mongoDB www.mongodb.org @mongodb

Replica sets

http://www.mongodb.org/display/DOCS/Replica+Sets

Replica sets are basically master/slave replication with automatic failover. The current master is called the ***primary*** and the servers syncing off of it are called ***secondaries***. If a primary becomes unreachable, a secondary may be elected to be primary without human intervention.

A majority of servers must be able to reach the primary, both to elect it and to keep it a primary.

What is a majority?

If your set consists of:
1 server, 1 server is a majority
2 servers, 2 servers are a majority
3 servers, 2 servers are a majority
4 servers, 3 servers are a majority
...

If you have two servers and you want one server to be elected primary if the other goes down, you can use an *arbiter*, a lightweight *mongod* process that only participates in elections. You need at most 1 arbiter.

Config Format

```
{
    "_id" : "nameOfSet"
    "members" : [
        {"_id" : 0, "host" : "host0"},
        {"_id" : 1, "host" : "host1"},
        ...
        {"_id" : n, "host" : "hostn"}
    ]
}
```

Member objects can also contain:

- `priority : n` — **Members will be elected primary in order of priority, if possible. `n=0` means this member will never be master**
- `slaveDelay : n` — **This member will always be a secondary and will lag *n* seconds behind the master**
- `arbiterOnly : true` — **This member will be an arbiter**
- `hidden : true` — **Do not show this member in `isMaster` output**
- `tags : [...]` — **Member location description, see http://www.mongodb.org/display/DOCS/Data+Center+Awareness**

Shell Helpers

`rs.initiate()` — Creates a new replica set with one member

`rs.add("host:port")` — Adds a member

`rs.add({"_id" : n, "host" : "host:port", slaveDelay : seconds})` — Adds a member with non-default properties

`rs.addArb("host:port")` — Adds an arbiter

`rs.remove("host:port")` — Removes a member

Administration

Please review the full documentation for more details about administration of replica sets.

`rs.status()` — See the status of each member

`rs.conf()` — See the current set configuration

`rs.reconfig(newConfig)` — Change the set configuration

`rs.isMaster()` — See which member is primary

`rs.stepDown(n)` — Force the primary to become a secondary for *n* seconds

`rs.freeze(n)` — Prevent a secondary from becoming the primary for *n* seconds. (*n* = 0 means unfreeze.)

Sharding

To add a replica set as a shard, run:

```
db.adminCommand({addShard :"setName/host:port[,host2:port2]"})
```

mongos will use the given host(s) as seeds and find the rest of the set. It will also automatically detect when a replica set's members change.

Tips

- Run one secondary with journaling enabled to ensure that you always have a clean copy of your data available.
- Start replica set members with the *--rest* option for useful information in the web interface.