**MongoDB Replication**

**Replication -** The replication is the process to synchronizing data across multiple servers. Replication provides redundancy and increase high availability with multiple copies of data on different database servers. Replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions. With additional copies of the data, you can dedicate one to disaster recovery, reporting, or backup.

There are some reasons to maintains the Data Replication

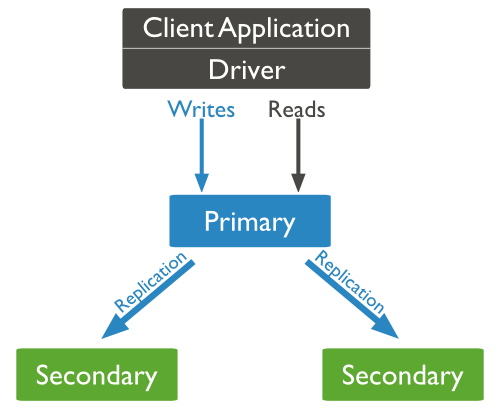
1. To keep data safe.
2. High availability
3. Disaster recovery
4. No downtime for maintenance (like backups, rebuild index, compaction)
5. Replica set is transparent to the application.

**How replication works**

MongoDB achieves replication by using replica set. The **Replica Set** is a group of **mongod** instances that keeps the same data set. In replica set, one node is primary node that receives all write operations. All other instances that is known as secondary nodes. The primary records all changes to its data sets in its operation log, i.e. oplog. The secondaries replicate the primary’s oplog and apply the operations to their data sets such that the secondaries data sets reflect the primary’s data set. If the primary is unavailable, an eligible secondary will hold an election to elect itself the new primary.

1. The replica set is group of two or more nodes.
2. In replica set, one node is primary, others nodes are secondaries
3. All data replicate from primary to secondary.
4. At the time of automatic failover or maintenance, election establishes for primary and a new primary node is elected.
5. After the recovery of failed node, it again join the replica set and works as a secondary node.

The following pictures shows the replication



**Setting Up Replica Set Syntax and commands**

**Steps to setup replica set.** To convert all ready running mongodb to replica set we will follows the given steps

Step 1- Shutdown already mongoDB server.

Step 2- Start mongdb server by specifying –replSet option

mongod --port "PORT" --dbpath "YOUR\_DB\_DATA\_PATH" --replSet "REPLICA\_SET\_INSTANCE\_NAME"

For Example :

mongod --port 27017 --dbpath "D:\set up\mongodb\data" --replSet rs0

1. It will start a mongod instance with name rs0, on port 27017
2. Now start the command prompt and connect to this mongod instance.
3. In Mongo client, issue the command rs.initiate() to initiate a new replica set.
4. To check the replica set configuration, issue the command rs.conf(). To check the status of replica set issue the command rs.status().

**Add Members to Replica Set**

To add members to replica set, start mongod instances on multiple machines. Now start a mongo client and issue a command rs.add().

Syntax :

rs.add(HOSTNAME:PORT);

For Example :

Suppose your instance name is pcSap1 and running mongod port is 27017

rs.add("pcSap1:27017");

You can add mongod instance to replica set only when you are connected to primary node.

To check whether you are connected to primary or not, issue the command db.isMaster() in mongo client.

**How to Setup Replica on windows machine with Example :**

1. Start standalone server as shown below.

"C:\Program Files\MongoDB\Server\3.0\bin\mongod.exe" --config "C:\Program Files\MongoDB\mongod.cfg" --journal --serviceName MongoDB --serviceDisplayName MongoDB –install

Where mongod.cfg contains the following lines

dbpath = E:\MongoData

logpath = E:\MongoData\mongo.log

logappend = true

#port number

port=27017

#replica set name

replSet=rs1

1. Connect to the server with port number 27017

mongo --port 27017

1. Then, create variable rsconf

rsconf={\_id:"rs1",members:[{\_id:0,host:"localhost:27017"}]}

rs.initiate(rsconf);

1. Start secondary server on the port 27018
2. "C:\Program Files\MongoDB\Server\3.0\bin\mongod.exe" --config "C:\Program Files\MongoDB1\mongod1.cfg" --journal --serviceName MongoDB1 --serviceDisplayName MongoDB1 –install

The mongod1.cfg file contains the lines

dbpath = E:\MongoData1

logpath = E:\MongoData1\mongo1.log

logappend = true

#port number

port=27018

#replica set name

replSet=rs1

1. Logon to secondary server

mongo --port 27018

1. Run the following commands on Primary server

rs.add("localhost:27017");

rs.add("localhost:27018");

1. Now go to secondary servers and run below command on both the secondary servers

rs.slaveOk();

**How we can verify replica set**

1. When we connect with primary it shows the following prompt

mongo –port 27017  
rs1:PRIMARY>

1. Then run use mydb;

rs1:PRIMARY>use mydb;

When you insert any records in book collection. For example

It will reflect in secondary

1. Now connect with secondary

mongo –port 27018

It shows

rs1:SECONDARY>

then run the use mydb

and run the following command to view the inserted records by primary.

db.book.find({});