MongoDB OpsManager

Contents

- Using OpsManager
- Architecture & Components
  - Automation
  - Monitoring
  - Backup
- Common issues
- More details on CloudManager
- Where are we going?

What, When and Why

- What:
  - “MongoDB Monitoring Service”, “MongoDB Management Service”
  - Tool to manage MongoDB deployment
  - Cloud and On-Prem versions
- When (Monitoring, Backup, Automation):
  - Cloud: 9/2011, 4/2013, 10/2014 (attach 2/2015)
  - On-Prem: 7/2013, 4/2014, 3/2015
- Why:
  - Ops are ready to pay for a better experience

OpsManager Version History

- Cloud Manager
  - New version every 3 weeks
- OpsManager
  - MMS 1.0 – 7/2013
  - MMS 1.5 – 9/2014
  - OpsManager 1.6 – 3/2015
  - OpsManager 1.8 – 5/2015
  - OpsManager 2.0 - 12/2015

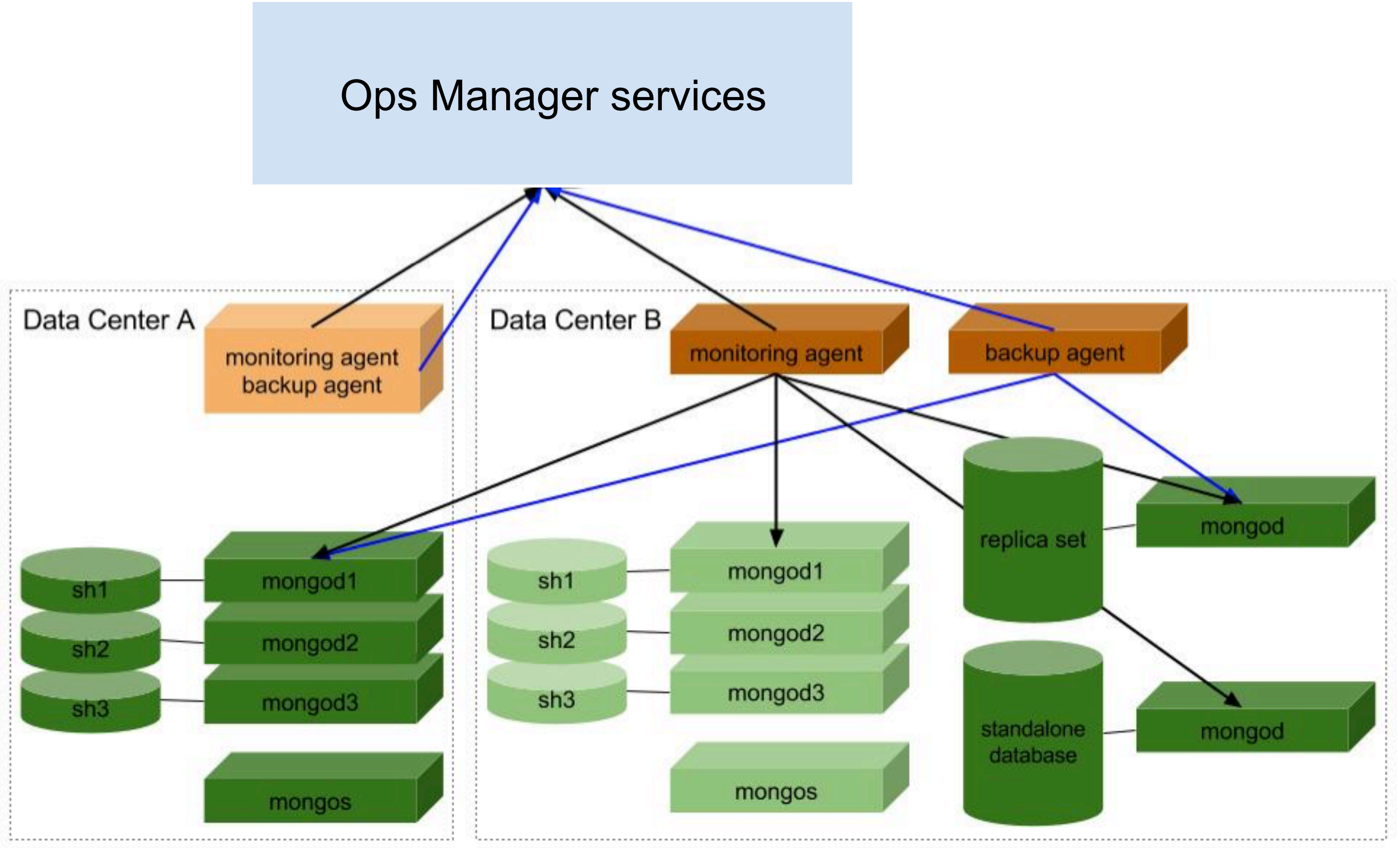
CloudManager vs OpsManager

CloudManager	**OpsManager**
• free or low cost	• must have a MongoDB Advanced Enterprise Subscription
• buy few additional hosts	• buy many additional hosts
• backups at $2.50/GB/month	• backups at $X
• easy to install	• hard to install
• maintained by experts (us)	• maintenance by local IT
• data is outside org	• data is inside org

