

My experience With MongoDB cluster

Generally, we use the master-slave architecture for MongoDB.

I worked on one-master-multiple-slave structure. As it is advisable to have an odd number of nodes in a cluster that helps in electing the master when the master node is down. I personally worked on a cluster of three nodes, one master and two slaves.

Where the master node can both read and write data. The op-log will synchronize the updates to all the connected slave nodes.

The slave node can only read data, but not write data. It automatically synchronizes data from the master node.

The replica set consists of MongoDB instances having the same data. It has the following roles:

The primary node receives all the write requests and then it synchronizes the changes to all the secondary nodes.

A replica set can only have one primary node. When the primary fails, the other secondary nodes or the arbiter node will re-elect a primary node.

The primary node receives read requests for processing by default.

The service will remain unaffected if the secondary node goes down, but if the primary node goes down, the system will initiate a re-election of the primary node.

To make sure that a production cluster has no single point of failure, it is important to provide a shared cluster for high availability.