

Mongo Clusters

Mongo can be used in many ways. In our group so far we have been using mongo as a standalone server. This page will describe some of the results we have had with setting up and using a mongo cluster. The two methods to setup clusters (that can be combined) are replication clusters as well as sharded clusters.

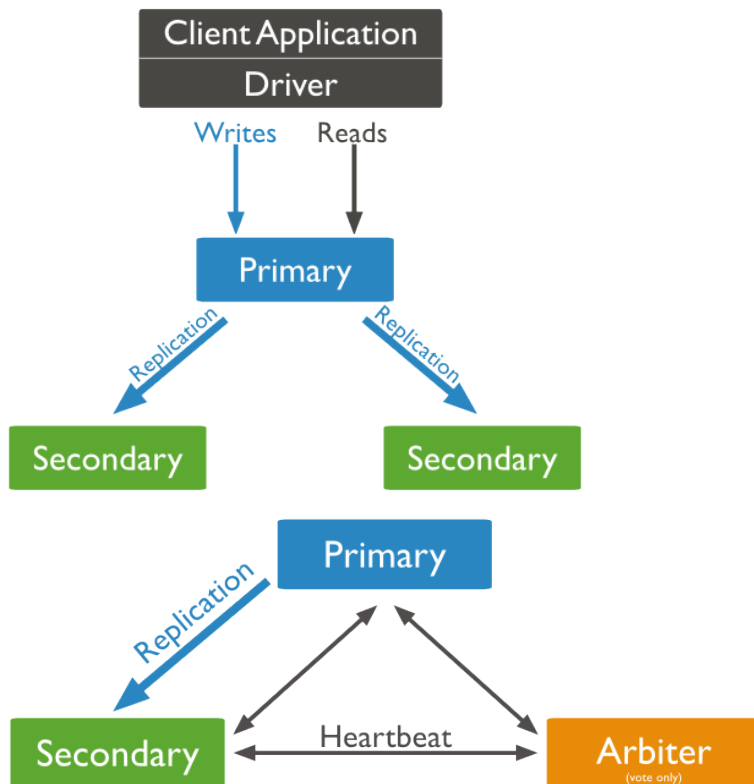
Replicated Clusters

A replicated cluster is where a set of mongo servers have replicated data. This will allow for fault tolerance. Any time a node of the cluster disappears another node can take over being the primary node. This can also improve the read performance (since reads can be done on multiple nodes). For more information see <https://docs.mongodb.com/manual/replication/>.

To get this to work in clowder, add the following to your custom.conf file:

```
mongodbURI = "mongodb://mongo-1.os.ncsa.edu:27017,mongo-2.os.ncsa.edu:27017,mongo-3.os.ncsa.edu:27017/sead2-dev?replicaSet=ISDA"
```

You don't need to specify all nodes in the cluster, however if the one you specify is down, you will not be able to get the other nodes in the cluster.



A replicated system should have a minimum of 3 nodes. If you have only two nodes in the cluster and one disappears, the second node will NOT be selected the new primary node, and your cluster will not be reachable. One of these nodes can be an Arbiter (an arbiter is a regular mongo instance, but does not store any data). Once one of the nodes disappears the left over nodes will vote on who will become the next primary. Normally each node will have the same priority, but this can be changed during the setup.

To create the cluster the following changes had to be made to /etc/mongod.conf

- Ubuntu will bind mongo to localhost interface only. To undo this make sure to remove the "bindIp" element
- You will need to add the replication name, this is done in the replication: section by setting replSet
- To have all databases go into separate directories add directoryPerDB: true to the storage: section

Following is the mongod.conf that is used:

/etc/mongod.conf

```
# mongod.conf
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
  journal:
    enabled: true
  directoryPerDB: true

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# network interfaces
net:
  port: 27017

# replication options
replication:
  replSetName: ISDA
```

Following are the commands used to create the cluster.

```
# To initialize the cluster (can be done on any node), after
# this command you will have a cluster with 2 nodes.
rs.initiate({
  _id: "ISDA",
  version: 1,
  members: [
    { _id: 0, host : "mongo-1.os.ncsa.edu:27017" },
    { _id: 1, host : "mongo-2.os.ncsa.edu:27017" }
  ]
})

# To add a new node to the cluster, this will start a copy. You
# can copy the mongod folder from one of the nodes before you
# add the new node to the cluster
rs.add("mongo-3.os.ncsa.edu:27017")

# To remove a node from the cluster
rs.remove("mongo-3.os.ncsa.edu:27017")

# To add an arbiter to the cluster
rs.addArb("mongo-3.os.ncsa.edu:27017")
```

Sharded Cluster