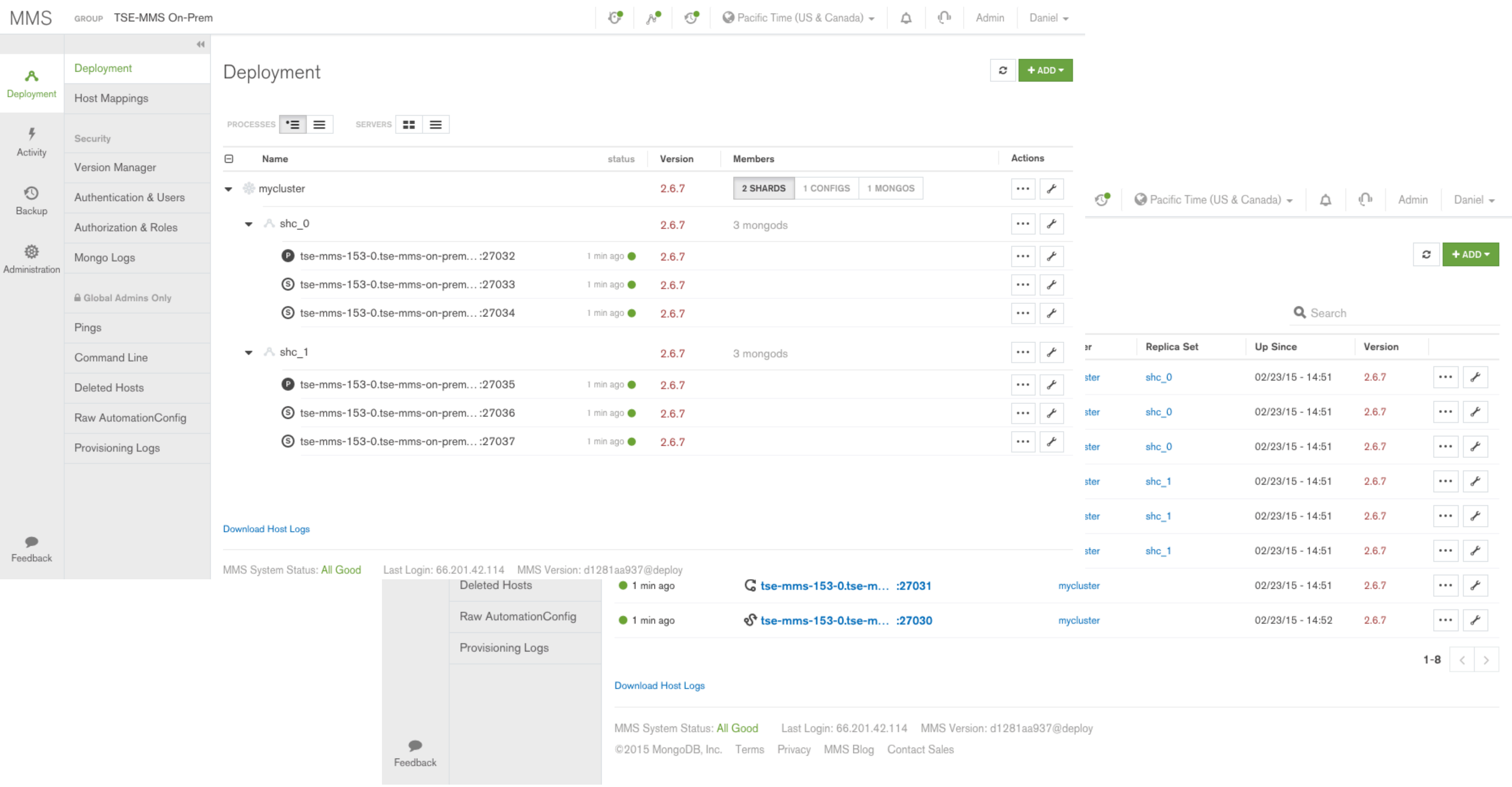
Using Ops Manager

- Create
  - A logical group for a set of clusters and users
- Install:
  - One automation agent per host
  - One monitoring agent per Ops manager group
  - One backup agent per Ops manager group
    - Not released officially, but you can use more than one backup agent to distribute the load
  - Additional monitoring/backup agents are in 'stand-by'
- The agents always initiate the communication to the Ops manager Server

Monitoring and Backup Agents



Monitoring UI



Deployment

Name	status	Version	Members	Actions
mycluster		2.6.7	2 SHARDS 1 CONFIGS 1 MONGOS	
shc_0		2.6.7	3 mongods	
tse-mms-153-0.tse-mms-on-prem... :27032	1 min ago ●	2.6.7		
tse-mms-153-0.tse-mms-on-prem... :27033	1 min ago ●	2.6.7		
tse-mms-153-0.tse-mms-on-prem... :27034	1 min ago ●	2.6.7		
shc_1		2.6.7	3 mongods	
tse-mms-153-0.tse-mms-on-prem... :27035	1 min ago ●	2.6.7		
tse-mms-153-0.tse-mms-on-prem... :27036	1 min ago ●	2.6.7		
tse-mms-153-0.tse-mms-on-prem... :27037	1 min ago ●	2.6.7		

