mongoDB

Replica Sets

Painless scaling and High Availability (HA)

Viktor Petersson

@vpetersson

Who am I?

```
{
 name : ‘Viktor Petersson’,
 description : [ ‘geek’, ‘entrepreneur’ ],
 founder_of : ‘wireload’,
 twitter : ‘@vpetersson’,
 email : ‘vpetersson@wireload.net’,
}
```

What is MongoDB?

- Key-value store
- Schemas less
- Distributed file system (GridFS)
- Built-in Map/Reduce
- Built to scale

MongoDB 101

Database structure (simplified)

Relational Database	MongoDB
Database	Database
Table	Collection
Row	Document

MongoDB 101

Node types

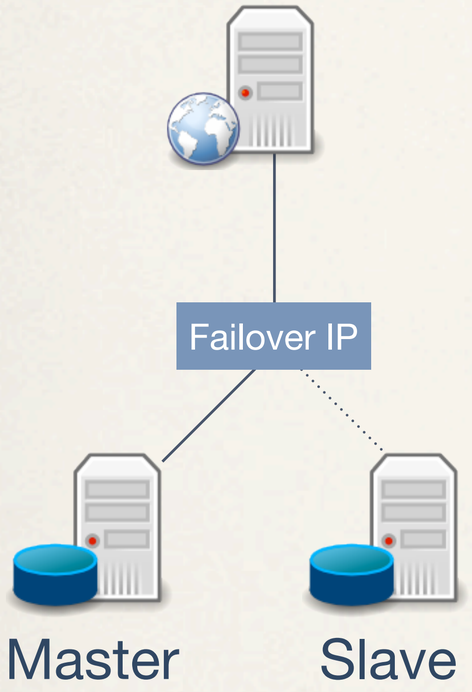
Relational Database	MongoDB
Master	Primary
Slave	Secondary
	Arbiter
	(Hidden)

What are Replica Sets and why should I care?

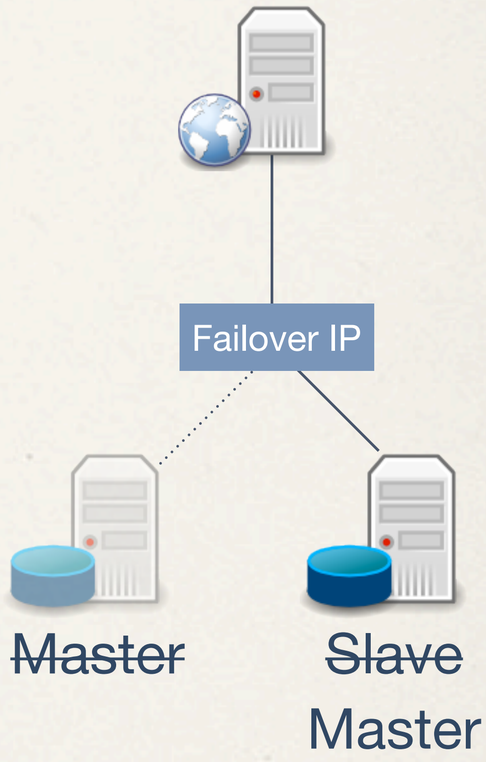
Typical failover with relational databases

(two servers, can lose one)

