to primary
- Reads can be from primary (default) or a secondary
- Any (one) node can be primary
- Consensus election of primary
- Automatic failover
- Automatic recovery

How MongoDB replication works



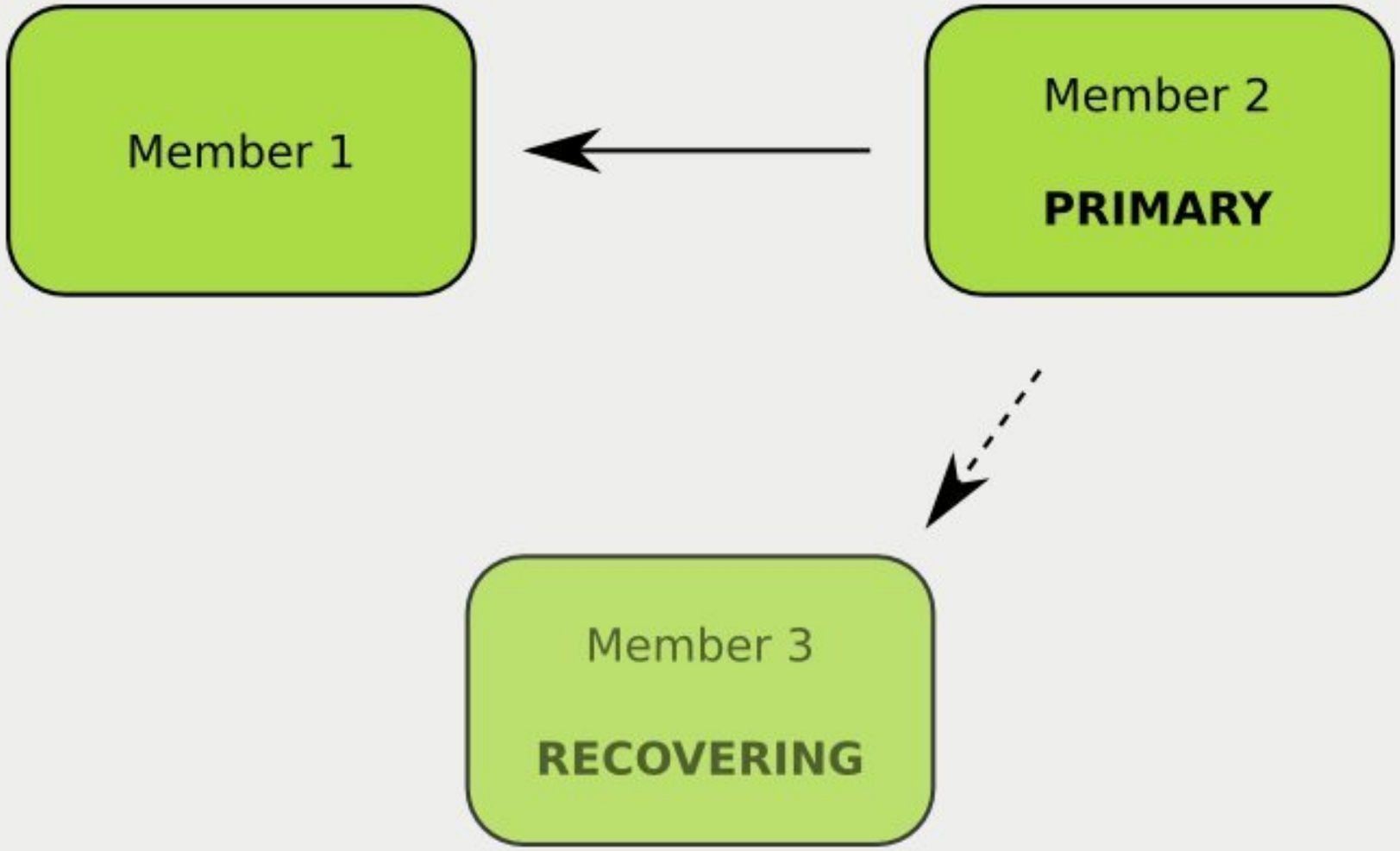
- Set is made up of 2 or more nodes
- It makes most sense to have an **odd** number of nodes

How MongoDB replication works



- Election establishes the PRIMARY
- Data replication from PRIMARY to SECONDARY

How MongoDB replication works



- Automatic recovery

How does replication work?

- Change operations are written to the oplog
  - The oplog is a capped collection (fixed size)
    - Must have enough space to allow new secondaries to catch up after copying from a primary
    - Must have enough space to cope with any applicable slaveDelay
- Secondaries query the primary's oplog and apply what they find
- All replicas contain an oplog

Setting up replication

Starting the daemons (2 data, 1 arbiter):

```
mongod -f /etc/mongodb.conf --dbpath=~/repl-slave0 --port 13000 --replSet seta
mongod -f /etc/mongodb.conf --dbpath=~/repl-slave1 --port 13001 --replSet seta
mongod -f /etc/mongodb.conf --dbpath=~/repl-arb --port 13002 --replSet seta --rest
```

Configure the set:

```
mongo whisky:13000

> db.adminCommand( {
  replSetInitiate: {
    _id: 'seta',
    members: [
     { _id: 0, host: 'whisky:13000', tags: { 'dc': 'west', 'use': 'accounting' } },
     { _id: 1, host: 'whisky:13001', tags: { 'dc': 'east', 'use': 'reporting' } },
    ]
  }
} );

PRIMARY> rs.addArb('whisky:13002');
```

Eventual consistency

- Read Preference / Slave Okay
  - driver will always send writes to Primary
  - driver will send read requests to Secondaries
- Warning!
  - Secondaries may be out of date
  - Not applicable for all applications

Any questions?

https://joind.in/13409

derick@mongodb.com — @derickr