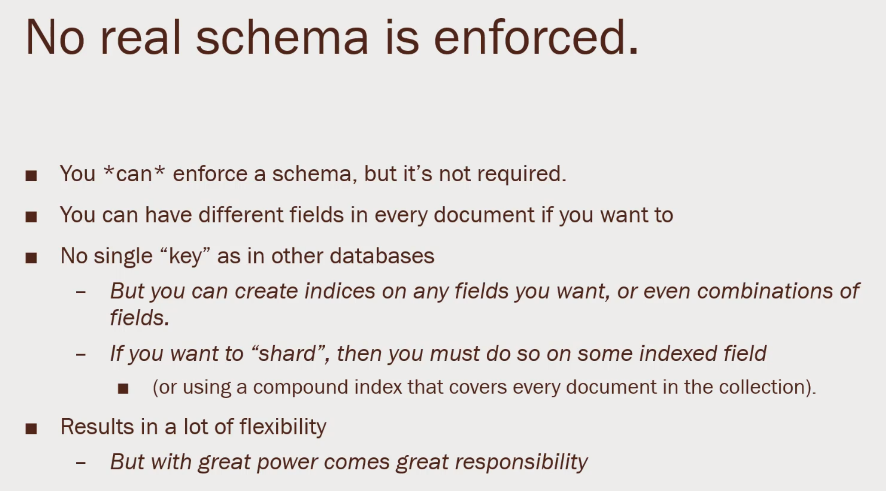
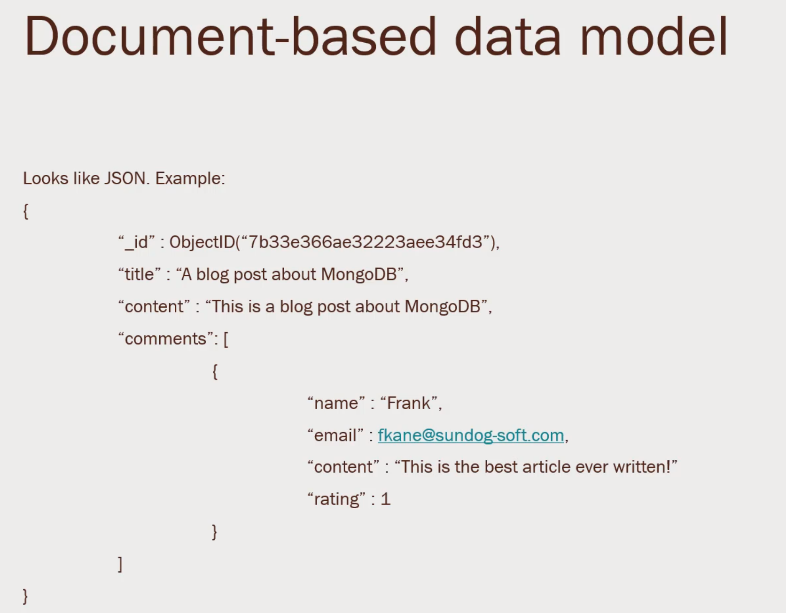
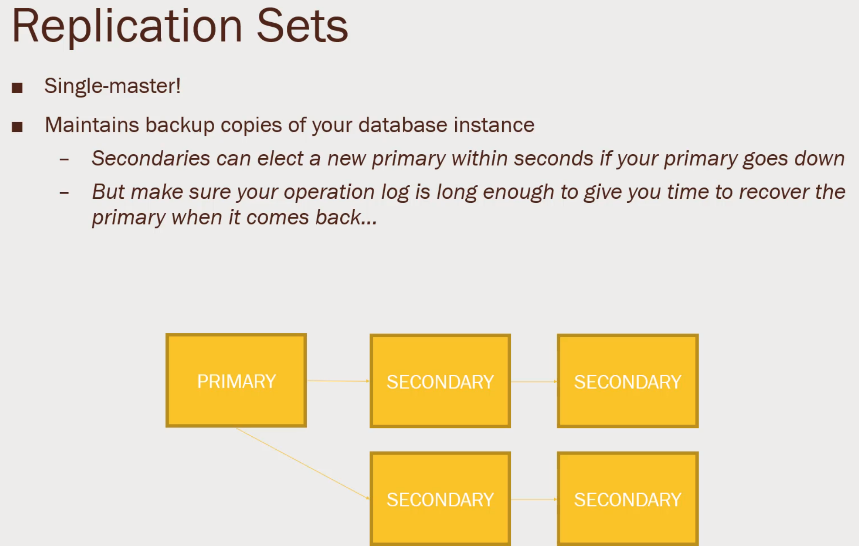