Normal



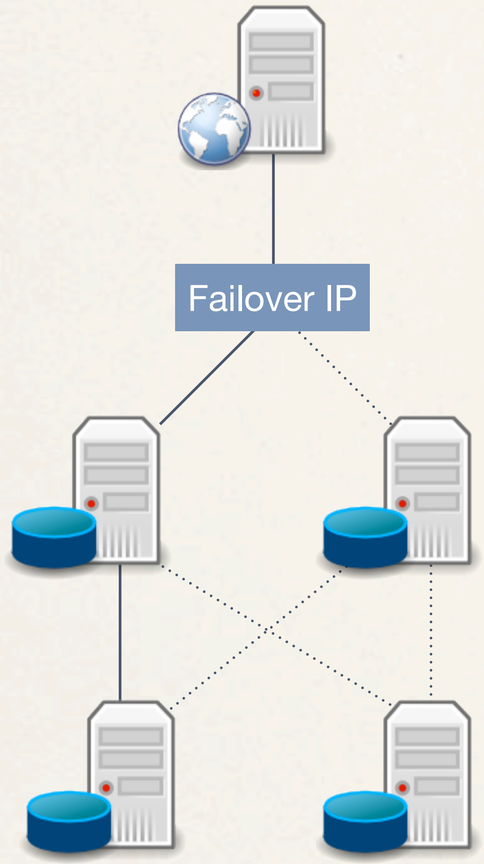
Degraded



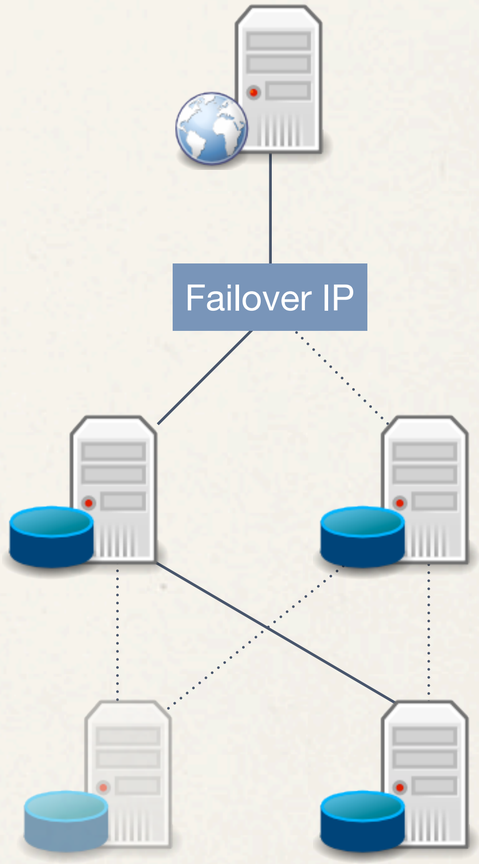
More complex failover with relational databases

(four servers, can lose one of each kind)

