Baltimore MongoDB Meetup, 23 July 2013
http://www.meetup.com/Baltimore-MongoDB-Users-Group/

MongoDB Intro

A basic overview

Michael Behrens, R2AD, LLC
Jose Rodriguez

Brief source:
https://github.com/r2ad/planeBigData
https://github.com/madaxx/mongoflight/

MongoDB Website: http://www.mongodb.org/

Categories of Databases

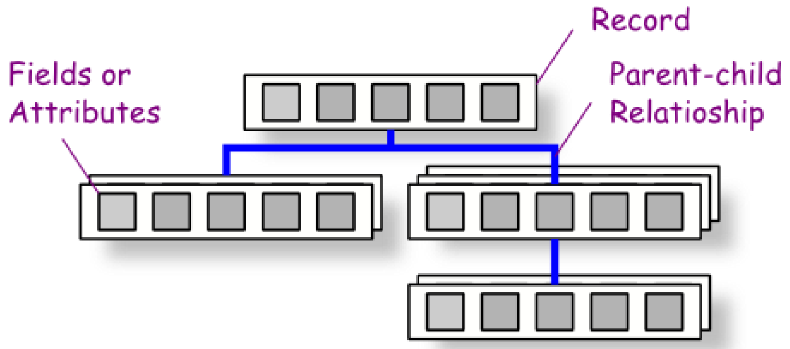
These categories help us to understand the basic benefits and limitations of various offerings

- Relational
  - Relational database management systems (RDBMSs) are set-theory-based systems implemented as two-dimensional tables with rows and columns.
- Document

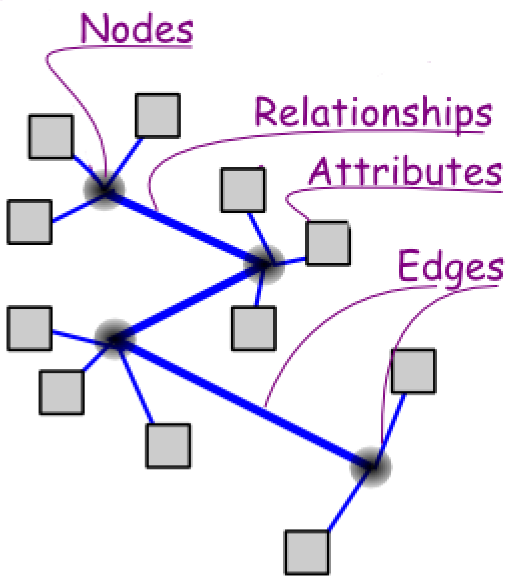
  Focus of this Brief
  - Stores documents based on a unique ID field. Documents can be JSON documents, XML, or anything.
- Key Value
  - Pairs keys to values in much the same way that a map (or hash table) would in any popular programming language.
- Columnar
  - Data from a given column (in the two-dimensional tabular sense) is stored together (column-oriented). Adding columns, for example is fast.
- Graph
  - Consists of nodes and relationships between nodes. Excels at dealing with highly interconnected data.

Depiction of Database Categories

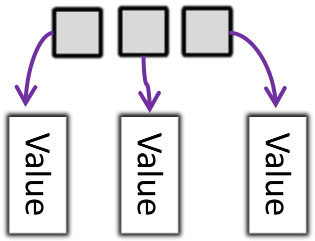
Relational Databases



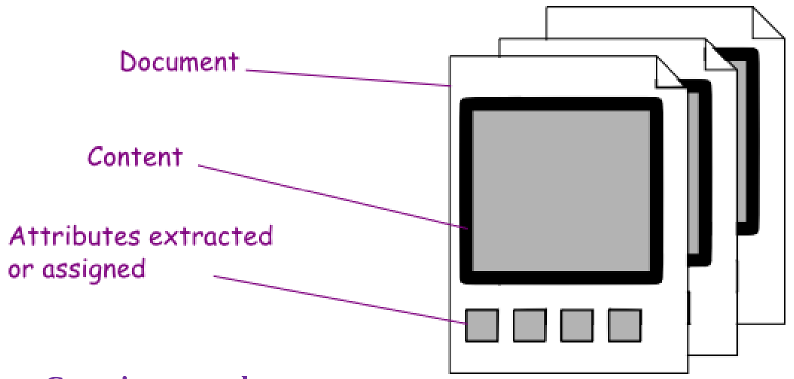
Graph Databases



Key Value

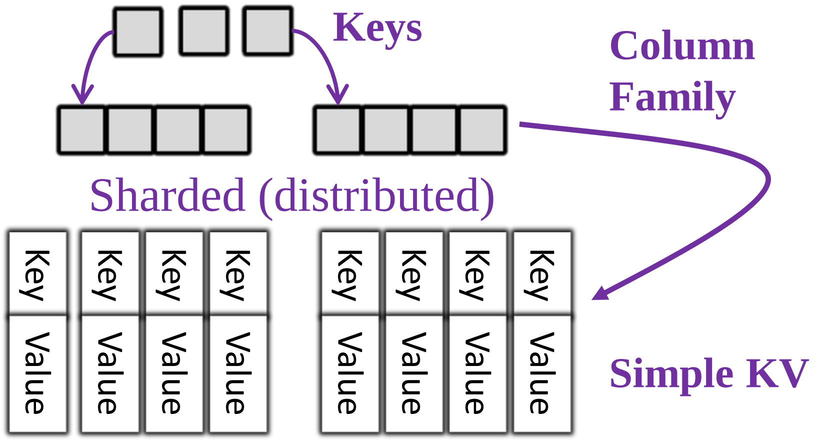


Document Databases



Containers can have multiple documents references by the same key (see CDMI)

Columnar Databases



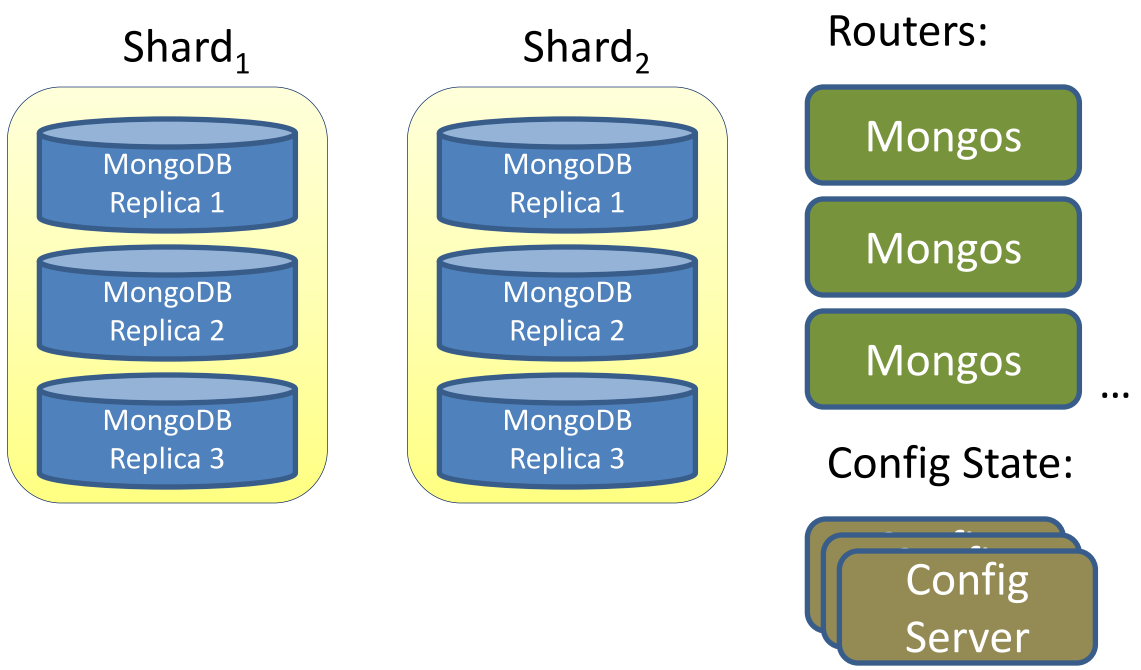
Images (w/Permission): http://www.cbsolution.net/ontarget/databases_relational_vs_object_vs

