



# Jim Garvin

@coderifous

<http://thegarvin.com>



I'm an independent software consultant & developer. I do full stack web development with Ruby/Rails. I use MongoDB on many projects, including BusyConf.

<http://busyconf.com>

# Hi

These slides are a bit useless without the talk and group discussion that went with them, but I added a bunch of these sticky notes to provide a substitute narration.

Also, I added some links to blog posts that were mentioned at the meetup. They're on the last slide.



Nathen pinged me on  
twitter about a week  
ago...

“can you speak  
about mongo at  
the meetup next  
week?”

“I’d love to!”





A week later...

“Are you still up  
for giving a  
presentation at  
tomorrow's  
mongo group?”

“Holy crap,  
that's  
tomorrow.  
MUST THINK  
OF TOPIC.”



So what should the topic be?

Should we do a talk on some of MongoDB's major feature points?

# Replica sets?

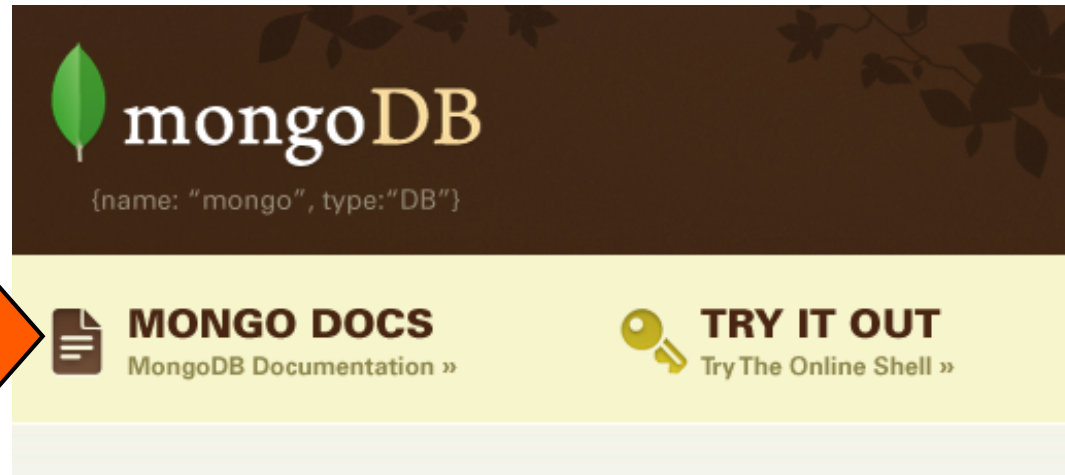
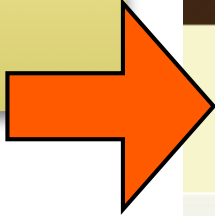
# Sharding?

# GridFS?

# MapReduce?

My feeling on that is: Nearly all of the value in a talk about Feature X is delivered in the first minute, when someone learns that Feature X exists at all.

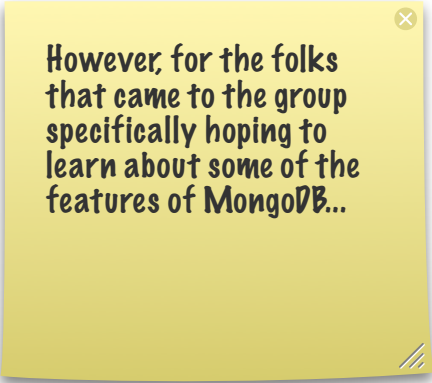
For all the details about Feature X, you're probably going to get more value out of the docs than the next 30 minutes of that presentation.



[www.mongodb.org](http://www.mongodb.org)



# Knowing MongoDB has a feature is half the battle...



However, for the folks  
that came to the group  
specifically hoping to  
learn about some of the  
features of MongoDB...

# Replica sets!

(master/slave with auto failover)

# Sharding!

(tons of data)

# GridFS!

(storing large files)

# MapReduce!

(batch processing)

So now for the  
**REAL** topic...

Why doesn't your  
business use MongoDB?



We talk about how lots of businesses are using MongoDB. But to be fair, many of them aren't using it for their critical data.

It's too young. It's not "durable".

The group shared reasons  
that they've heard for  
NOT using MongoDB.

These were mine.

Some are FUD.

No developers know it.

Where's the certified DB admin?

It's changing too quickly.

We already HAVE a database.

Feels like the same  
kind of resistance  
that Ruby/Rails got  
when it was new.

It's too new.      It's not "secure".

These were some of the things we remember hearing back in '05 and '06 when Ruby/Rails was new.

No developers know it.

It's not enterprise grade.

It's changing too quickly.

We already HAVE a programming language.





In a few years,  
MongoDB will  
be as common  
as Rails.



**Discuss!**



# Jim Garvin

@coderifous

<http://thegarvin.com>



I'm an independent software consultant & developer. I do full stack web development with Ruby/Rails. I use MongoDB on many projects, including BusyConf.

<http://busyconf.com>

# Post-Meetup Notes

Here's a few blog posts that were referenced during the meetup:

Most of the discussion about MongoDBs lack of "single server durability" is moot with the 1.8 release, which brings write-ahead journaling to stable/production MongoDB.

Kyle Banker's excellent article about how you might use MongoDB for e-commerce.

<http://kylebanker.com/blog/2010/04/30/mongodb-and-ecommerce/>

Post from 10gen talking about durability.

<http://blog.mongodb.org/post/381927266/what-about-durability>

Mikael Rogers' post about MongoDB and durability, which was a bit sensational at the time, and may have played a role in 10gens prioritization of durability for v1.8

[http://redbeard0531.s3.amazonaws.com/mikeals\\_blog\\_backup/MongoDB+Performance+&+Durability.html](http://redbeard0531.s3.amazonaws.com/mikeals_blog_backup/MongoDB+Performance+&+Durability.html)