Normal



Degraded



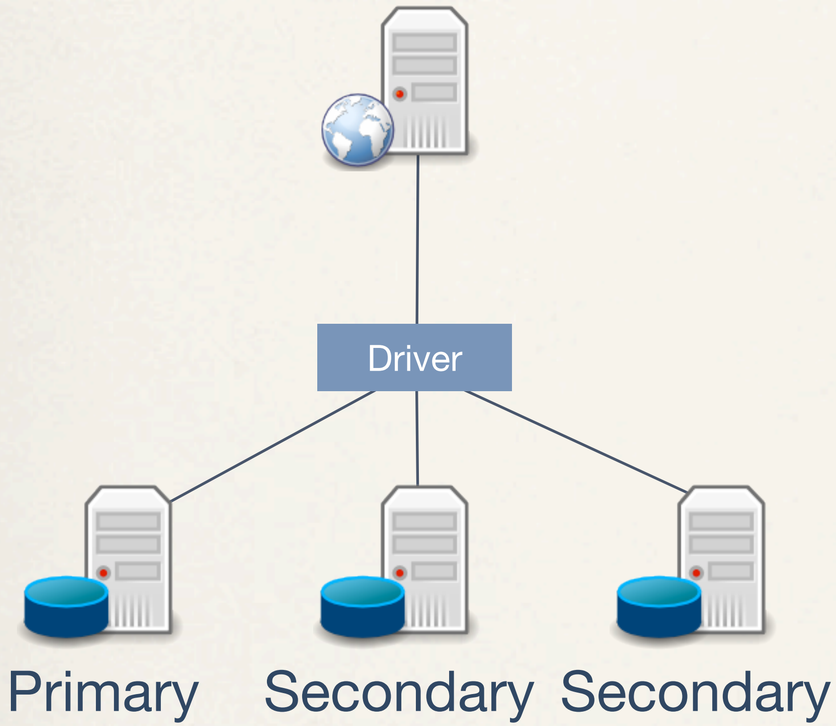
Proxies

Database servers

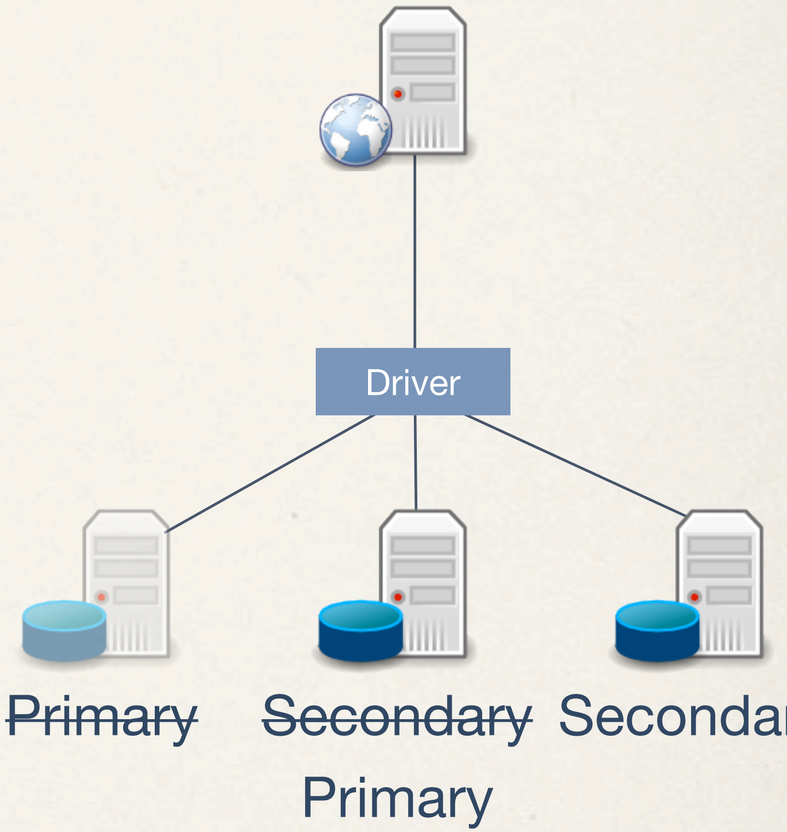
Failover with MongoDB

(three servers, can lose two)

