Exploring the replication in MongoDB

Date: Oct-4-2016

Pythian®
love your data

About us

Database Consultant @Pythian

OSDB managed services since 2014

https://tr.linkedin.com/in/okanbuyukyilmaz

Lead Database Consultant @Pythian

OSDB managed services since 2014

https://mk.linkedin.com/in/igorle

@igorLE

Overview

- Why Replication, replication definition
- How replication works, replication concepts
- Replica set features, deployment architectures
- Replica set configuration options
- Key metrics to monitor when running replica set
- MySQL replication compared to MongoDB replication
- Galera replication compared to MongoDB replication

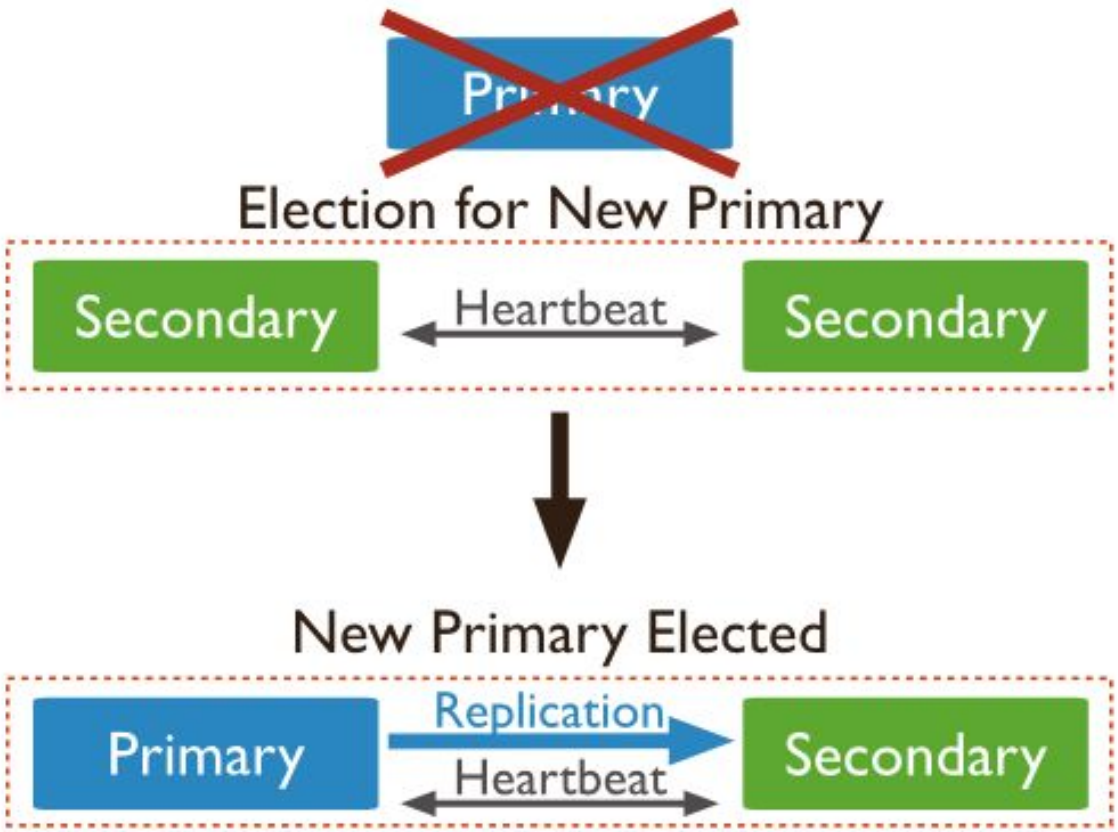
Definition

- Group of mongod processes that maintain the same data set
- Redundancy and high availability
- Increased read capacity

Replication concept

- Write operations go to the Primary node
- All changes are recorded into operations log - oplog
- Asynchronous replication
- Secondaries replicate the Primary oplog
- Secondary can use sync source other Secondary
- One Arbiter node per replica set
- Automatic failover. Any node can become Primary

Replication concept



Replica set oplog

- Special capped collection that keeps a rolling record of all operations that modify the data stored in the databases
- Idempotent
- Default oplog size

For Unix and Windows systems

Storage Engine	Default Oplog Size	Lower Bound	Upper Bound
In-memory	5% of physical memory	50MB	50GB
WiredTiger	5% of free disk space	990MB	50GB
MMAPv1	5% of free disk space	990MB	50GB

