Full backup procedure

1. lock writes on a secondary member db.fsyncLock()
2. Take snapshot
3. Record last position from oplog

db.oplog.rs.find().sort({$natural:-1}).limit(1).next().ts

1. Unlock writes db.fsyncUnlock()

Incremental backup procedure

1. lock writes on a secondary member
2. Dump oplog from the recorded oplog position on full (or latest incremental ) backup:
3. mongodump --host <secondary> -d local -c oplog.rs -o /mnt/mongo-test\_backup/1

--query '{ "ts" : { $gt : Timestamp(1437725201, 50) } }'

1. Record latest oplog position (same way as for full backups)
2. Unlock writes

Full backup restore procedure

1. stop all instances of mongod
2. copy snapshot to data dir of the box which will be the primary, but make sure to exclude all local\* and mongod.lock this restore technique is called [reconfigure by breaking mirror](http://docs.mongodb.org/manual/tutorial/reconfigure-replica-set-with-unavailable-members/#replica-set-reconfigure-by-replacing)
3. Start primary
4. reconfigure replicaset
5. start secondaries without any data, let them perform the initial sync. Or copy the data from the new primary with fresh local database

Restore incremental backup

When we created incremental backup it stored it like this:

/mnt/mongo-test\_backup/1/local/oplog.rs.bson

/mnt/mongo-test\_backup/1/local/oplog.rs.metadata.json

We're instered on oplog.rs.bson but we will have to rename it, so here are the steps:

1. change directory to the backup: cd /mnt/mongo-test\_backup/1/local
2. delete the json file rm \*.json
3. rename the bson file mv oplog.rs.bson oplog.bson
4. restore it:

mongorestore -h <primary> --port <port> --oplogReplay /mnt/mongo-test\_backup/1/local