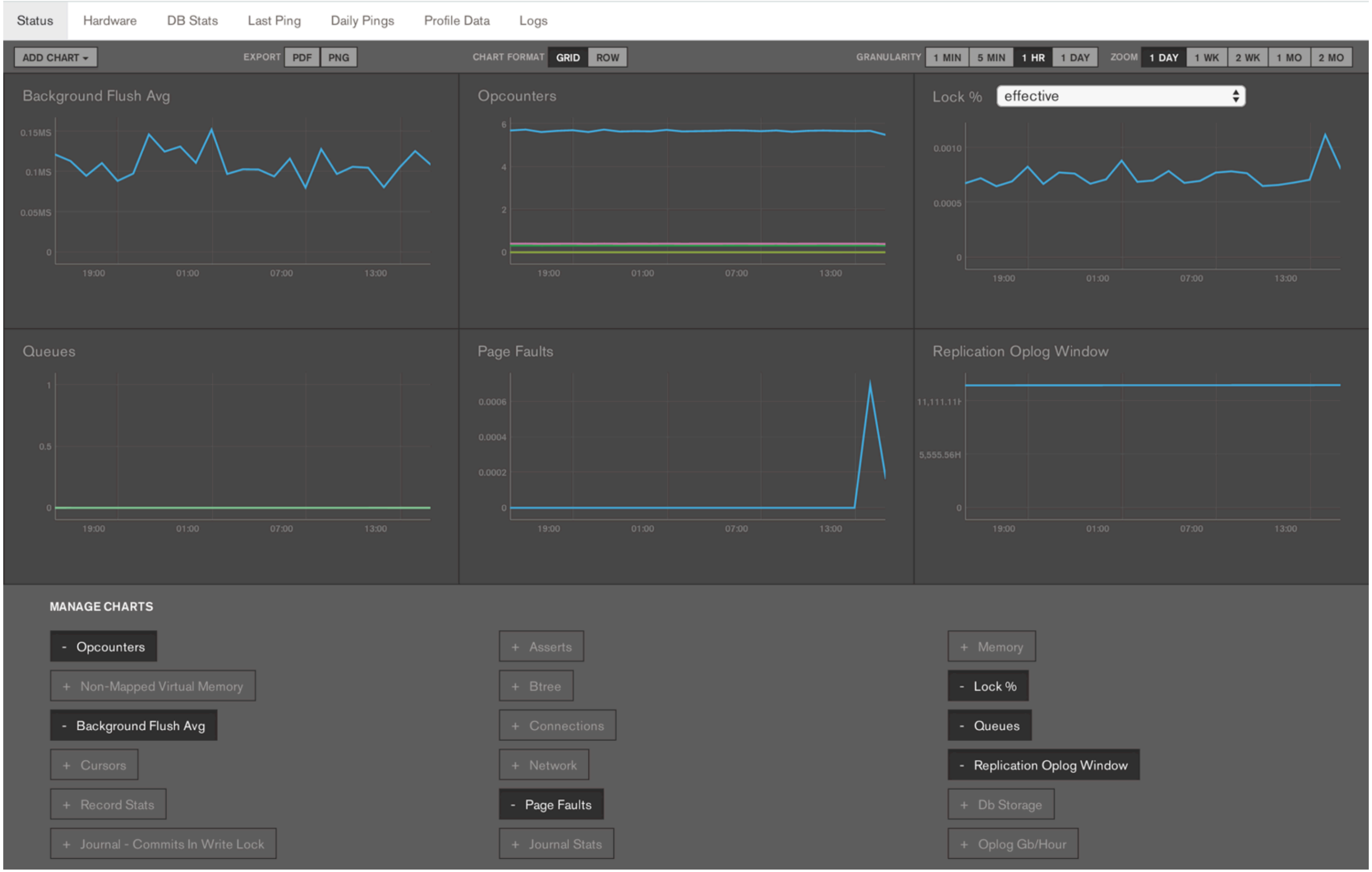
MMS System Status: All Good   Last Login: 66.201.42.114   MMS Version: d1281aa937@deploy

		...er	Replica Set	Up Since	Version
		ster	shc_0	02/23/15 - 14:51	2.6.7
		ster	shc_0	02/23/15 - 14:51	2.6.7
		ster	shc_0	02/23/15 - 14:51	2.6.7
		ster	shc_1	02/23/15 - 14:51	2.6.7
		ster	shc_1	02/23/15 - 14:51	2.6.7
		ster	shc_1	02/23/15 - 14:51	2.6.7
● 1 min ago	tse-mms-153-0.tse-m... :27031	mycluster		02/23/15 - 14:51	2.6.7
● 1 min ago	tse-mms-153-0.tse-m... :27030	mycluster		02/23/15 - 14:52	2.6.7

MMS System Status: All Good   Last Login: 66.201.42.114   MMS Version: d1281aa937@deploy