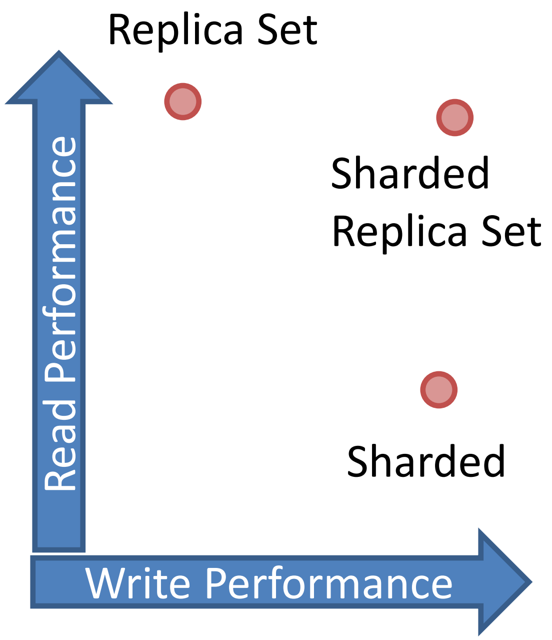
MongoDB Overview

- **MongoDB** (from "hu**mongo**us") is a document oriented, scalable, open source NoSQL database supported by 10gen.

- Highlights of MongoDB
  - JSON documents
  - Supports Ad Hoc Queries and M/R
  - Very Speedy, index capable
    - at the expense of data integrity in the event of a server crash
  - Scalable via replication of nodes
  - Geospatial support
  - Multiple Programming languages supported
  - Very popular with startups and other groups
  - Bundled in leading PaaS Distributions (OpenShift, CloudFoundry)

- Terminology
  - Table,View -> Collection
  - Row -> Document
  - Join -> Embedded
  - Foreign Key -> Reference

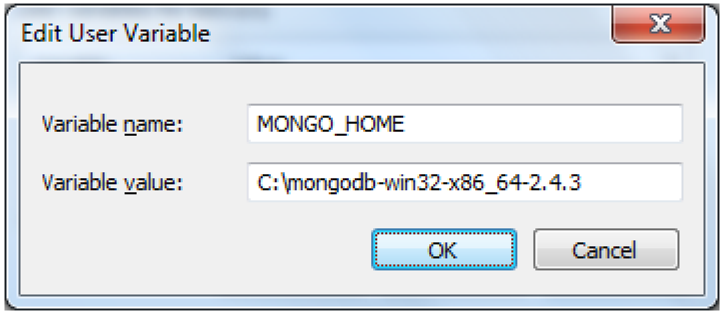




Setup/Install…Windows Example

- Install and set environment path…

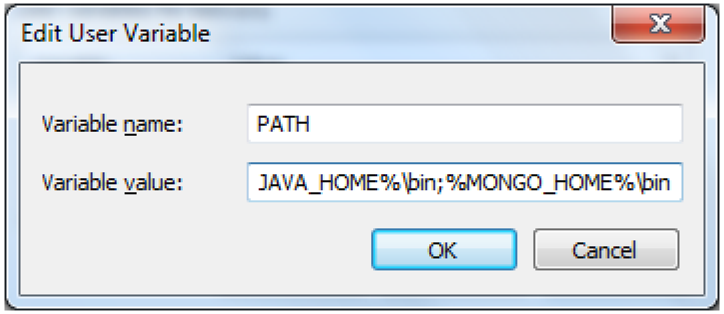
1)


Edit User Variable

Variable name: MONGO_HOME

Variable value: C:\mongodb-win32-x86_64-2.4.3

2)


Edit User Variable

Variable name: PATH

Variable value: JAVA_HOME%\bin;%MONGO_HOME%\bin

3)

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\>which mongod
C:\mongodb-win32-x86_64-2.4.3\bin/mongod.exe
```

Setup/Install…Linux Example

- First, setup Repo…then,

  – yum install mongo-10gen mongo-10gen-server
- Then start the service