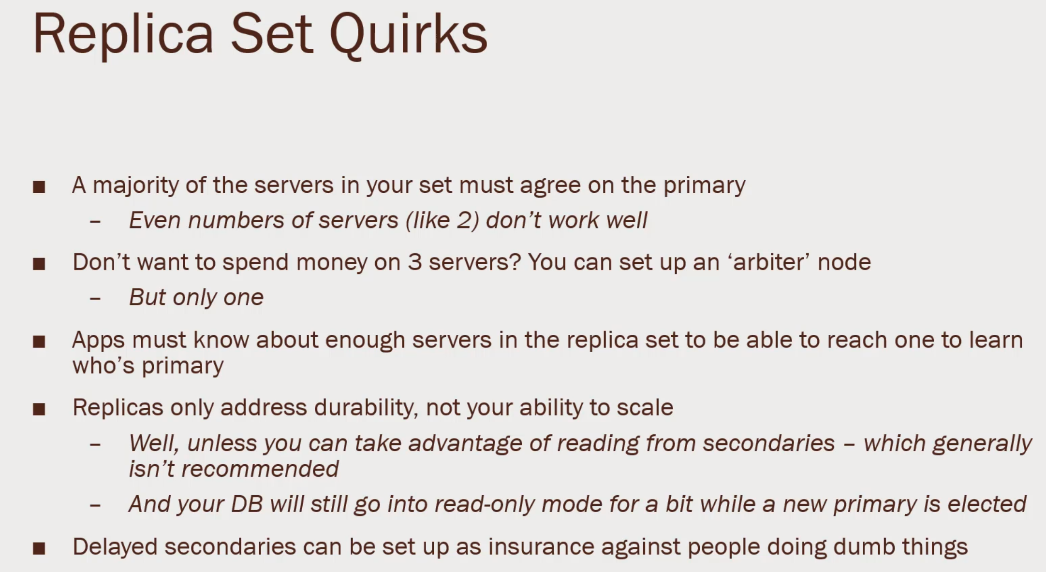
MongoDB – Managing HuMONGOus Data Consistency and Partion-Tolerance.