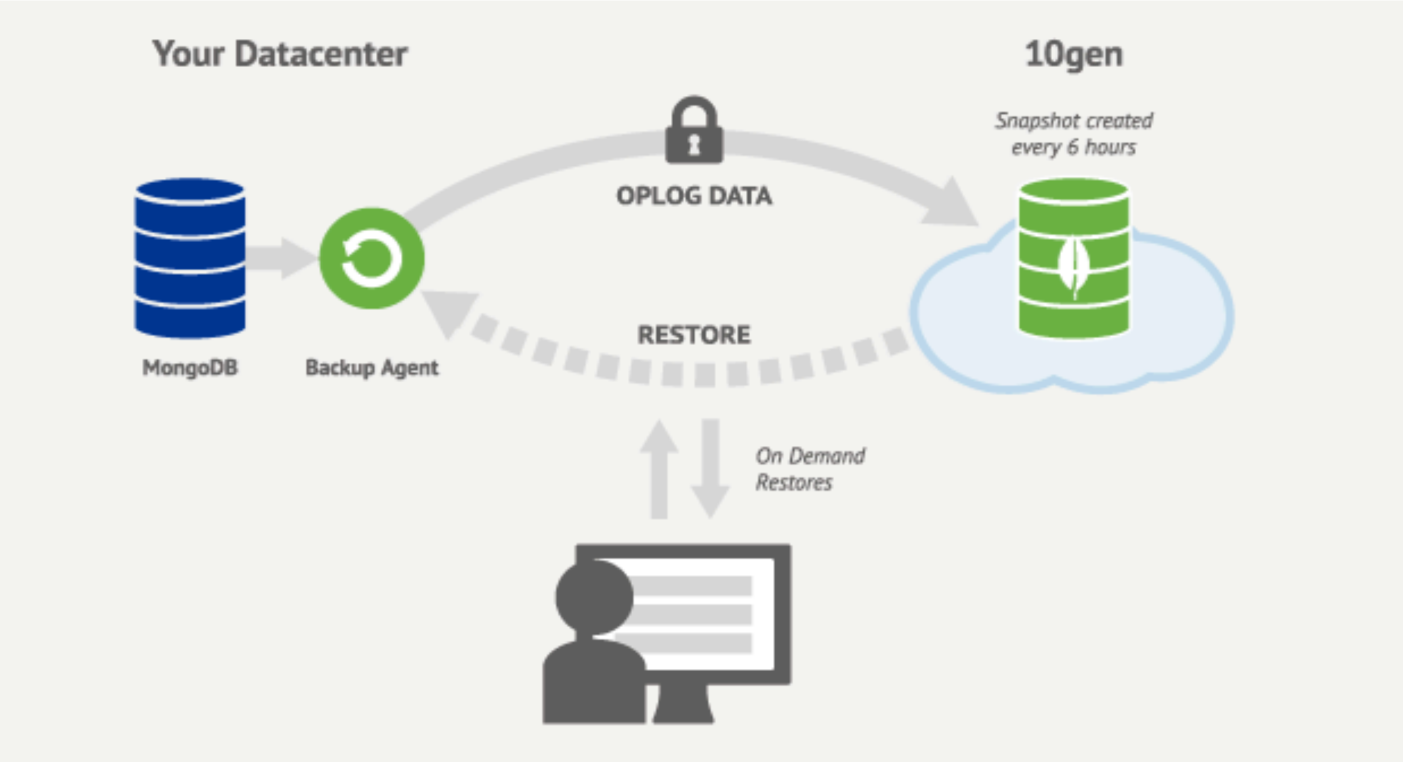
©2015 MongoDB, Inc.   Terms   Privacy   MMS Blog   Contact Sales

Monitoring UI



Backup with Ops Manager

- Easy to install
- Slow restores for large datasets



Backup with OpsManager

- Difficult to install
- Require a “sizing exercise” with customer

Restores

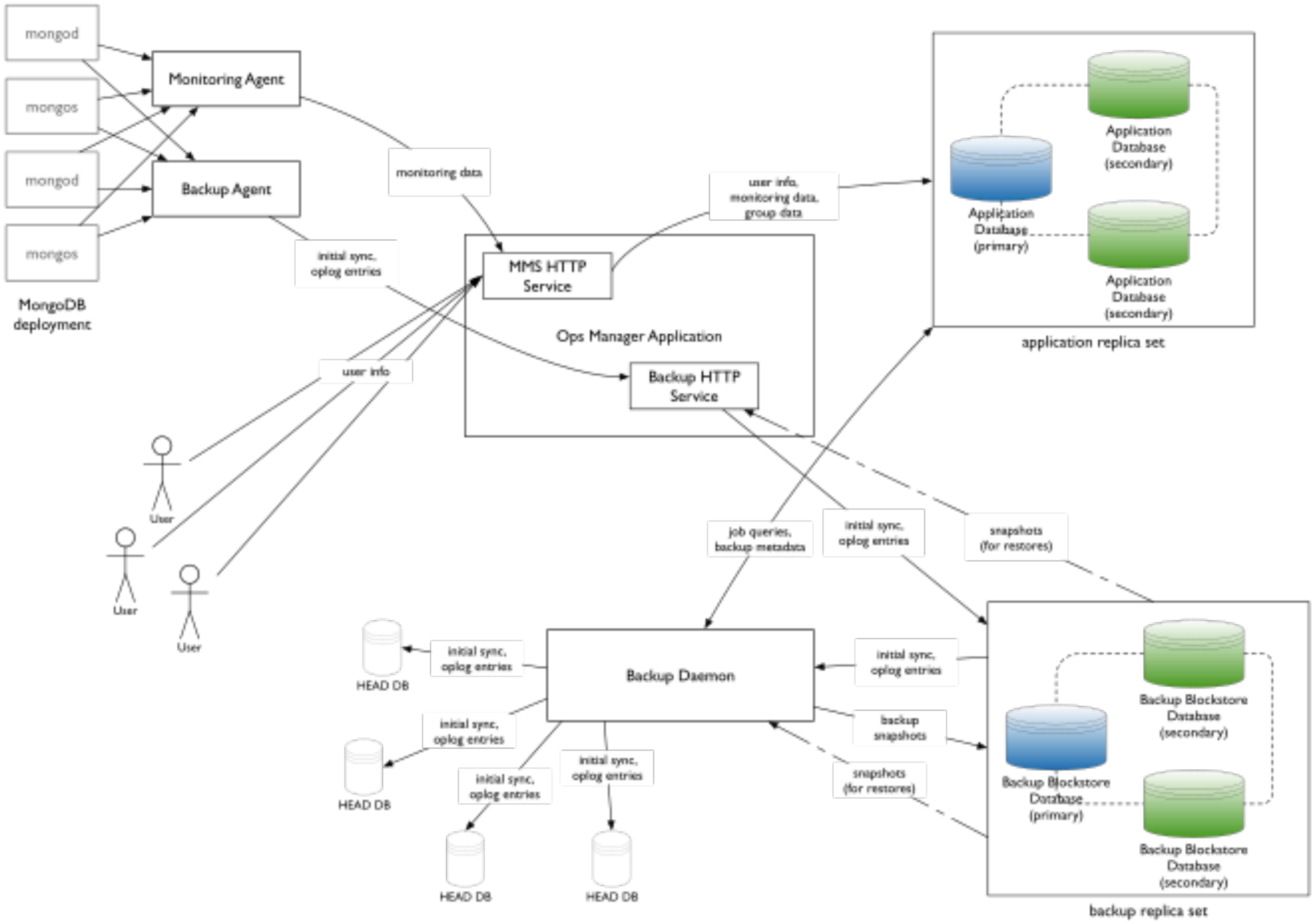
- Restore an existing Snapshot
  - SCP tar.gz to customer's host (compressed)
  - SCP data files to customer's host (uncompressed on disk, compressed over the wire)
  - One-time use download link of tar.gz
- Point-in-Time
  - of replica set. Within 24 hours (configurable up to 96 hours in on-prem).
  - of cluster at predefined checkpoints.
- Ship a script to add restore into an existing replica set via a seeded oplog.
- If cluster under 'Automation'