```
[root@r2adnet opt]# service mongod start
Starting mongod: about to fork child process, waiting until server is ready for
connections.
forked process: 14396
all output going to: /var/log/mongo/mongod.log
child process started successfully, parent exiting
                                                    [  OK  ]
[root@r2adnet opt]# which mongo
/usr/bin/mongo
```

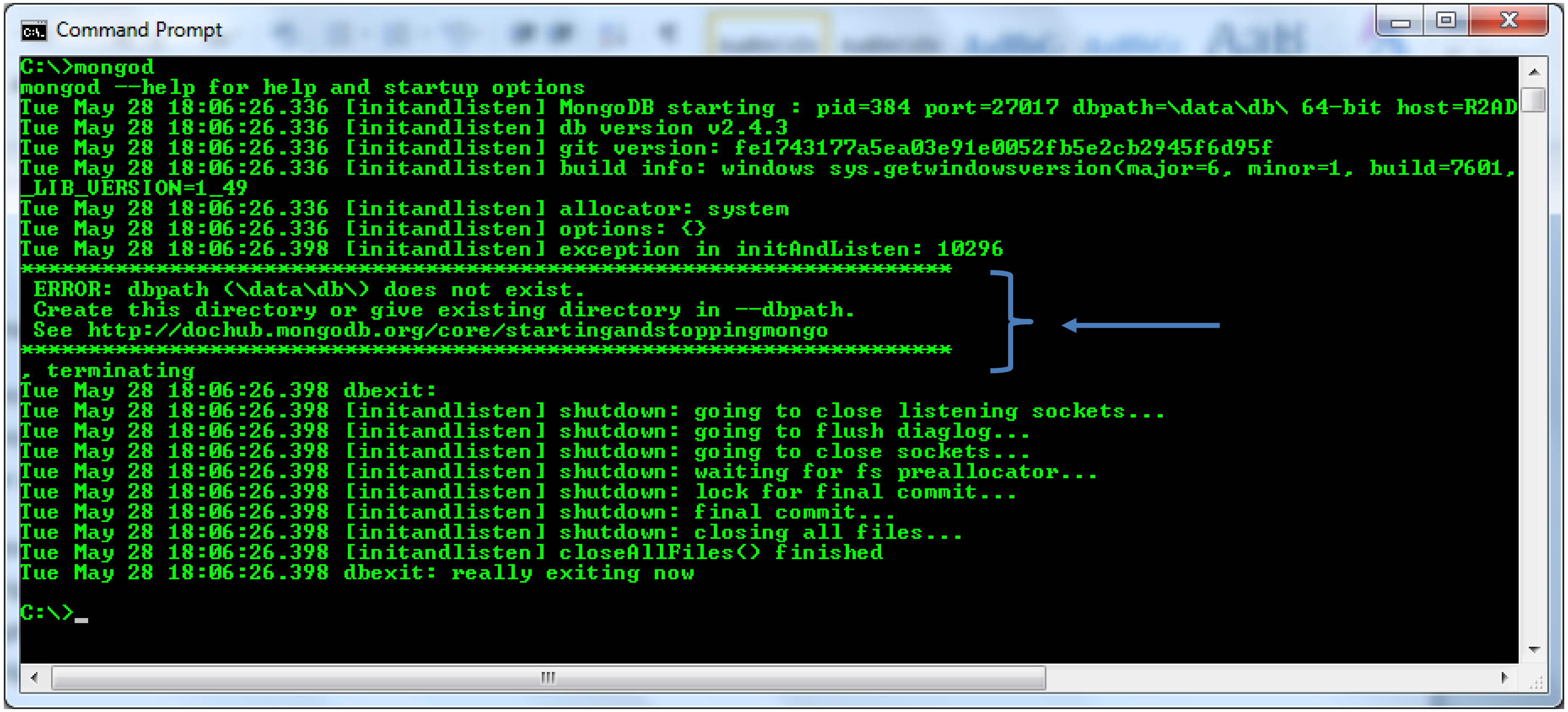
- Make it always start
  – chkconfig mongod on

Ref: http://docs.mongodb.org/manual/tutorial/install-mongodb-on-red-hat-centos-or-fedora-linux/

Starting the server, Windows

- To run a single server database:

```
C:\> mkdir /data/db
C:\> mongod
```

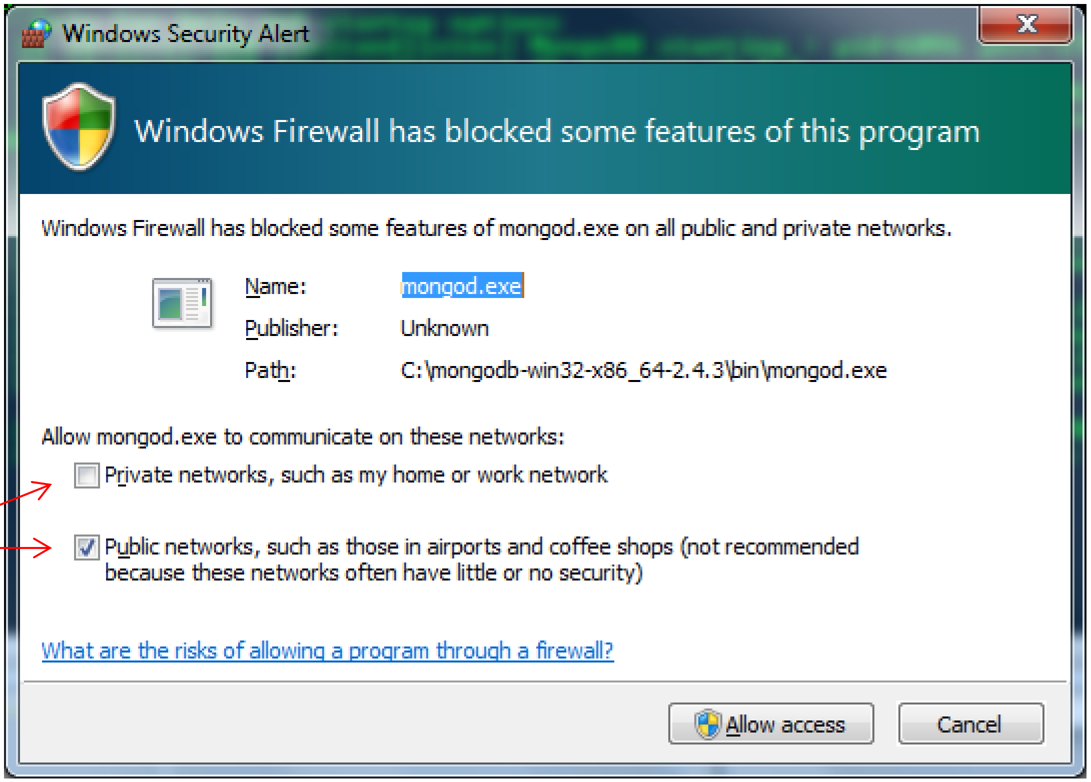


Command Prompt

```
C:\>mongod
mongod --help for help and startup options
Tue May 28 18:06:26.336 [initandlisten] MongoDB starting : pid=384 port=27017 dbpath=\data\db\ 64-bit host=R2AD
Tue May 28 18:06:26.336 [initandlisten] db version v2.4.3
Tue May 28 18:06:26.336 [initandlisten] git version: fe1743177a5ea03e91e0052fb5e2cb2945f6d95f
Tue May 28 18:06:26.336 [initandlisten] build info: windows sys.getwindowsversion(major=6, minor=1, build=7601,
_LIB_VERSION=1_49
Tue May 28 18:06:26.336 [initandlisten] allocator: system
Tue May 28 18:06:26.336 [initandlisten] options: {}
Tue May 28 18:06:26.398 [initandlisten] exception in initAndListen: 10296
*********************************************************************
 ERROR: dbpath (\data\db\) does not exist.
 Create this directory or give existing directory in --dbpath.
 See http://dochub.mongodb.org/core/startingandstoppingmongo
*********************************************************************
, terminating
Tue May 28 18:06:26.398 dbexit:
Tue May 28 18:06:26.398 [initandlisten] shutdown: going to close listening sockets...
Tue May 28 18:06:26.398 [initandlisten] shutdown: going to flush diaglog...
Tue May 28 18:06:26.398 [initandlisten] shutdown: going to close sockets...
Tue May 28 18:06:26.398 [initandlisten] shutdown: waiting for fs preallocator...
Tue May 28 18:06:26.398 [initandlisten] shutdown: lock for final commit...
Tue May 28 18:06:26.398 [initandlisten] shutdown: final commit...
Tue May 28 18:06:26.398 [initandlisten] shutdown: closing all files...
Tue May 28 18:06:26.398 [initandlisten] closeAllFiles() finished
Tue May 28 18:06:26.398 dbexit: really exiting now