Replica set oplog

- Workloads that might require larger oplog size
  - updates to multiple documents at once
  - Deletions equal the same amount of data as inserts
  - Significant number of in-place updates
- Oplog status
  - rs.printReplicationInfo() - on a Secondary node
  - rs.printSlaveReplicationInfo() - on a Primary node
  - db.getReplicationInfo() - by using the local database

Deployment

MongoDB configuration file: /etc/mongod.conf

Before MongoDB 3.0

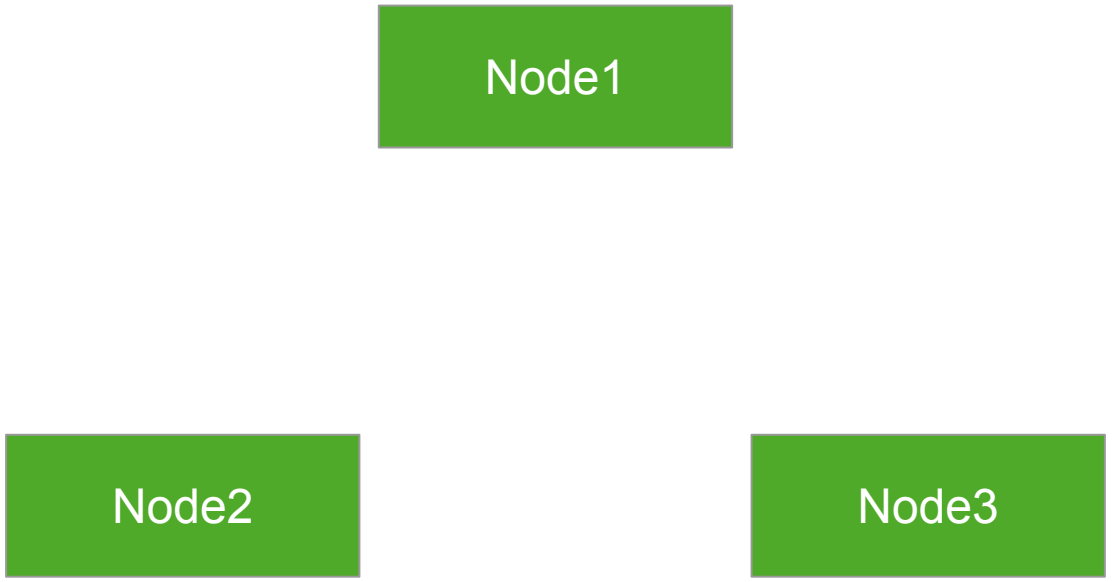
```
replSet=MyRepl
oplogSize=4096MB
```

YAML format since 3.0

```
replication:
    replSetName: MyRepl
    oplogSizeMB: 4096
```