Normal

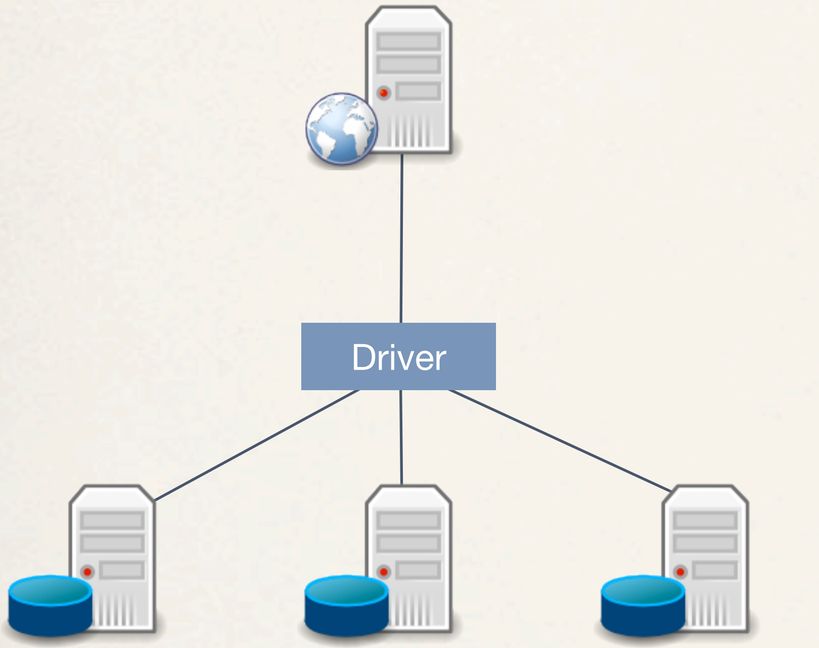


Degraded



Automatic recovery

Down -> Recovering

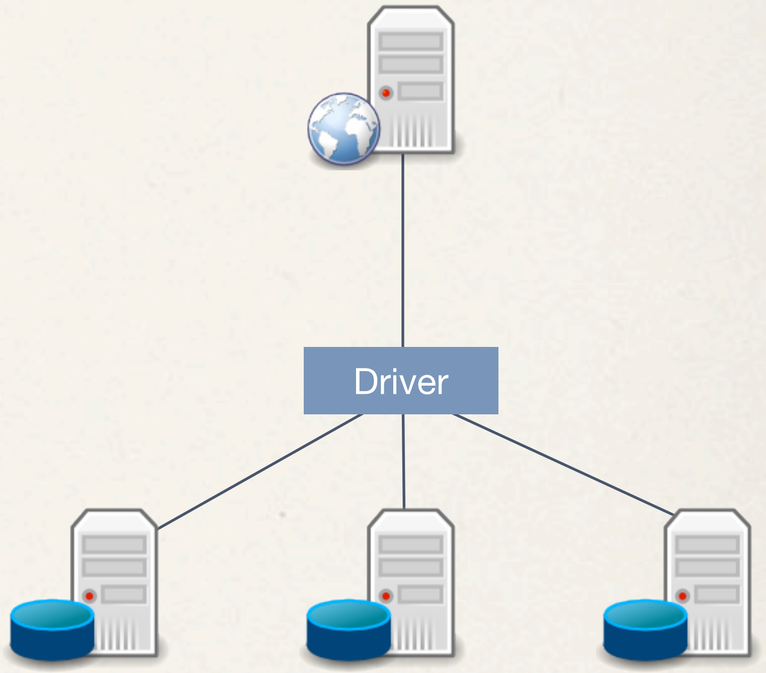


~~Primary~~ Recovering

~~Secondary~~ Primary

Secondary

Recovering -> Secondary



~~Primary~~ Secondary

~~Secondary~~ Primary

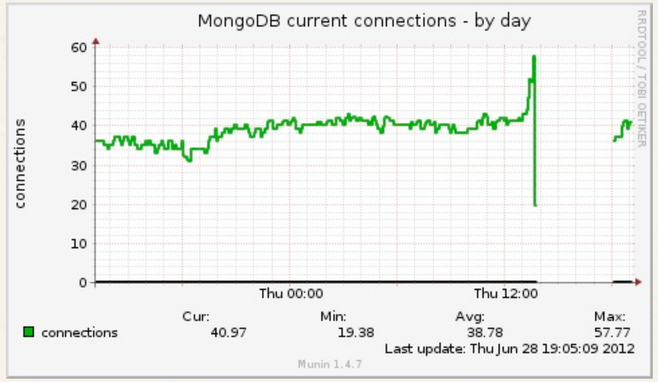
Secondary

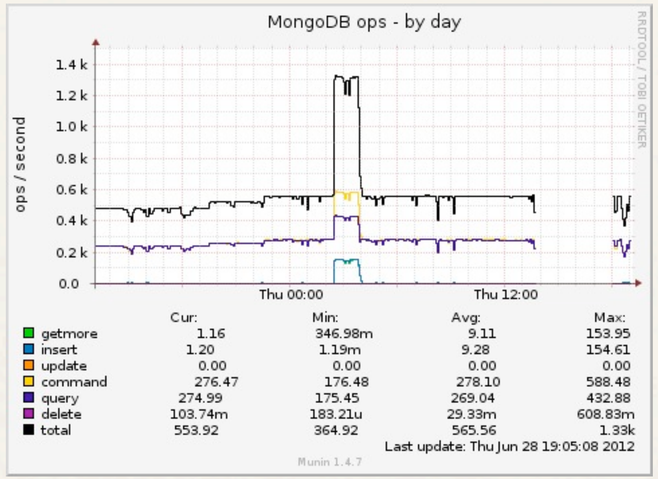
Real life example:

Taking down the primary for maintenance.

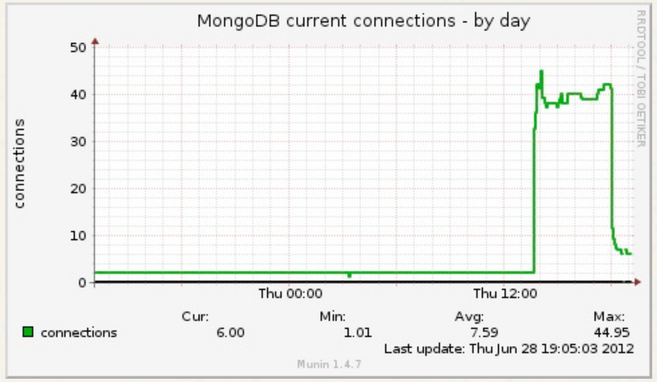
No dropped connections.

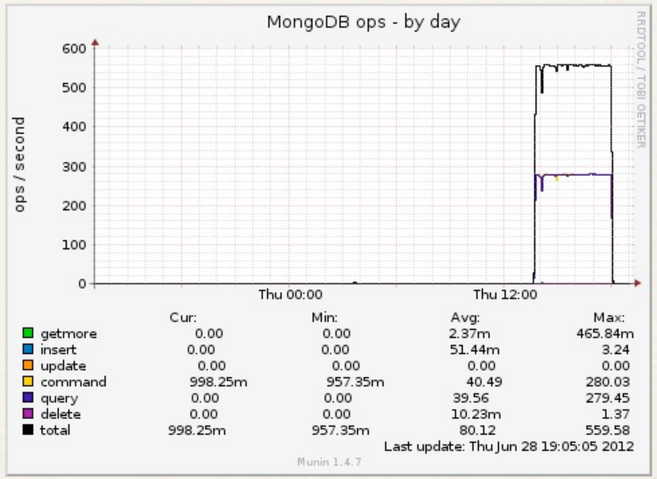
Primary





Secondary





Creating a Replica Set is dead simple.