C:\>_
```

Be aware of ports…firewall & Security

- By default:
  - mongod is waiting for connections on port **27017**
  - web status page is waiting for connections on port **28017**



Windows Security Alert

Windows Firewall has blocked some features of this program

Windows Firewall has blocked some features of mongod.exe on all public and private networks.

Name: mongod.exe
Publisher: Unknown
Path: C:\mongodb-win32-x86_64-2.4.3\bin\mongod.exe

Allow mongod.exe to communicate on these networks:

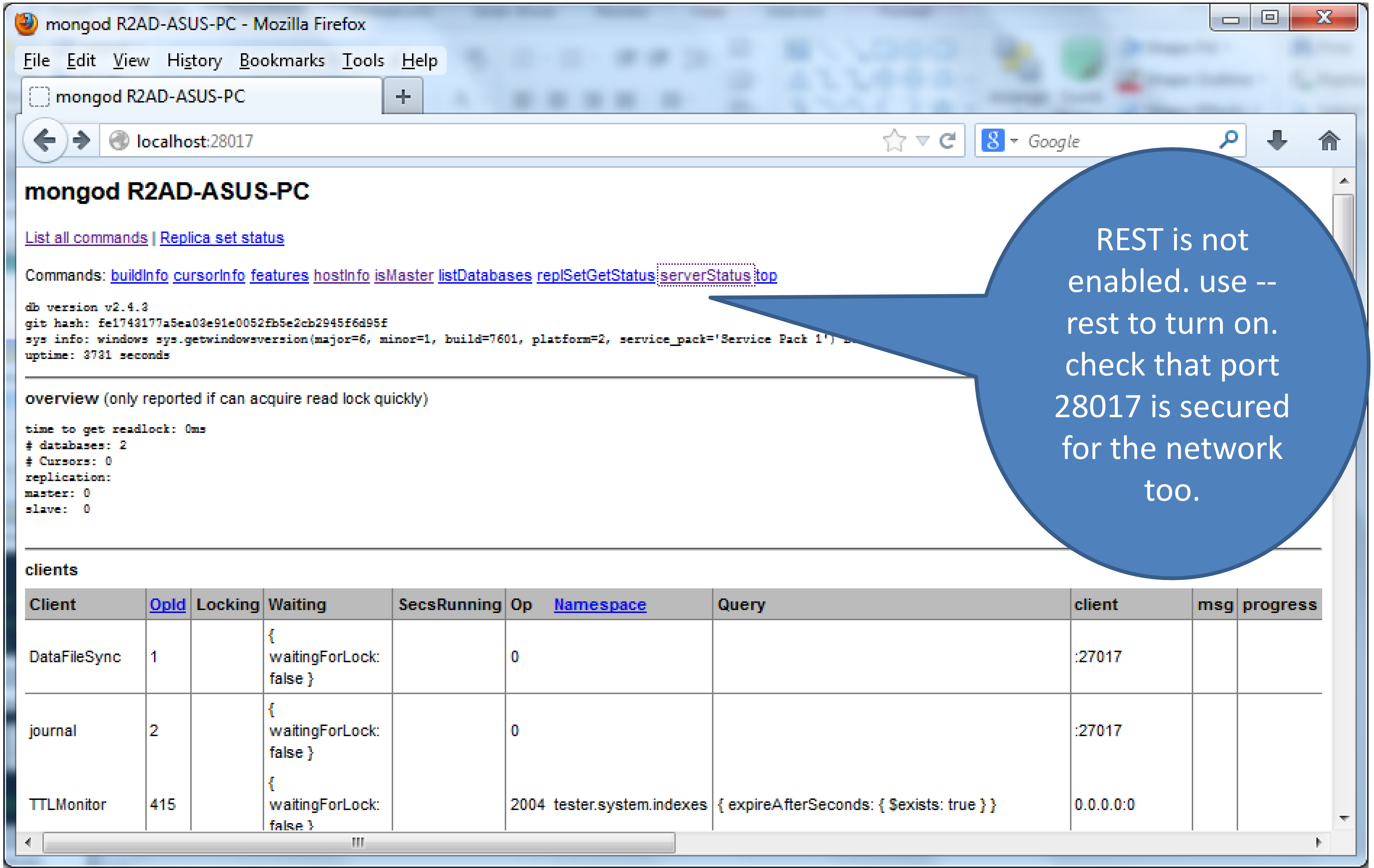
- [ ] Private networks, such as my home or work network
- [✓] Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing a program through a firewall?

decide

Note: Best to allow database access only to server(s) that needed it…..keep traffic inside as much as possible – good security practice. See: http://docs.mongodb.org/manual/core/security/

Web Status Console


mongod R2AD-ASUS-PC - Mozilla Firefox
File Edit View History Bookmarks Tools Help
mongod R2AD-ASUS-PC
localhost:28017
Google

mongod R2AD-ASUS-PC

List all commands | Replica set status

Commands: buildInfo cursorInfo features hostInfo isMaster listDatabases replSetGetStatus serverStatus top

```
db version v2.4.3
git hash: fe1743177a5ea03e91e0052fb5e2cb2945f6d95f
sys info: windows sys.getwindowsversion(major=6, minor=1, build=7601, platform=2, service_pack='Service Pack 1')
uptime: 3731 seconds
```

overview (only reported if can acquire read lock quickly)

```
time to get readlock: 0ms
# databases: 2
# Cursors: 0
replication:
master: 0
slave:  0
```

clients

Client	OpId	Locking	Waiting	SecsRunning	Op	Namespace	Query	client	msg	progress
DataFileSync	1		{ waitingForLock: false }		0			:27017		
journal	2		{ waitingForLock: false }		0			:27017		
TTLMonitor	415		{ waitingForLock: false }		2004	tester.system.indexes	{ expireAfterSeconds: { $exists: true } }	0.0.0.0:0		

REST is not enabled. use --rest to turn on. check that port 28017 is secured for the network too.

mongo – the shell

```
[root@r2adnet opt]# mongo
MongoDB shell version: 2.4.4
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
>
> help
        db.help()                    help on db methods
        db.mycoll.help()             help on collection methods
        sh.help()                    sharding helpers
        rs.help()                    replica set helpers
        help admin                   administrative help
        help connect                 connecting to a db help
….
```

mongo is an interactive JavaScript shell interface to MongoDB

Adding data manually