Deployment

- 3 node replica set

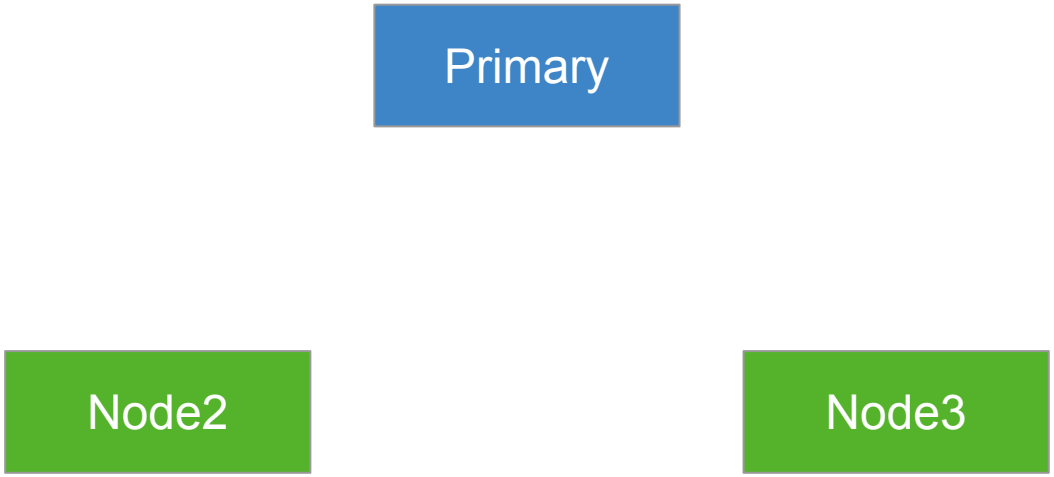


Deployment

Node1

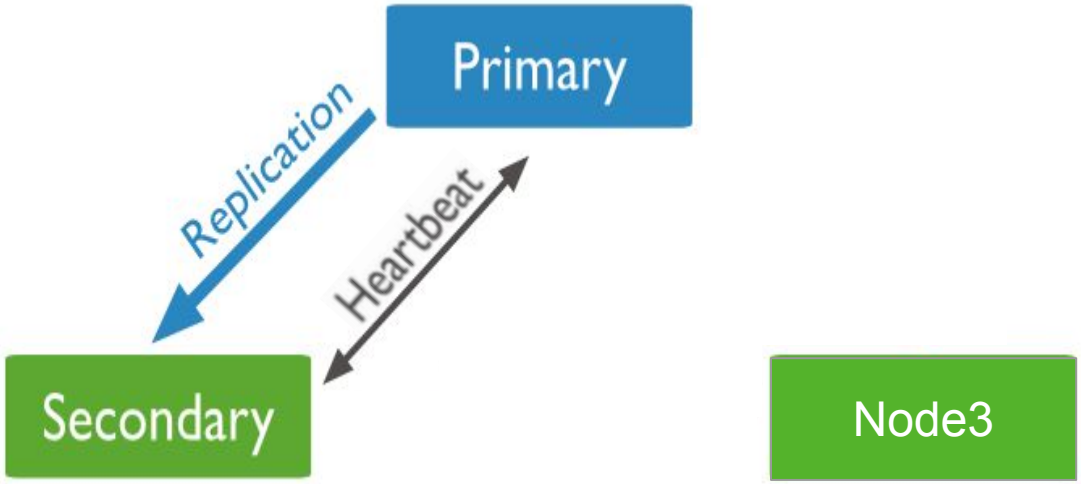
/usr/bin/mongod -f mongod.conf

> rs.initialize()



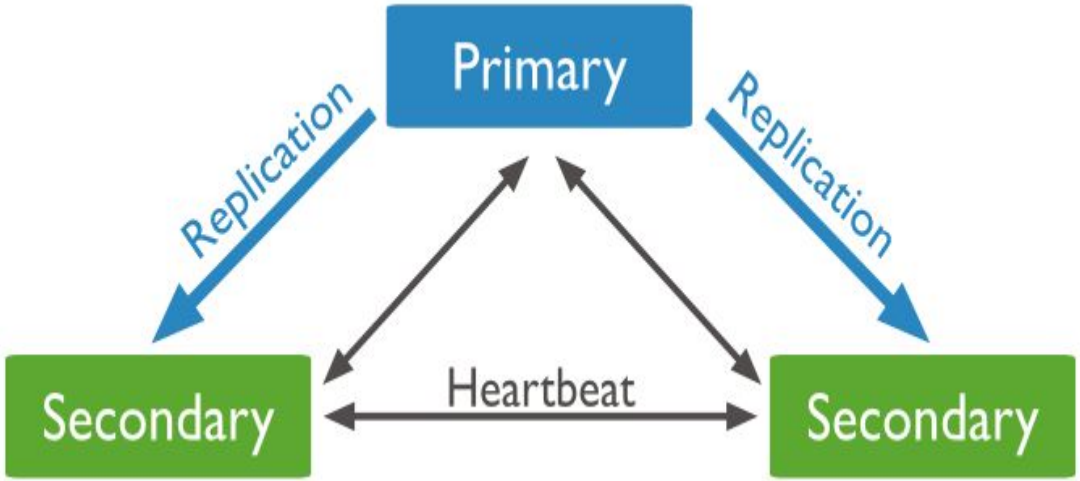
Deployment

- rs.add()
- rs.add(“node2:27017”)