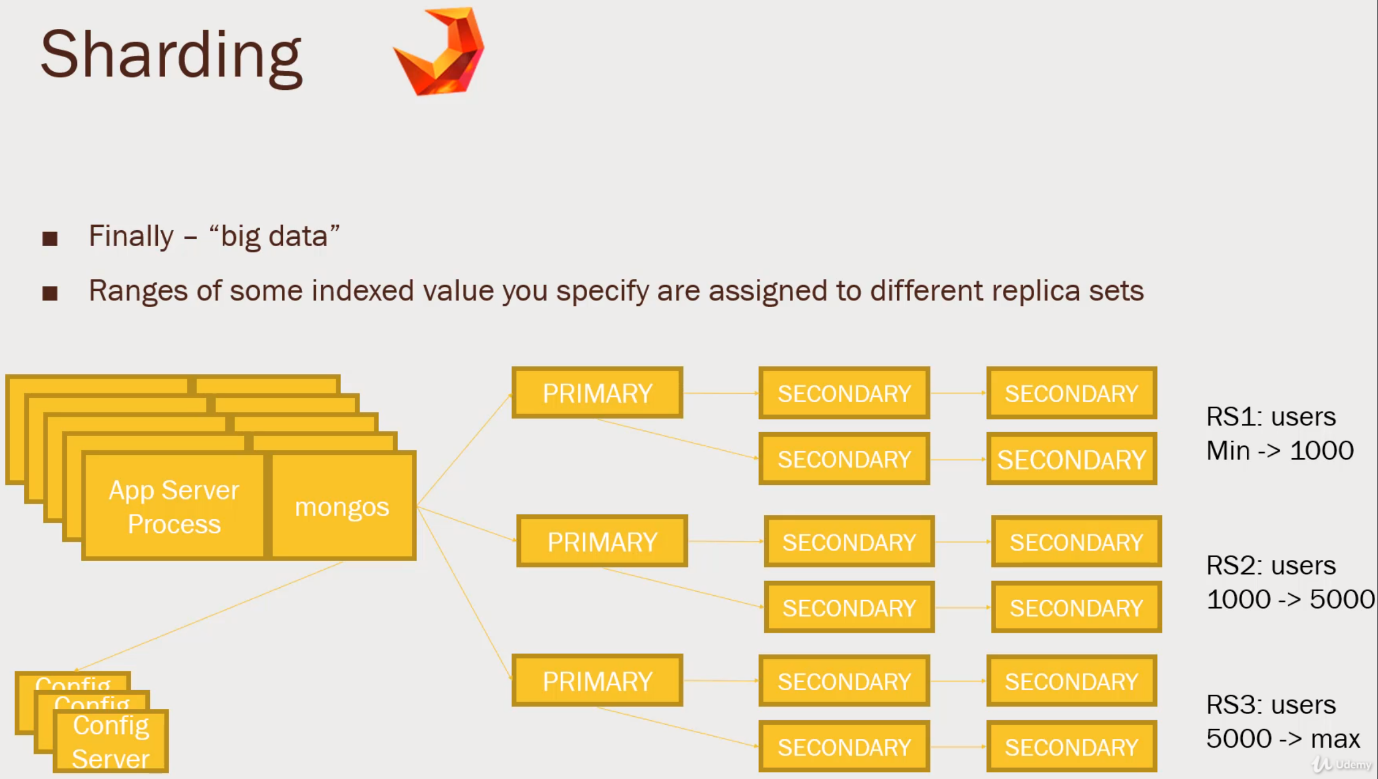
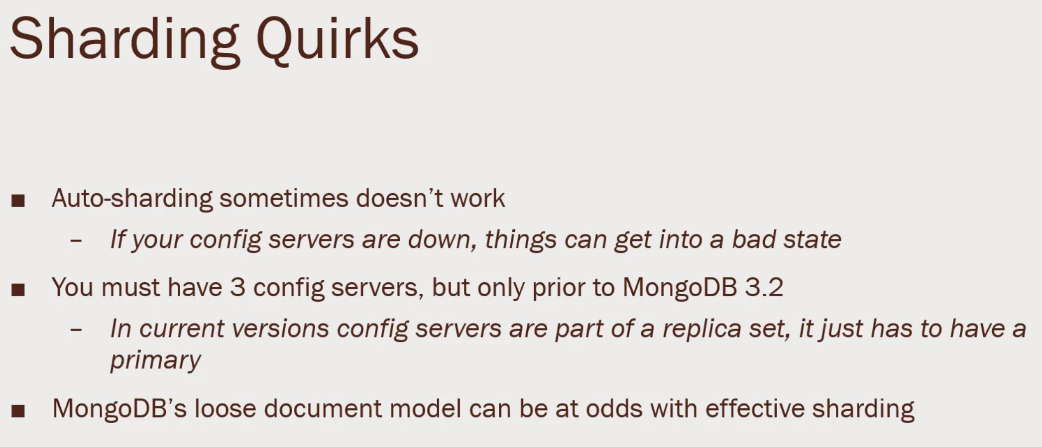
Commercial, its document data model.