```
> db.plane.insert({
... "callsign" : "UAL653",
...     "model" : "757-200",
...     "manufacturer" : "Boeing",
...     "lat" : "33.97",
...     "lon" : "118.203611",
...     "altituteFt" : "31000",
...     "speedKt" : "480",
...     "heading" : "121",
...     "timestamp" : "11:29PM EDT",
...     "PositionsCount" : "0",
... }
... )
>
> db.plane.find()
{ "_id" : ObjectId("51cb6ee9cb8ffebb1dbede9f"), "callsign" : "UAL653", "model" : "757-
200", "manufacturer" : "Boeing", "lat" : "33.97", "lon" : "118.203611", "a
ltituteFt" : "31000", "speedKt" : "480", "heading" : "121", "timestamp" : "11:29PM
EDT", "PositionsCount" : "0" }
>
```

Adding a JSON document….note that it automatically adds an object ID

Note also that I did not have to create a table or database ahead of time! The “plane” collection is created automatically

Test database is the default

The “use” command can change databases.

Adding data manually

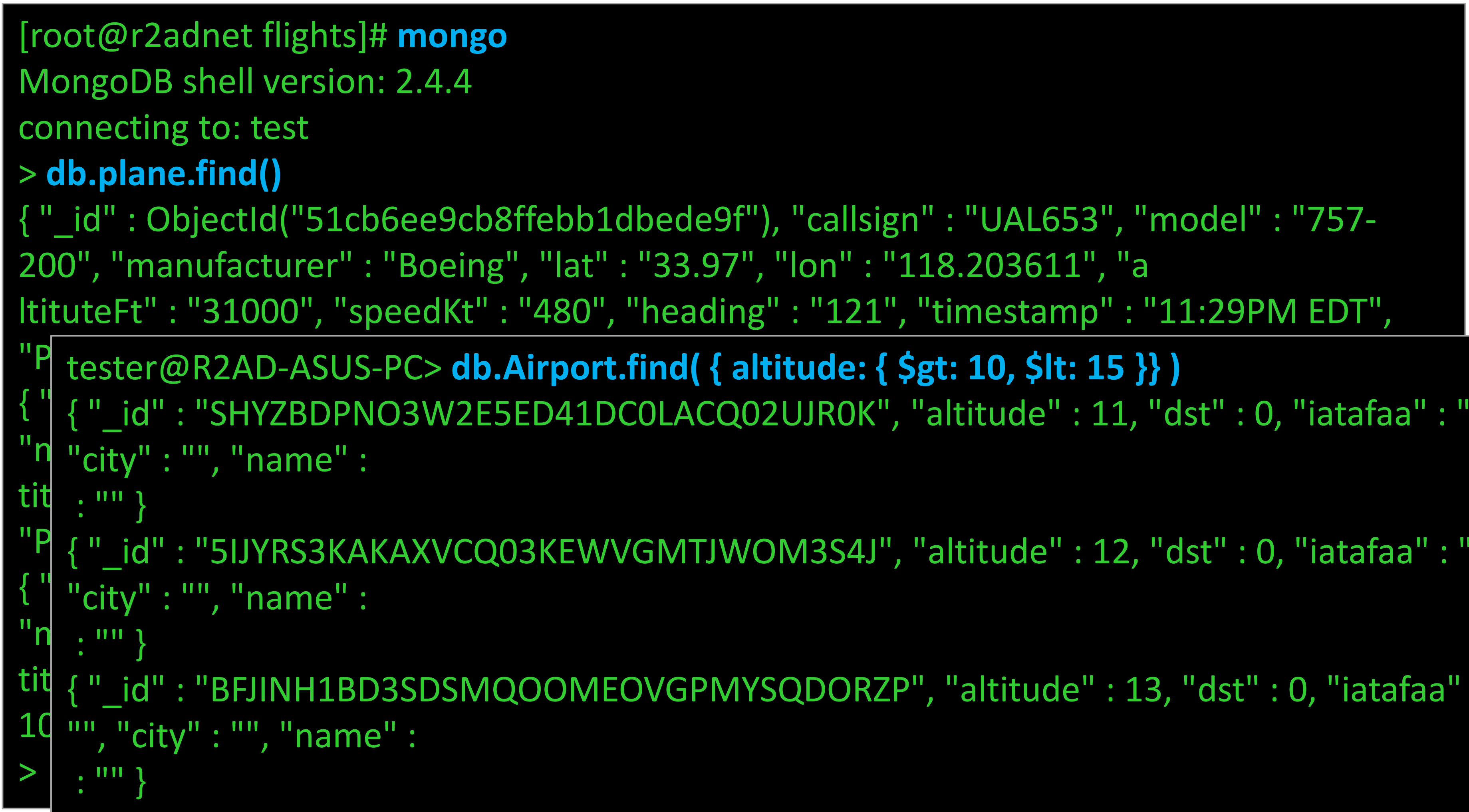
Added some more records....how many do we have?

```
> db.plane.count()
3
```

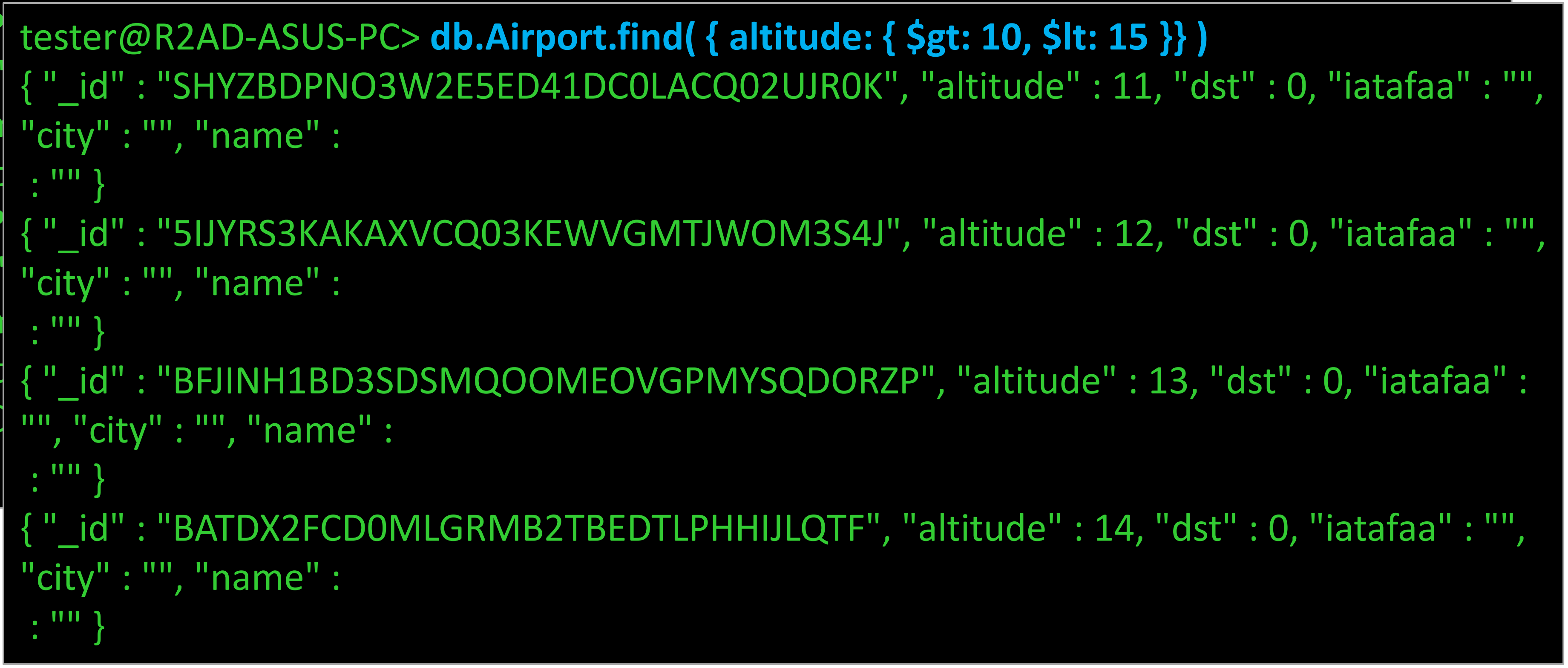
The log file can be examined from the OS or from the web page.