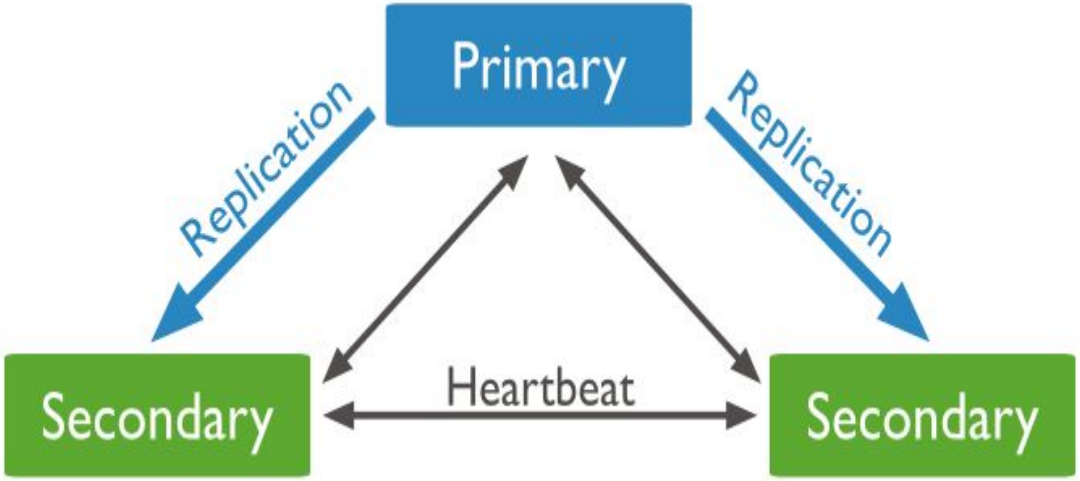
Deployment

- rs.add()
- rs.add(“node3:27017”)



Replica set maintenance

- rs.status()
- rs.conf()
- rs.reconfig()
- rs.remove()
- rs.stepDown()
- rs.freeze()
- rs.isMaster()
- rs.syncFrom()
- rs.printReplicationInfo()
- rs.printSlaveReplicationInfo()



Configuration options

- 50 members per replica set (7 voting members)
- Arbiter node
- Hidden node
- Priority 0 node
- Delayed node
- Build indexes
- Fault tolerance
- Write concern
- Read preference

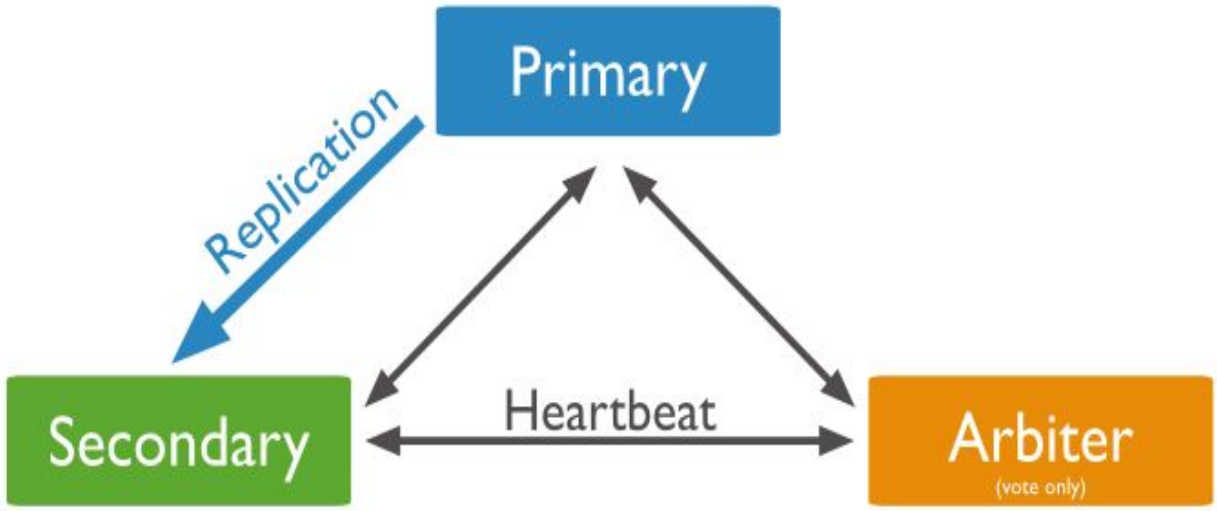
Configuration options

```
rsTest:PRIMARY> rs.conf()
{       "_id" : "rsTest",
        "version" : 2,
        "members" : [
                {
                        "_id" : 0,
                        "host" : "node1:27017",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 1,
                        "tags" : {"dc" : "apac"},
                        "slaveDelay" : 0,
                        "votes" : 1
                },
```