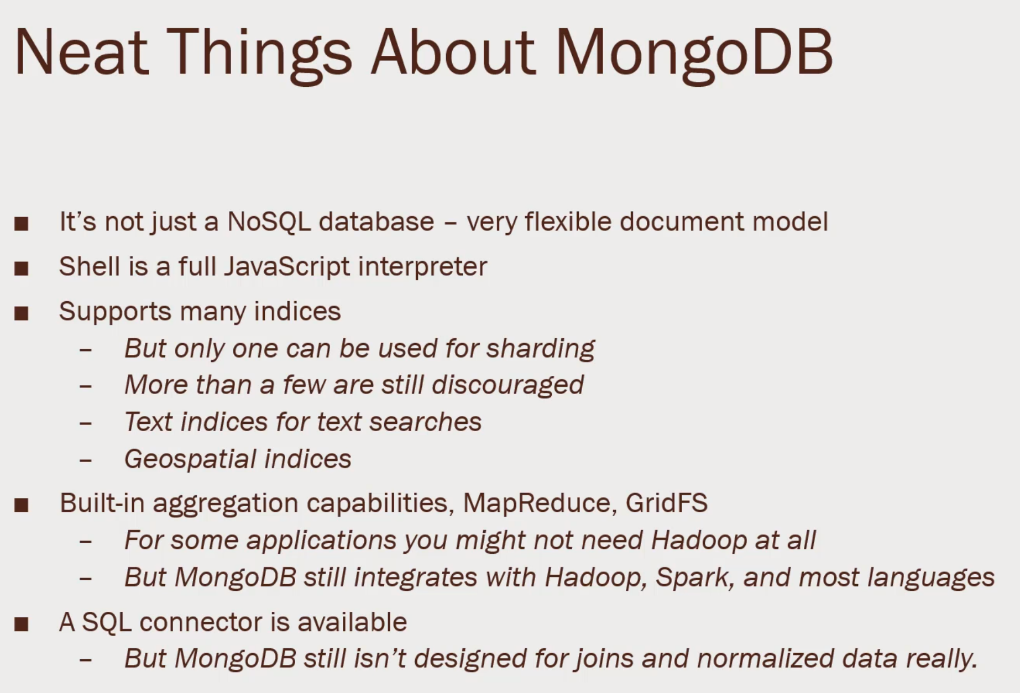
Has a Single Master. If it goes down, it has primary node, that allows reads but disables writes for the time that the single master went down.

