General Ops Notes

Getting a single instance started

Need a directory for data files:

```
mkdir -p /data/db
```

Start up mongod:

```
mongod(.exe) --dbpath /data/db
```

Things to add in production:

```
--logpath, --logappend, --fork, --rest, ...
```

Use a config file:

```
dbpath = /data/db
logpath = /path/to/logfile
logappend = true
fork = true
rest = true
```

Then use from the command line:

```
mongod(.exe) -f /path/to/config/file
```

Monitoring

Important tools:

```
mongostat (--discover)
iostat -x 2
http://localhost:<1000 greater than --port>
mms.10gen.com
```

Getting stats from the shell:

```
db.serverStatus()
db.stats()
db.<collection name>.stats()

# Do this in another shell to generate load
db.foo.drop()
for (var i=0; i<1000000; i++){
    db.foo.insert({_id: i});
    for (var j=0; j< 10; j++){
        db.foo.findOne({_id: (i-j)});
    }
}
# Add a big string to show disk util.
```

See what the server is currently doing:

```
db.currentOp()

# Use this in another shell to have a query slow enough to see
db.foo.find({$where: 'var i=1000; while(i--); return this.blah != undefined'})
```

Plugins for external tools:

```
Munin, Nagios, Cacti, Ganglia
```

Backup

For backup/restore on a live system:

```
mongodump -d <database> -c <collection> ...
mongorestore -d <database> -c <collection> ...
```

You can also copy/rsync/snapshot the data files:

```
# Make sure you lock the db first
db.fsyncLock()
db.fsyncUnlock()

# On MongoDB pre 2.0
db.runCommand({fsync: 1, lock: 1})
db.$cmd.sys.unlock.findOne()
```

Usually you want to do backups from a slave/secondary.