  => restore done for you

Comparing restore modes

- HTTPS pull restore is slowest transfer speeds, but is available almost immediately (assuming not-PIT restore)
- SCP individual files is the fastest transfer speeds and is available almost immediately (again assuming non-PIT), but needs the most space on the receiving disk
- SCP tar.gz is fast transfer speeds (on par with individual files), but the snapshot has to be assembled on our side, which is slow according to the snapshot size, but the recieving disk can be a little smaller

Architecture of Ops Manager


mongod
mongos
mongod
mongos
MongoDB deployment
Monitoring Agent
Backup Agent
monitoring data
initial sync, oplog entries
user info
User
User
User
MMS HTTP Service
Ops Manager Application
Backup HTTP Service
user info, monitoring data, group data
Application Database (secondary)
Application Database (primary)
Application Database (secondary)
application replica set
job queries, backup metadata
initial sync, oplog entries
snapshots (for restores)
Backup Daemon
HEAD DB
initial sync, oplog entries
initial sync, oplog entries
HEAD DB
initial sync, oplog entries
HEAD DB
initial sync, oplog entries
HEAD DB
initial sync, oplog entries
backup snapshots
snapshots (for restores)
Backup Blockstore Database (secondary)
Backup Blockstore Database (primary)
Backup Blockstore Database (secondary)
backup replica set

Ops Manager components

Monitoring/Automation

- Monitoring Agent
- Automation Agent
- Application Server
  - UI
  - Backup HTTP
  - Backup Alert
- Application Database (mongod)

… plus Backup

- Backup Agent
- Backup Daemon
- Blockstore Database
  - Blockstore DB
  - Sync DB
  - Oplog DB

The Application and App DB

- “The UI”
- The monitoring data
- The settings
- The users, permissions
- The administration functions/panel
- The groups

Backup Block store

- “The Bank”
- Backups/snapshots of Customer Data
- Stored in blocks
  - compressed
  - de-duplicated
- Periodically groomed (a process that goes through and compresses/deletes/reclaims space)
- New in 2.0, alternatively you can specify a file system location

Backup Block store vs File System

- New in 2.0
- Can use Blockstore for a cluster and FileSystem for another cluster
- Snapshots for a cluster go to a single storage unit (one FileSystem store or one Blockstore)
  - File path for file system store, must manage manually
- Choosen at assignment time, if both types active, FileSystem store has priority.

File System Store example

```
•   [mms-user@tse-opsmgr-200-0 rs1]$ pwd
•   /fsstore/568321cbe4b0b7a9401cde5e/rs1

•   [mms-user@tse-opsmgr-200-0 rs1]$ ls -al
•   total 20
•   drwxr-xr-x 5 mongodb-mms mongodb-mms 4096 Feb 12 15:00 .
•   drwxr-xr-x 3 mongodb-mms mongodb-mms 4096 Feb 12 02:01 ..
•   drwxr-xr-x 2 mongodb-mms mongodb-mms 4096 Feb 12 02:01 56bd3d15e4b0ae3917de7b1c
•   drwxr-xr-x 2 mongodb-mms mongodb-mms 4096 Feb 12 09:00 56bd9f11e4b0ae3917de7e60
•   drwxr-xr-x 2 mongodb-mms mongodb-mms 4096 Feb 12 15:00 56bdf371e4b0ae3917de8139

•   [mms-user@tse-opsmgr-200-0 rs1]$ ls -al 56bd3d15e4b0ae3917de7b1c/
•   total 52
•   drwxr-xr-x 2 mongodb-mms mongodb-mms 4096 Feb 12 02:01 .
•   drwxr-xr-x 5 mongodb-mms mongodb-mms 4096 Feb 12 15:00 ..
•   -rw-r--r-- 1 mongodb-mms mongodb-mms 2224 Feb 12 02:01 collection-0--2722514553113671915.wt.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms  375 Feb 12 02:01 index-1--2722514553113671915.wt.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms 4020 Feb 12 02:01 manifest
•   -rw-r--r-- 1 mongodb-mms mongodb-mms  384 Feb 12 02:01 _mdb_catalog.wt.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms  433 Feb 12 02:01 sizeStorer.wt.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms   96 Feb 12 02:01 storage.bson.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms   60 Feb 12 02:01 WiredTiger.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms   51 Feb 12 02:01 WiredTigerLAS.wt.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms   41 Feb 12 02:01 WiredTiger.lock.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms  528 Feb 12 02:01 WiredTiger.turtle.gz
•   -rw-r--r-- 1 mongodb-mms mongodb-mms 1474 Feb 12 02:01 WiredTiger.wt.gz
```