All it takes is three servers (s0, s1, s2) and three commands[1].

```
$ mongo
  > rs.initiate()
  > rs.add('s1')
  > rs.add('s2')
```


1. After adding "replSet=myRepSet" and "rest = true" to mongo.conf

Live demo!

3 servers with 1 client

Election

1. Most up to date
2. Highest priority
3. Less than 10s behind Primary

Configure node priority

```
$ cfg = rs.conf()
$ cfg.members[x].priority = n
$ rs.reconfig(cfg)
```

x = Member id, n = Priority (0 - 100), default is 1.

Leverage your secondary-nodes

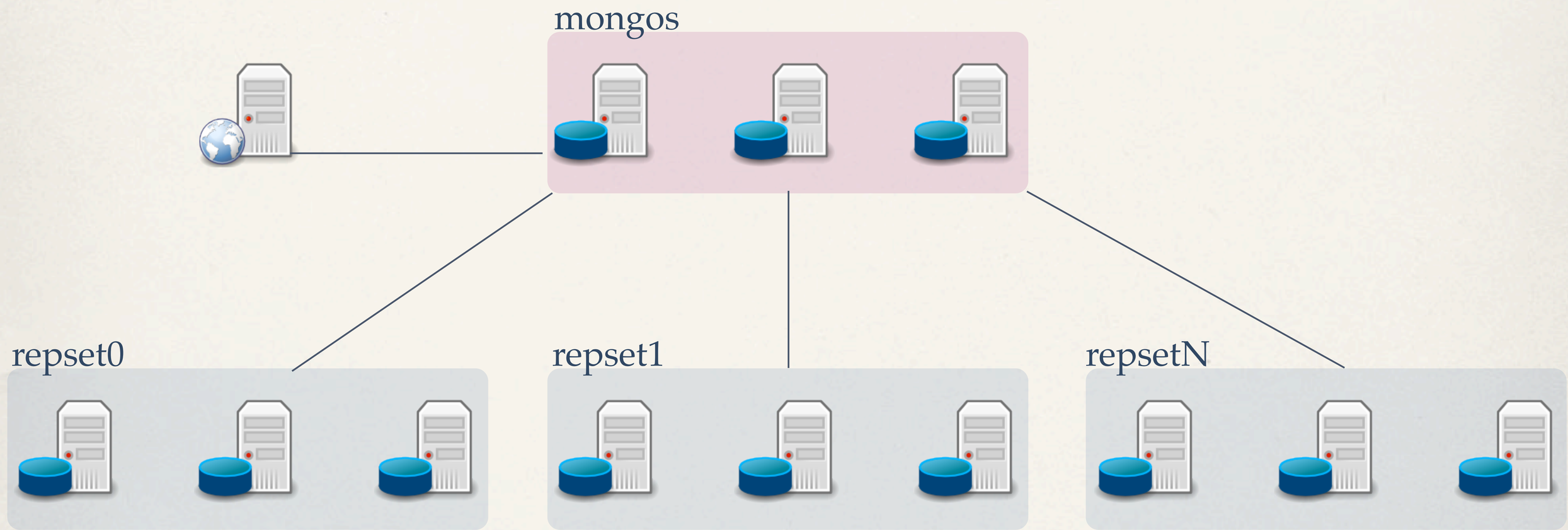
Read preference

- Primary (default)
- Secondary
- Secondary_only

Warning: Secondaries may be out-of-date.

What happens when I've outgrown my Replica Set?

Sharding



Useful commands

View server status

```
$ db.serverStatus()
```

View the replication status

```
$ rs.status()
```

Step down as Primary

```
$ rs.stepDown(n)
```

What's new in MongoDB 2.2?

(released yesterday)

- Improved location awareness
- Improved concurrent capacity
- TTL collections

Full release notes:
http://docs.mongodb.org/manual/release-notes/2.2/

Summary

- Replica Sets are extremely easy to set up
- Doesn’t require failover IPs or custom scripts
- Minimal maintenance
- Built to scale

Questions?

Thanks to

Contact me!

email: info@viktorpetersson.com

www: viktorpetersson.com

twitter: @vpetersson

This deck will be available at ViktorPetersson.com.

Also visit

WireLoad.net

YippieMove.com

Useful resources

- MongoDB: GridFS, sharding and deploying in the cloud (http://goo.gl/1QAV1)
- MongoDB Replica Set (http://goo.gl/D2pkq)
- 10gen (http://goo.gl/3m4Kn)