Arbiter node

- Votes in elections
- Does not hold copy of data

```
{
  "_id" : <num>
  "host" : <hostname:port>,
  "arbiterOnly" : true,
….
}
```



Hidden node

- Never becomes primary
- Not visible to application
- Use cases
  - reporting
  - backups

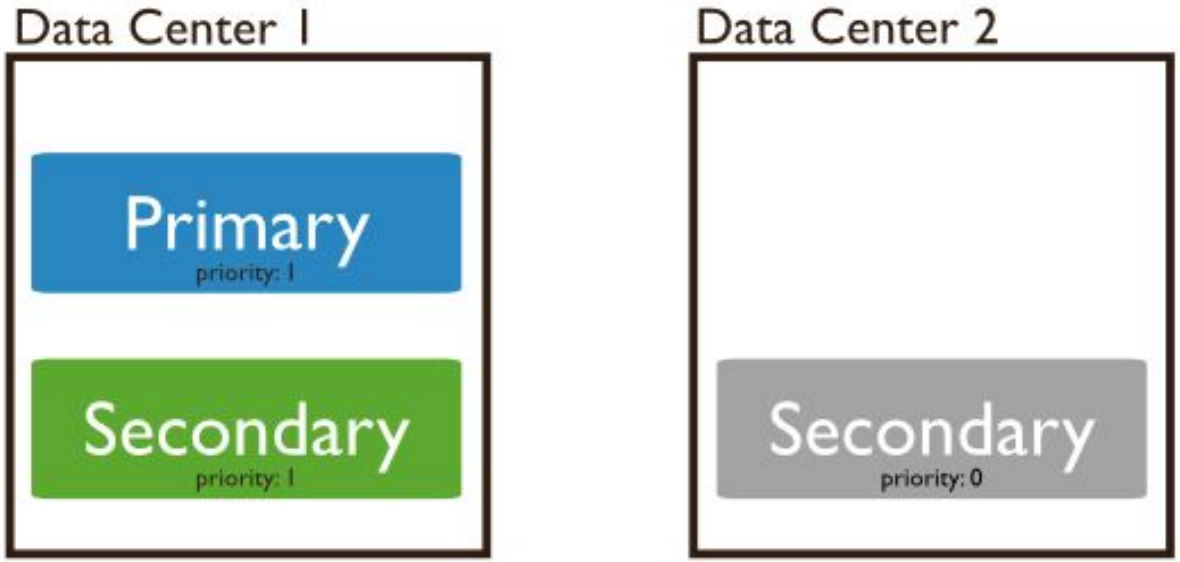
```
{
  "_id" : <num>
  "host" : <hostname:port>,
  "priority" : 0,
  "hidden" : true
}
```

Priority 0 node

Priority - floating point (i.e. decimal) number between 0 and 1000

- Hidden members and Delayed slave with priority 0 by default
- Never becomes primary
- Visible to application

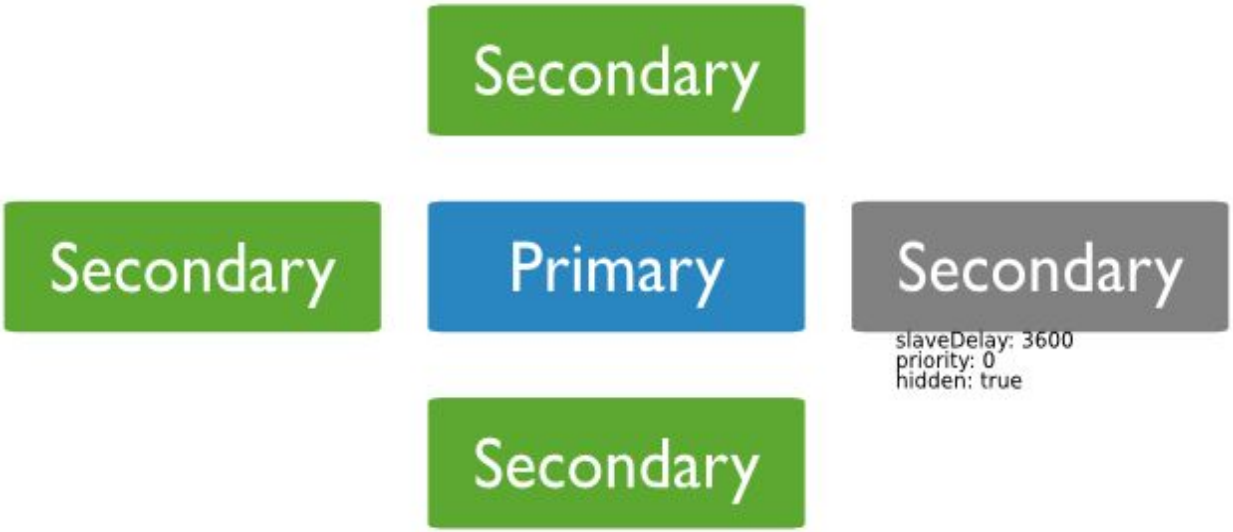
```
{
  "_id" : <num>
  "host" : <hostname:port>,
  "priority" : 0
}
```