The Backup Daemon

- “The Workhorse”
- Copies backups from the backup agent to the blockstore
- You need at least one of them
- Creates point in time restore files
- Is resource intensive
- Requires 2-3x data space, as it holds two copies of the customer data when generating a Point In Time (PIT) Recovery file

The Head DB

- “The Hidden Replica”
- Copy of the replica set under backup
- Kept up to date by applying the oplogs on it
- Not part of a package, the Backup Daemon installs, starts and stops this mongod process
- Uses a similar MongoDB version as the one used by the replica

The Backup Agent

- “The Streamer”
- Streams the data to the backup service (‘oplog’). For a sharded cluster, the Backup Agent tails the primary of each shard and each config server.
- The agent ships initial sync and oplog data over HTTPS back to the Ops Manager Backup service.
- You need one per group
- Must have access to all members of the deployment

The Monitoring Agent

- "The Watcher"
- Watches the deployment
- Records performance statistics
- Tells the backup agent and backup components 'what exists' and the health of the system
- Must have access to all members of the deployment

You cannot use backup without monitoring.
You cannot use backup without a monitoring agent.

The Automation Agent

- “The Butler”
- Algorithm
  - Retrieves a “goal configuration”
  - Looks a differences
  - Build plans of steps
  - Apply differences until reaches plan
- Side effects
  - Uses it own repository of versions
  - Will revert some changes done out of the UI:
    - rs.conf
    - users

Backup Initial Sync

1. Transfering - data from all collections is sent to the backup server
2. Building - a new mongod is started and all data from step 1 is inserted
3. Apply Oplogs - Phase 1 - All ops that happened during step 1 are applied and any missed documents (documents missed due to moves) are identified
4. Fetching Missing Documents - fetch and insert all docs identified in step 3
5. Apply Oplogs - Phase 2 - All ops that happened between the end of step 1 and the end of step 4 are applied
6. Creating Indexes - all indexes are created.

 - replace 'nodes' by 'hostname' for anything regarding licensing

Daemon

- Owns backup for multiple replica sets
- Work
  - Find the replica set with most critical work pending
  - Create initial head DB
    - Apply sync slices
    - Apply oplogs
    - Create snapshot
  - ApplyOps
    - Start mongod
    - Read oplog data from Oplog Data DB
    - Apply ops to mongod
    - Stop mongod
  - Snapshot
    - Read data files (actual mongod data files)
    - De-dupe (file level then block level)
    - Save

Saving space in the Blockstore

- Slice the files in chunks
- De-duplication of identical chunks
- Grooming
  - Garbage collection like process
    1. walk back the used chunks for a set of snapshots for a replica
    2. …making a copy of the chunk in a new collection
    3. once done, remove the previous collection
  - Needs temp disk space as big as the replica set to groom

Sharded Cluster

- Same constraints as doing backups with mongodump or LVM
- Cluster Snapshot (aka Clustershot)
  - Backup Agent stops the balancer and waits for migrations to finish (1 hour timeout)
  - Backup Agent inserts a "no-op" token into the oplog of all shards and config servers
  - Backup Agent restarts the balancer
  - Backup Daemon looks for tokens
  - Failure to stop the balancer or reach some shards results in an "inconsistent" clustershot

Sizing exercise (1/3)

- USE CASE:
  1. Do you intend to use Ops Manager only for Monitoring? If yes, skip the next two questions.
  2. Do you intend to use Ops Manager for Automation?
  3. Do you intend to use Ops Manager for Backups?
  4. Is this for a production environment?
  5. Is this a centralised deployment for multiple projects and environments?

Sizing exercise (2/3)

- ENVIRONMENT:
  1. Will you be using LDAP or other security options like Kerberos, x509, or SSL?
  2. Do you have root access for the servers on which you will be installing the application?
  3. Do the servers hosting the Ops Manager application have access to the outside internet in order to download the latest versions of MongoDB/agents? Please describe any external or internal firewall constraints
  4. Is there a standard machine you would like/need to use with this deployment? If so, what are the CPU/RAM/Disk (type, size, and disk configuration) specifications?
  5. Would you like your Ops Manager deployment within multiple data centers?
  6. How many replica sets and sharded clusters will be monitored/managed by Ops Manager?
  7. How much HA/redundancy do you want in the deployment? A typical production deployment has three Ops Manager Application database servers and another three Blockstore database servers for Ops Manager Backup (if used).