```
[root@r2adnet flights]# cat /var/log/mongo/mongod.log

***** SERVER RESTARTED *****

Wed Jun 26 21:26:56.209 [initandlisten] MongoDB starting : pid=14396 port=27017 dbpath=/var/lib/mongo 64-bit
host=r2adnet
Wed Jun 26 21:26:56.209 [initandlisten] db version v2.4.4
Wed Jun 26 21:26:56.209 [initandlisten] git version: 4ec1fb96702c9d4c57b1e06dd34eb73a16e407d2
Wed Jun 26 21:26:56.209 [initandlisten] build info: Linux ip-10-2-29-40 needed
Wed Jun 26 21:26:56.299 [FileAllocator] allocating new datafile /var/lib/mongo/local.ns, filling with zeroes...
Wed Jun 26 21:26:56.299 [FileAllocator] creating directory /var/lib/mongo/_tmp
Wed Jun 26 21:26:56.388 [FileAllocator] done allocating datafile /var/lib/mongo/local.ns, size: 16MB,  took 0.051 secs
Wed Jun 26 21:26:56.389 [FileAllocator] allocating new datafile /var/lib/mongo/local.0, filling with zeroes...
Wed Jun 26 21:26:57.737 [FileAllocator] done allocating datafile /var/lib/mongo/local.0, size: 64MB,  took 1.348 secs
Wed Jun 26 21:26:57.783 [initandlisten] command local.$cmd command: { create: "startup_log", size: 10485760, capped: true
} ntoreturn:1 keyUpdates:0  reslen:37
1486ms
Wed Jun 26 21:26:57.784 [websvr] admin web console waiting for connections on port 28017
Wed Jun 26 21:26:58.212 [initandlisten] waiting for connections on port 27017
Wed Jun 26 21:31:16.455 [initandlisten] connection accepted from 127.0.0.1:55611 #1 (1 connection now open)
```

Quick search to see if data exists



```
[root@r2adnet flights]# mongo
MongoDB shell version: 2.4.4
connecting to: test
> db.plane.find()
{ "_id" : ObjectId("51cb6ee9cb8ffebb1dbede9f"), "callsign" : "UAL653", "model" : "757-
200", "manufacturer" : "Boeing", "lat" : "33.97", "lon" : "118.203611", "a
ltituteFt" : "31000", "speedKt" : "480", "heading" : "121", "timestamp" : "11:29PM EDT",
"P
{ "
"n
tit
"P
{ "
"n
tit
10
>
```



```
tester@R2AD-ASUS-PC> db.Airport.find( { altitude: { $gt: 10, $lt: 15 }} )
{ "_id" : "SHYZBDPNO3W2E5ED41DC0LACQ02UJR0K", "altitude" : 11, "dst" : 0, "iatafaa" : "",
"city" : "", "name" :
 : "" }
{ "_id" : "5IJYRS3KAKAXVCQ03KEWVGMTJWOM3S4J", "altitude" : 12, "dst" : 0, "iatafaa" : "",
"city" : "", "name" :
 : "" }
{ "_id" : "BFJINH1BD3SDSMQOOMEOVGPMYSQDORZP", "altitude" : 13, "dst" : 0, "iatafaa" :
"", "city" : "", "name" :
 : "" }
{ "_id" : "BATDX2FCD0MLGRMB2TBEDTLPHHIJLQTF", "altitude" : 14, "dst" : 0, "iatafaa" : "",
"city" : "", "name" :
 : "" }
```

Find and Pretty

```
> db.plane.find({"speedKt":"480"})
{ "_id" : ObjectId("51cb6ee9cb8ffebb1dbede9f"), "callsign" : "UAL653", "model" : "757-
200", "manufacturer" : "Boeing", "lat" : "33.97", "lon" : "118.203611", "a
ltituteFt" : "31000", "speedKt" : "480", "heading" : "121", "timestamp" : "11:29PM EDT",
"PositionsCount" : "0" }
> db.plane.find({"speedKt":"480"}).pretty()
{
        "_id" : ObjectId("51cb6ee9cb8ffebb1dbede9f"),
        "callsign" : "UAL653",
        "model" : "757-200",
        "manufacturer" : "Boeing",
        "lat" : "33.97",
        "lon" : "118.203611",
        "altituteFt" : "31000",
        "speedKt" : "480",
        "heading" : "121",
        "timestamp" : "11:29PM EDT",
        "PositionsCount" : "0"
}
>
```

Review…..on-line tutorial available

```
> show dbs
local   0.078125GB
mydb    0.203125GB
test    0.203125GB
> db
mydb
> show collections
system.indexes
testData
> db.testData.find()
{ "_id" : ObjectId("51ed9b855ad739bfd0f31088"), "name" : "mongo" }
{ "_id" : ObjectId("51ed9b875ad739bfd0f31089"), "x" : 3 }
>
>
```

This is the simple example from:
http://docs.mongodb.org/manual/tutorial/getting-started/

Collections…..

```
[root@r2adnet flights]# mongo
MongoDB shell version: 2.4.4
connecting to: test
> show collections
plane
system.indexes
> use bigdata
switched to db bigdata
> show collections
> use test
switched to db test
> show collections
plane
system.indexes
>
```

Custom Prompt

UNIX

```
[root@r2adnet flights]#
[root@r2adnet flights]# cat > ~/.mongorc.js
host = db.serverStatus().host;

prompt = function() {
        return db+"@"+host+"> ";
    }
[root@r2adnet flights]#
[root@r2adnet flights]# mongo
MongoDB shell version: 2.4.4
connecting to: test
test@r2adnet> use bigdata
switched to db bigdata
bigdata@r2adnet> exit
bye
```

Windows

```
C:\mongodb-win32-x86_64-2.4.3\bin>copy
con .mongorc.js
host = db.serverStatus().host;

prompt = function() {
        return db+"@"+host+"> ";
    }
^Z
     1 file(s) copied.

