WEB335 Discussion 8.1 – Replication

What are replica sets?

A replica set is a group of database instances that all have the same data set[[1]](#footnote-1). The purpose of a replica set is to protect the data on the database against problems such as hardware failure, interruptions of service, or downtime for maintenance, as well as to increase the availability of data. A database instance is basically a node, which would usually be a specific server for the database. So replica sets basically involves keeping copies of the same data on multiple servers. Within the group of servers or nodes, one particular node becomes the primary, which means it handles all write operations, while the other nodes, called secondaries, simply copy the data from the primary and handle only read operations.

How does replica set members replicate new data?

Replication of the data involves the use of an operation log (oplog)[[2]](#footnote-2). When the primary receives a change in data, it records the changes in its oplog. The secondaries then replicate the primary’s oplog and applies the operations in the oplog to their own data sets so that theirs reflects the primary’s. This is done in an asynchronous process[[3]](#footnote-3).

How are failures and rollbacks managed?

A failure occurs when a primary cannot communicate with the client and/or the secondaries. When that happens, the secondaries hold an election to pick one secondary to become the new primary. No writes can be processed until a new primary is elected, although read operations can continue. Once the original primary is back online, it will join the group again as a secondary.

If any writes had been accepted by the original primary before its failure that were not replicated to the secondaries, the primary will roll back those changes so that its database matches the secondaries. It does not occur if the write operations made it to one of the secondaries and that secondary remains available and accessible to a majority of the replica set. The rollbacks are saved in a BSON file that administrators can access to make sure the writes make it back into the database if the administrator so chooses.

1. MongoDB – Replication (tutorial) (no author). (n.d.). Retrieved from <https://www.tutorialspoint.com/mongodb/mongodb_replication.htm> [↑](#footnote-ref-1)
2. MongoDB: Replication (article). (no author). (n.d.). Retrieved from <https://docs.mongodb.com/manual/replication/> [↑](#footnote-ref-2)
3. MongoDB Replication (article). (no author). (n.d.). Retrieved from <https://kubedb.com/docs/0.12.0/guides/mongodb/clustering/replication_concept/> [↑](#footnote-ref-3)