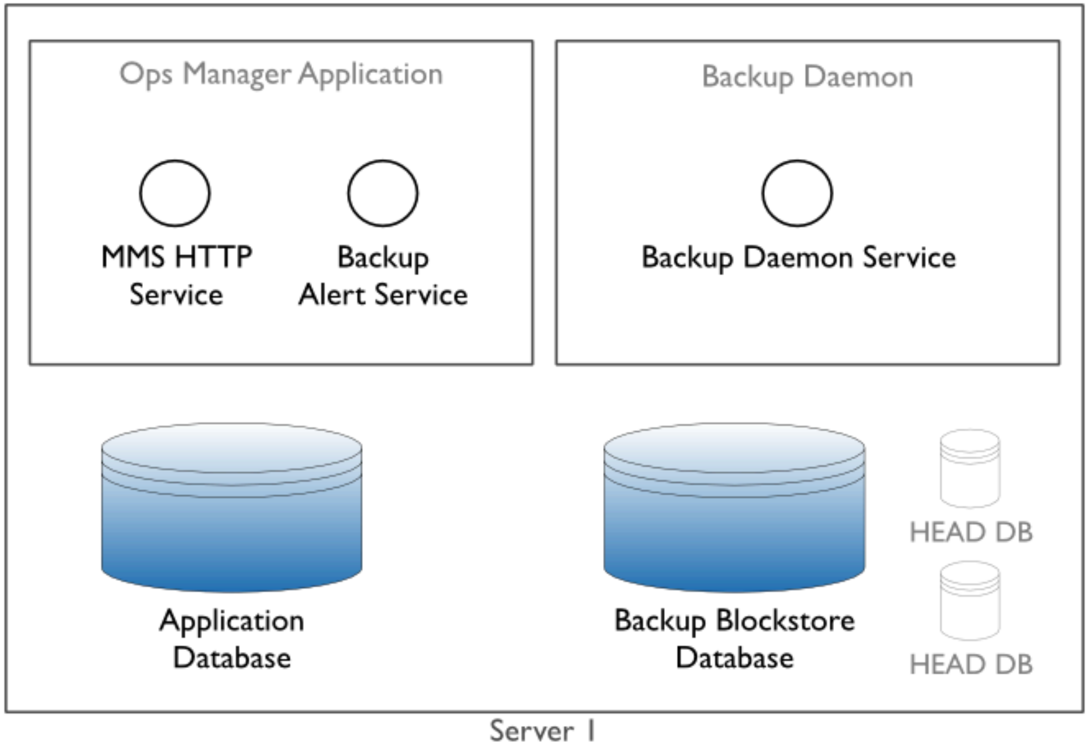
Sizing exercise (3/3)

- BACKUPS (IF APPLICABLE):
  1. How many replica sets and sharded clusters will be backed up by Ops Manager?
  2. For each replica set and shard, what is the:
  3. Oplog/day per replica set (GB)
  4. File size per replica set (GB)
  5. How compressible is the data? Is it text, videos, binaries?
  6. What is your expected growth over the next 6, 12, and 18 months? If it is somewhat difficult to procure hardware at your organization, we recommend sizing up for more than you will need.
  7. Retention:
  8. What is the Point In Time Restore requirement in hours? (default is 48)
  9. What is the required number of Daily, Weekly, and Monthly snapshots to keep? (default is 19)
  10. What is the required number of hourly snapshots to keep? (default is 8)
  11. What is the required time between hourly snapshots? (default is 6 hrs)?
  12. Do you desire high availability, a minimal configuration, or something in between?
  13. Point In Time Restore Requirements (hours)
  14. How long would you like to store the oldest snapshot? We are able to store information for up to 13 months, however, the retention period for snapshots can dramatically affect your total data size.
  15. Our default settings allow for snapshots every 6 hours (4 per day), would you like more or less frequently than this?