C:\mongodb-win32-x86_64-
2.4.3\bin>mongo
MongoDB shell version: 2.4.3
connecting to: test
test@R2AD-ASUS-PC>
```

Assigning the variable prompt can override the default “>” prompt. Also, to make it permanent, put the commands in this file in your home directory: .mongorc
This file gets read in and processed each time the mongo shell starts.

Running a Script

```
[root@r2adnet flights]# cat > findone.js
db.plane.findOne()
[root@r2adnet flights]#
[root@r2adnet flights]# mongo < findone.js
MongoDB shell version: 2.4.4
connecting to: test
{
        "_id" : ObjectId("51cb6ee9cb8ffebb1dbede9f"),
        "callsign" : "UAL653",
        "model" : "757-200",
        "manufacturer" : "Boeing",
        "lat" : "33.97",
        "lon" : "118.203611",
        "altituteFt" : "31000",
        "speedKt" : "480",
        "heading" : "121",
        "timestamp" : "11:29PM EDT",
        "PositionsCount" : "0"
}
bye
```

Note: The redirection character “<“ is needed on linux, not windows.

Lots more to learn!

- This is just a quick intro….for our demo, we wanted to add a Web Front End…
  - Thought about using node.js along with restler, etc.

```
[root@r2adnet opt]# npm install restler
npm http GET https://registry.npmjs.org/restler
npm http 200 https://registry.npmjs.org/restler
npm http GET https://registry.npmjs.org/restler/-/restler-2.0.1.tgz
npm http 200 https://registry.npmjs.org/restler/-/restler-2.0.1.tgz
restler@2.0.1 node_modules/restler
[root@r2adnet opt]#
```

- However, after talking with Jose….
  - Used Lift, a powerful/rapid web framework: http://liftweb.net/
  - and Crudify, which automatically adds CRUD (Create, read, update and delete) operations

Jose’s Simple Application

- Jose started an simple interface for a MongoDB database using lift and crudify. He pushed a beta up to GIT:
  - https://github.com/madaxx/mongoflight/
- To use it…install and configure the following:
  - SBT: https://github.com/sbt/sbt
    - support flexible and powerful build definitions
  - Java 7
  - Scala: http://www.scala-lang.org/downloads
    - Object oriented language which interoperates with Java

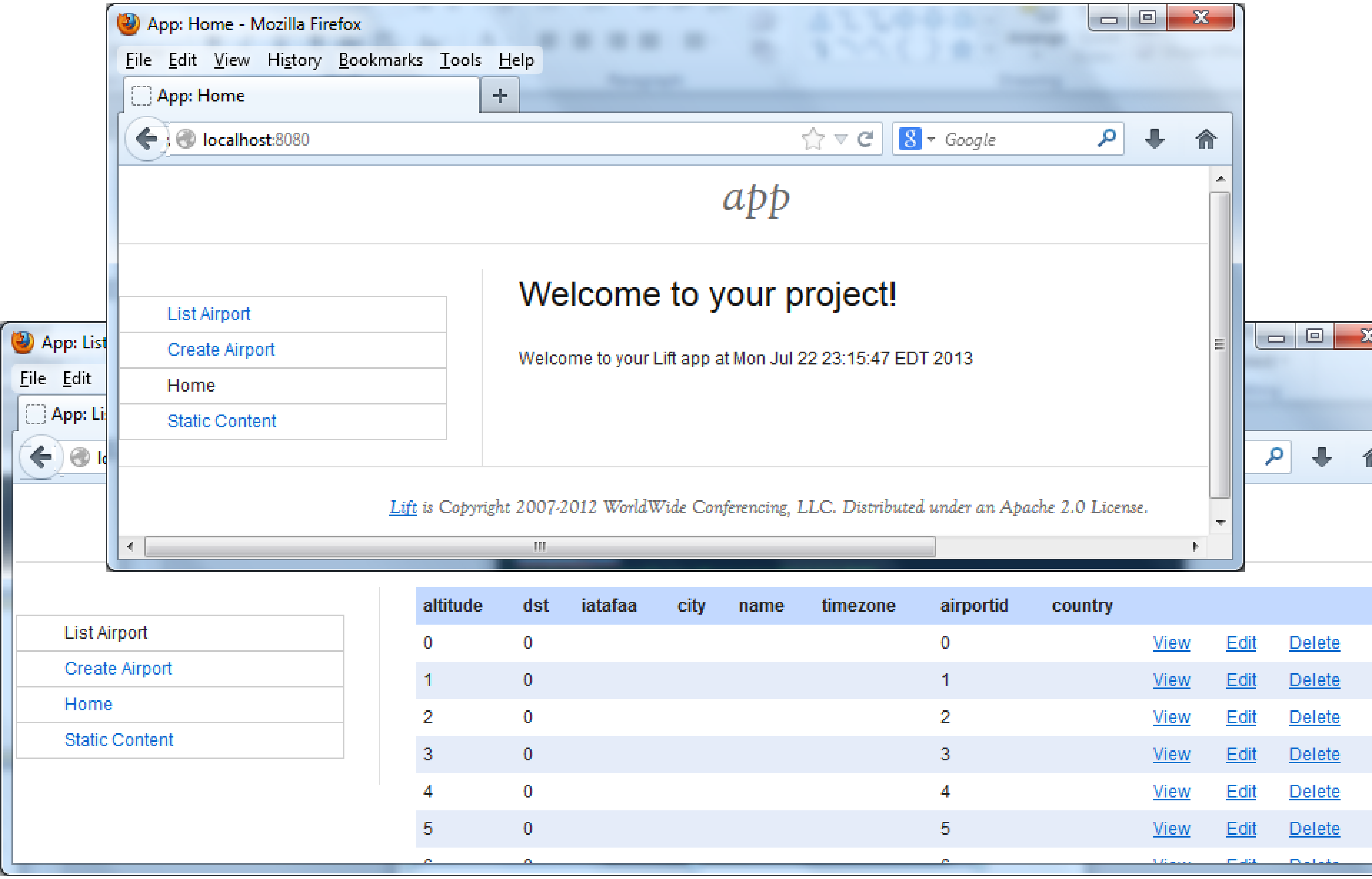
Jose's Simple Application

- Currently configured to use the local mongodb installation but can be set to use any. So start the mongod.
- Then download the project and then type the following commands in the project directory:
  - SBT
  - update
  - reload
  - container:Start
- If all goes well , open your browser to localhost:8080.
- As you update it will create a database and a collection called airports. Changing the Object Model is easy. There are a few kinks for example the latlong mongotype is not fully working w/crudify.
  - BTW type container:stop to exit.

Starting...