Delayed node

- must be priority 0 member
- should be hidden member (not mandatory)
- has votes 1 by default
- considerations (oplog size)
- mainly used for backups

```
{
  "_id" : <num>,
  "host" : <hostname:port>,
  "priority" : 0,
  "slaveDelay" : <seconds>,
  "hidden" : true
}
```



Build indexes

members[n].buildIndexes : false

- Can be only set when adding a member to a replica set
- Can not be changed on running node
- Indexes on _id will still exist
- Priority must be set to 0, recommended as hidden:true

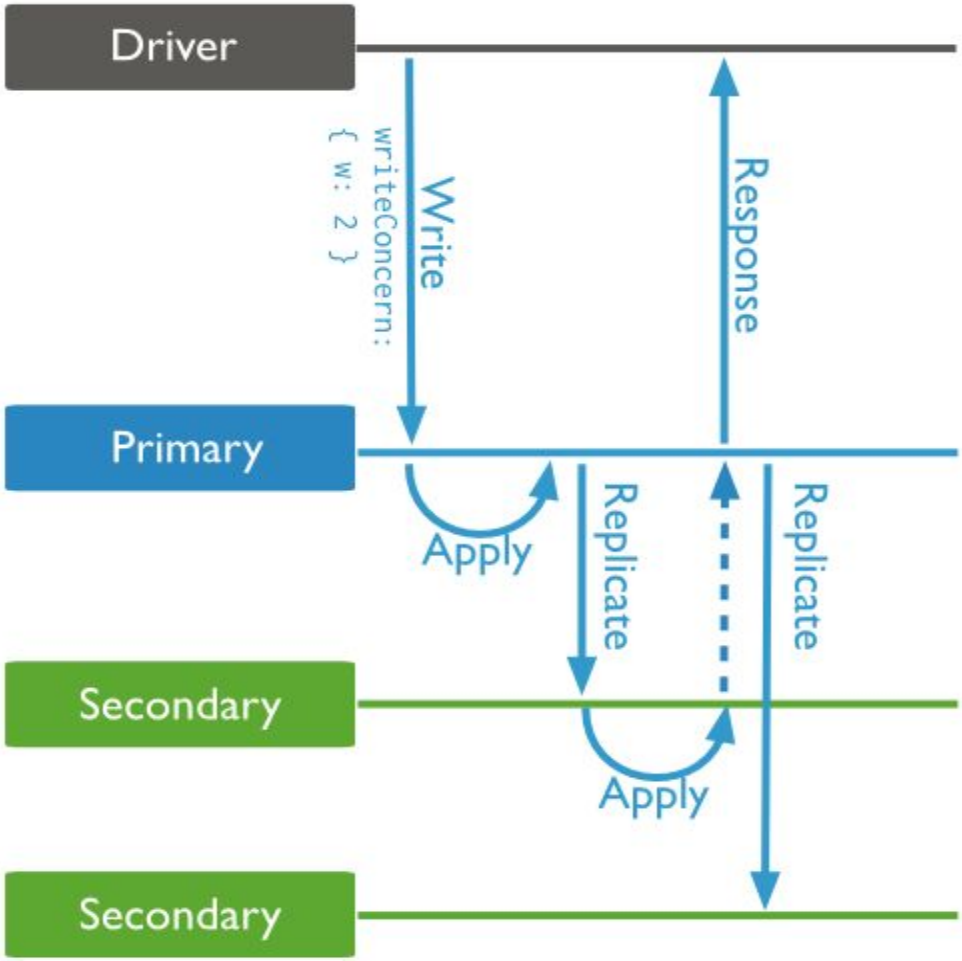
Useful when

- Node is used for backups using mongodump
- Node does not receive queries from clients
- Adding indexes and maintenance overburdens the host system