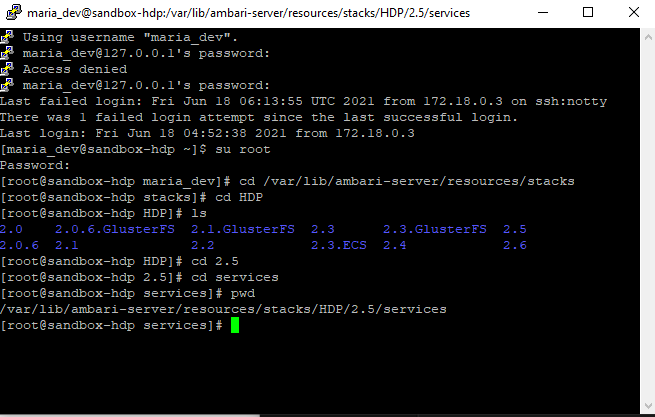
* Databases
* Collections
* Documents

Architecture. Need to ensure that the Primary is back up running quickly. Talking about back up copies.

MongoDB 3.6 - DNS Seedless connection format.

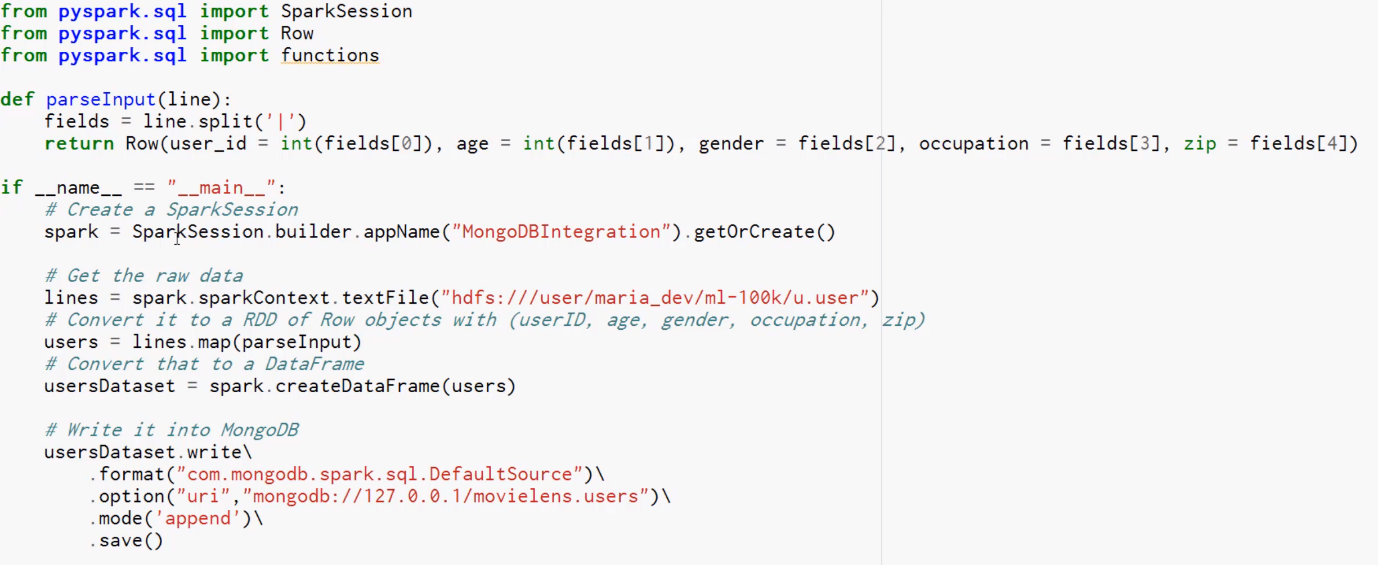
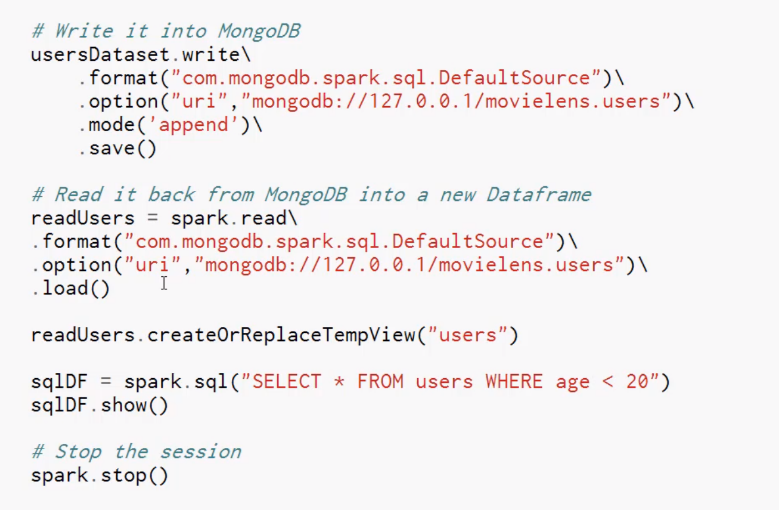
Zookeeper to maintain the services.



Connect to Putty. Set up MongoDB

Restart the Ambari service and Login to the web interface as Admin. Go to add services, MongoDB and install everything.

Install – pip install pymongo

Tie the dataset into mongoDB

Load into the dataframe.

To run the file,

Copy In the script, wget <http://media.sundog-soft.com/hadoop/MongoSpark.pyy>

Export SPARK\_MAJOR\_VERSION = 2

Spark-submit --packages org.mongodb.spark:mongo-spark-connector\_2.11:2.0.0 MongoSpark.py

Mongo Shell

Type in mongo, shell prompt

use movielens

db.users.find({user\_id: 100})

MongoDB doesn’t do setup index like Cassandra. To take a look at explanation, type in

db.users.explain().find({user\_id:100}) it does a scan, and see a Winning scan

Make an index

db.users.createIndex({user\_id: 1}) ,MongoDB will do an Index scan.

MongoDB Built in Aggregation function.

db.users.aggreagate([ $group: {\_id: { occupation: “$occupation”, avgAge: {$avg: ‘age’}}}

])

db.users.count()

db.getCollectionInfos()

db.users.drop()