```
D:\mongoflight-master>sbt
D:\mongoflight-master>set SCRIPT_DIR=D:\mongoflight-master\
D:\mongoflight-master>java -XX:+CMSClassUnloadingEnabled -XX:MaxPermSize=256m -
Xmx1024M -Xss2M -jar "D:\mongoflight-master\\sbt-launch-0.12.1.jar"
Getting org.scala-sbt sbt 0.12.2 ...
downloading http://repo.typesafe.com/typesafe/ivy-releases/org.scala ...
        [SUCCESSFUL ] org.scala-sbt#sbt;0.12.2!sbt.jar (733ms)
> update
[info] Updating {file:/D:/mongoflight-master/}default-5db528...
[info] Resolving org.scala-lang#scala-library;2.10.0 ...
> reload
[info] Loading project definition from D:\mongoflight-master\project
[info] Set current project to Mongotracks (in build file:/D:/mongoflight-master/)
> container:start
[info] Started SelectChannelConnector@0.0.0.0:8080
[success] Total time: 12 s, completed Jul 22, 2013 11:11:08 PM
```

End Result...



App: Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
App: Home
localhost:8080
Google

app

- List Airport
- Create Airport
- Home
- Static Content

Welcome to your project!

Welcome to your Lift app at Mon Jul 22 23:15:47 EDT 2013

App: List
File Edit
App: Li

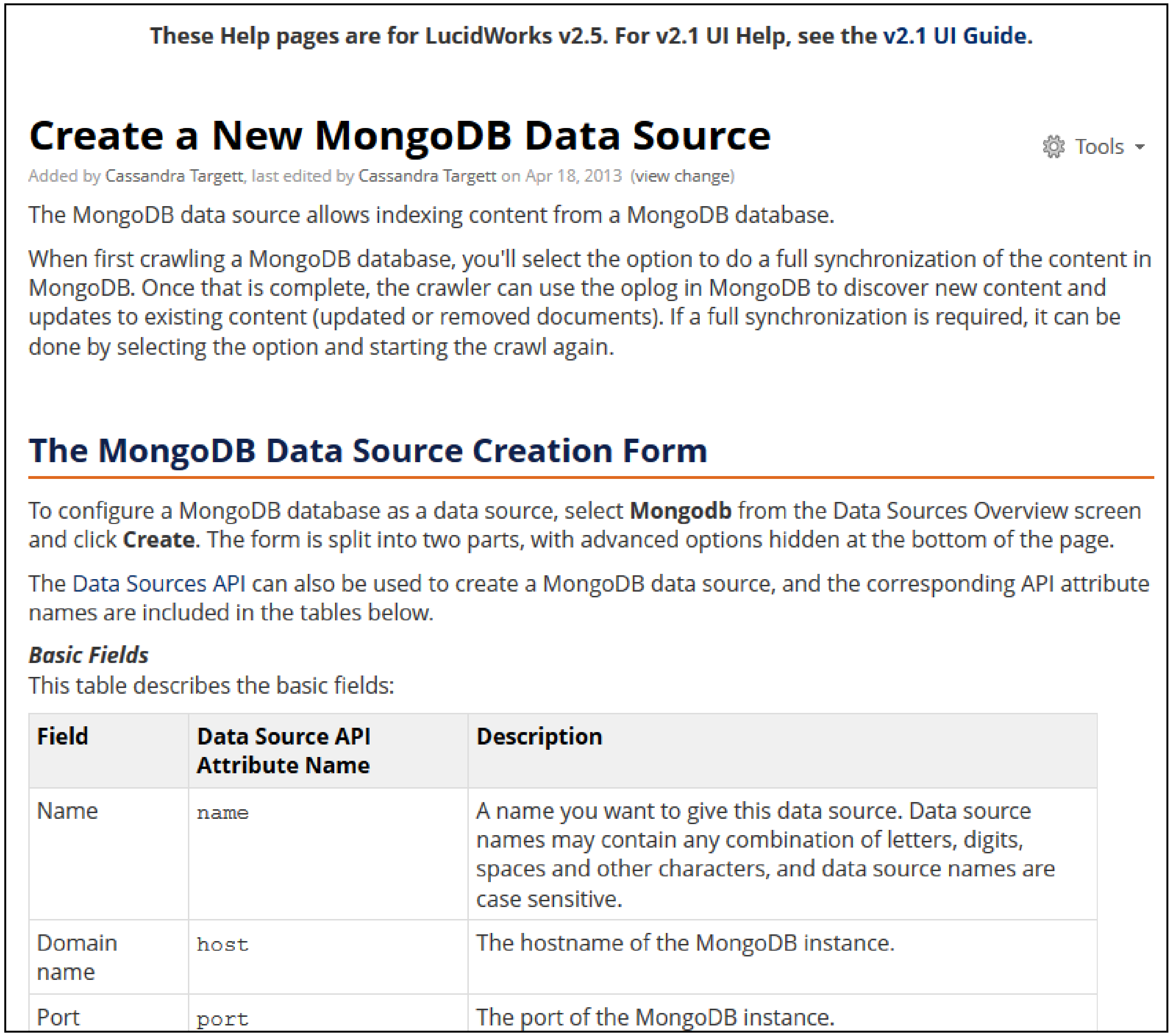
- List Airport
- Create Airport
- Home
- Static Content

altitude	dst	iatafaa	city	name	timezone	airportid	country			
0	0					0		View	Edit	Delete
1	0					1		View	Edit	Delete
2	0					2		View	Edit	Delete
3	0					3		View	Edit	Delete
4	0					4		View	Edit	Delete
5	0					5		View	Edit	Delete

Next Goal: Adding search – an example

A good experiment would be to configure Lucene/Solr to search for content in Mongo.

Lucid-works has a nice page describing how to do this with their release.



These Help pages are for LucidWorks v2.5. For v2.1 UI Help, see the v2.1 UI Guide.

Create a New MongoDB Data Source

Added by Cassandra Targett, last edited by Cassandra Targett on Apr 18, 2013 (view change)

The MongoDB data source allows indexing content from a MongoDB database.

When first crawling a MongoDB database, you'll select the option to do a full synchronization of the content in MongoDB. Once that is complete, the crawler can use the oplog in MongoDB to discover new content and updates to existing content (updated or removed documents). If a full synchronization is required, it can be done by selecting the option and starting the crawl again.

The MongoDB Data Source Creation Form

To configure a MongoDB database as a data source, select **Mongodb** from the Data Sources Overview screen and click **Create**. The form is split into two parts, with advanced options hidden at the bottom of the page.

The Data Sources API can also be used to create a MongoDB data source, and the corresponding API attribute names are included in the tables below.

Basic Fields
This table describes the basic fields:

Field	Data Source API Attribute Name	Description
Name	`name`	A name you want to give this data source. Data source names may contain any combination of letters, digits, spaces and other characters, and data source names are case sensitive.
Domain name	`host`	The hostname of the MongoDB instance.
Port	`port`	The port of the MongoDB instance.

http://docs.lucidworks.com/display/help/Create+a+New+MongoDB+Data+Source

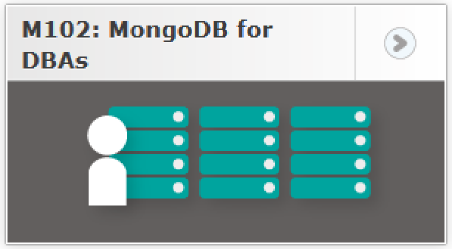
FYI: MongoDB Training

- 10gen offers training classes…check on-line
  - 2 day developer
  - One-on-One
- Also, they support free on-line course too:
  - https://education.10gen.com/

M101J: MongoDB for Java Developers

M101JS: MongoDB for Node.js Developers

M101P: MongoDB for Developers


M102: MongoDB for DBAs