Fault tolerance

Number of members	Majority Required to Elect new Primary	Fault Tolerance
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3

Write concern



Write concern

Write concern specification

{ w: <value>, j: <boolean>, wtimeout: <number> }

- w - number
  - Option for “majority”
- j - write operation has been written to the journal
- wtimeout - time limit, in milliseconds, for the write concern

Replica set get write concern setting

```
rsTest:PRIMARY> rs.conf().settings['getLastErrorDefaults']
{ "w" : 1, "wtimeout" : 0 }
```

Read preference

- primary
- primaryPreferred
- secondary
- secondaryPreferred
- nearest

Key metrics to monitor for RS

- Replica set has no Primary
- Number of healthy/unhealthy members
- Replication Oplog Window
- Replication lag

What is MySQL

Mysql is…

- Transactional
- Relational
- Well documented
- Open source

MySQL replication

Mysql Replication can be...

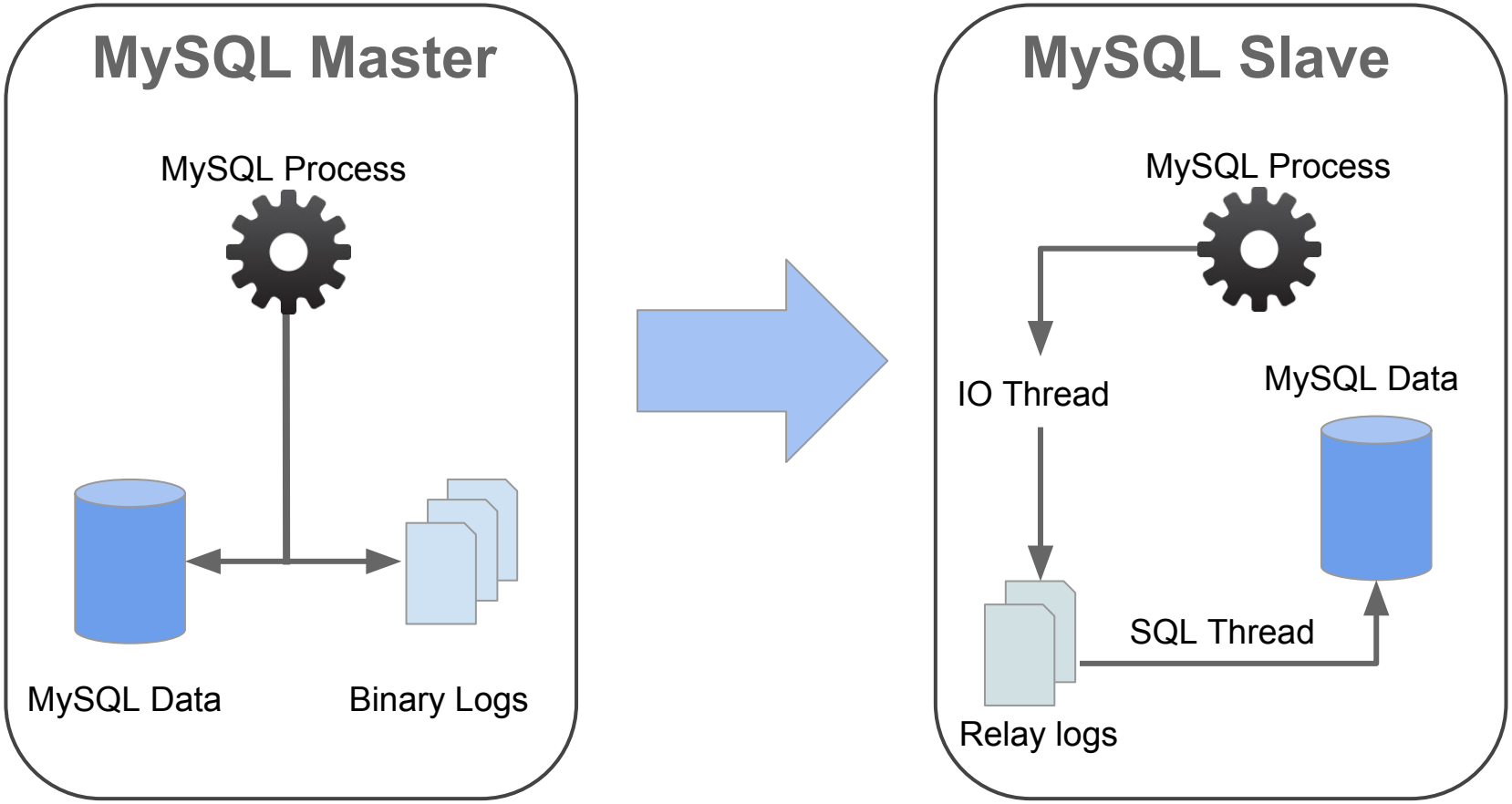
- Asynchronous
  - Built-in replication feature
  - Semi-synchronous by plugin *rpl_semi_sync_master*

- Synchronous
  - Galera setup with wsrep api

MySQL replication

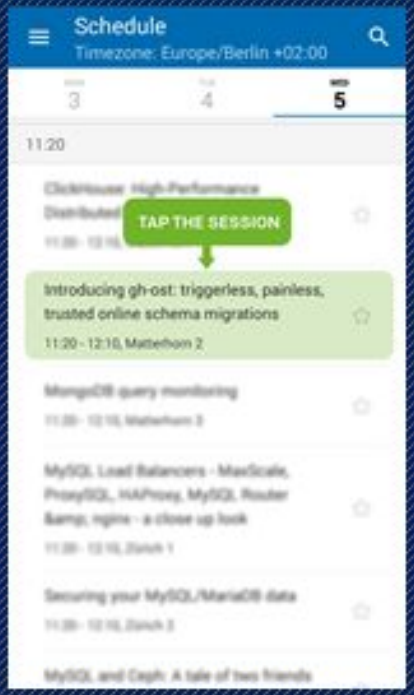
- Group of mysqld processes that maintain the same data set
- Write operations go to the Master node, reads can go to Slave
- All changes are recorded into binary log on the master
- Asynchronous replication, Semi-Synchronous replication
- Slave node replicates the events by copying binary logs as relay logs
  - IO Thread, SQL Thread
  - Binary Logs format - Statement, Row, Mixed
- Expire logs days dynamic
- Replication filters

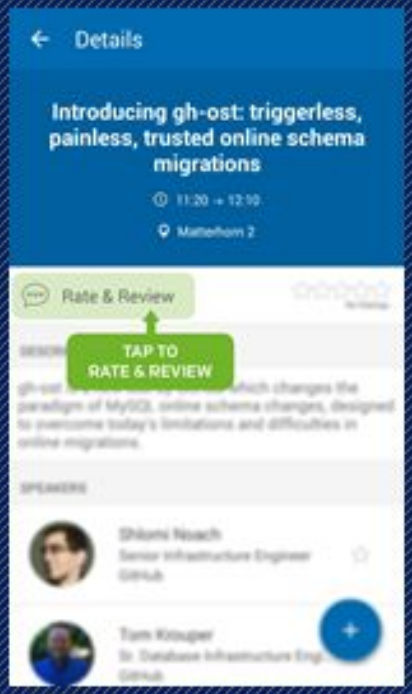
MySQL replication

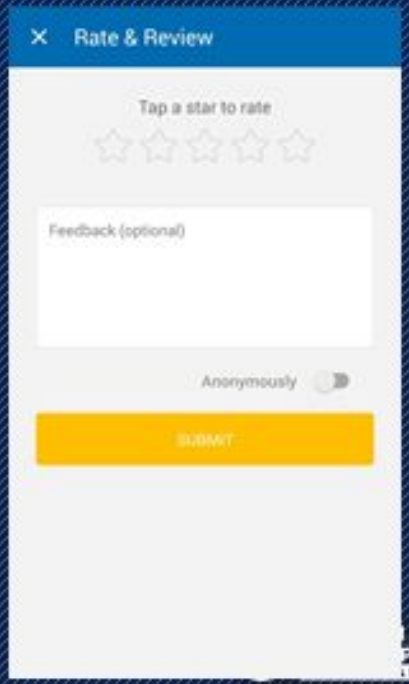


Questions?

Rate My Session!







We’re Hiring!

https://pythian.com