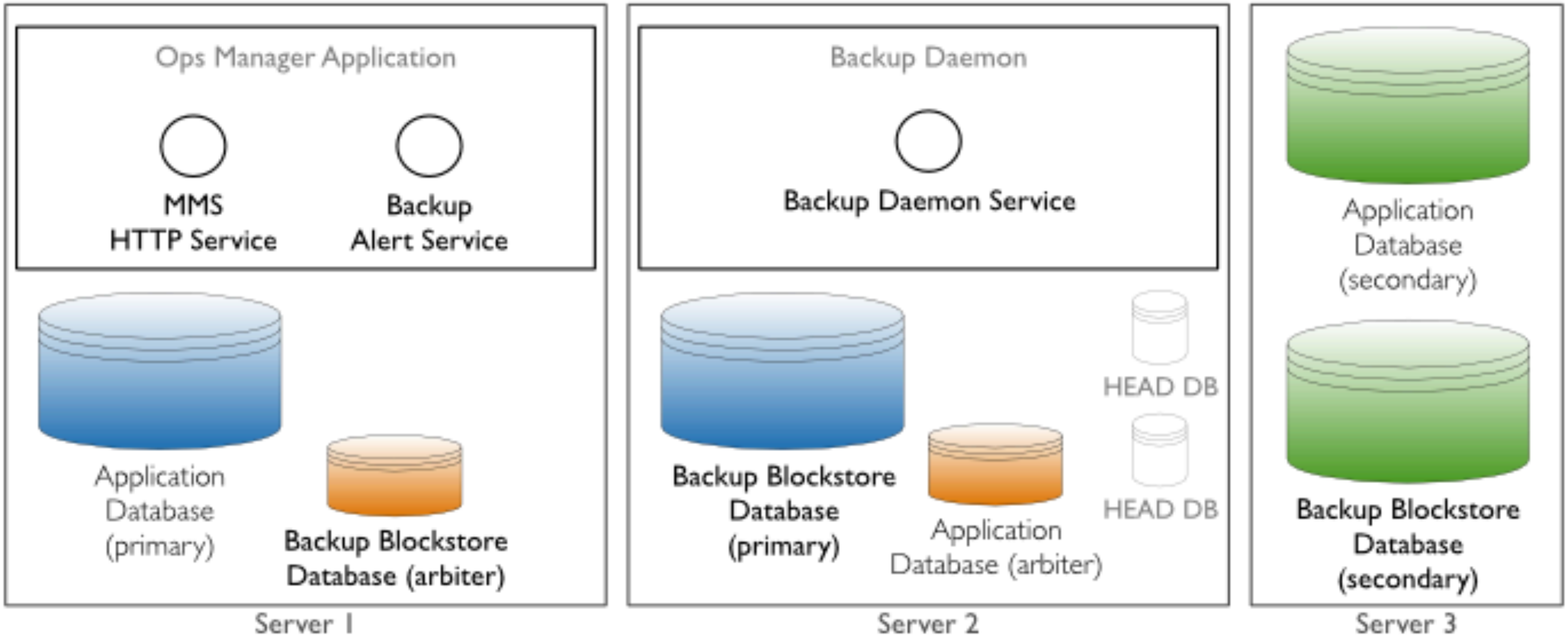
Ops Manager Backup Calculator

- To run monitoring, you need
  - MMS Application requires 15GB RAM and 4GB disk
  - MMS Application Database requires 8GB RAM and 200GB disk minimum
  - A monitoring agent (little resources)
- To run backup you, need:
  - backup daemon
    - with 2-3x disk space of the data being backed up
    - blockstore size(GB)/125 => X GB RAM
  - A blockstore with 2-3x the disk space of the data being backed up
  - A backup agent

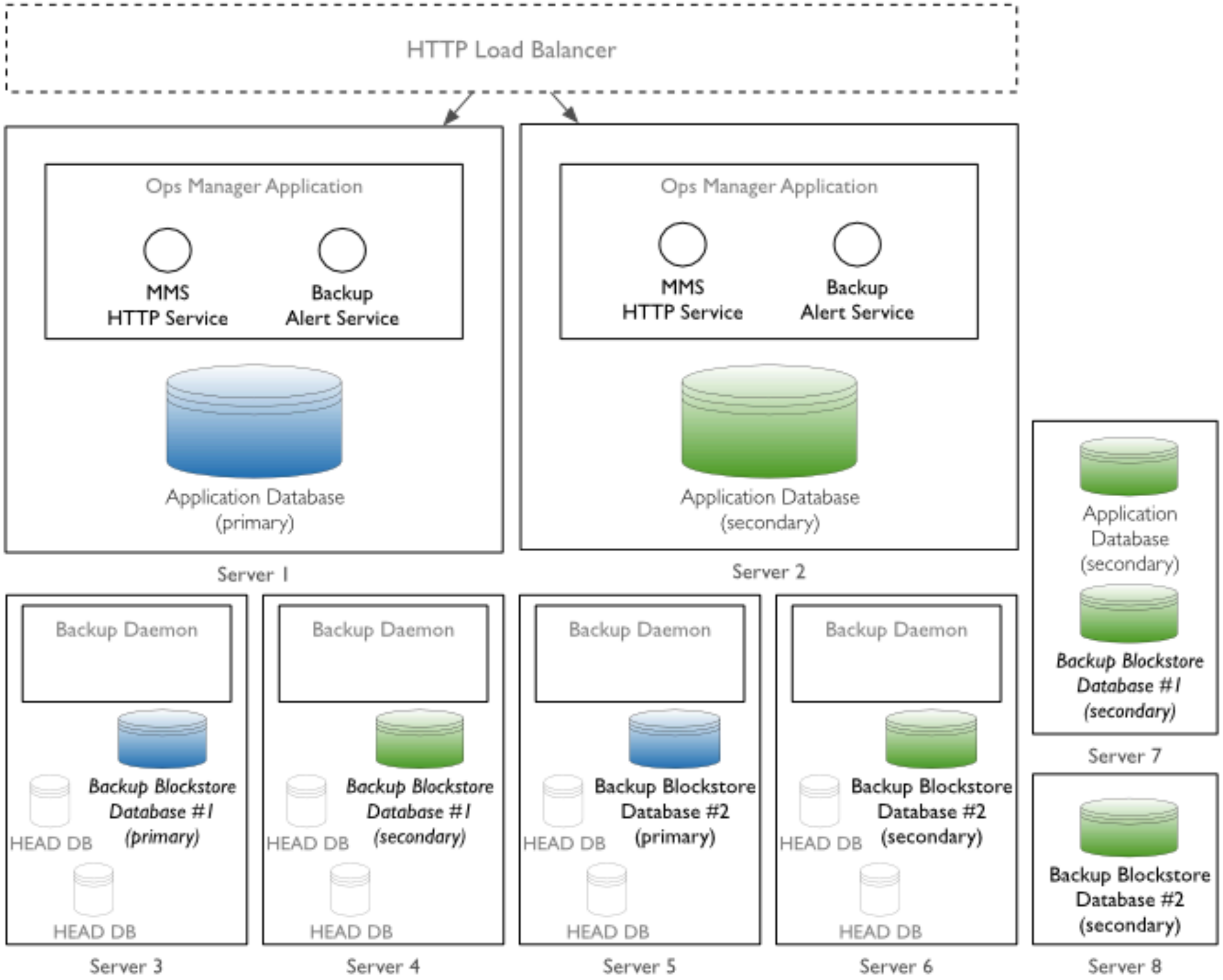
Evaluation deployment of Ops Manager



Basic deployment of Ops Manager


Ops Manager Application
MMS HTTP Service
Backup Alert Service
Application Database (primary)
Backup Blockstore Database (arbiter)
Server 1
Backup Daemon
Backup Daemon Service
HEAD DB
HEAD DB
Backup Blockstore Database (primary)
Application Database (arbiter)
Server 2
Application Database (secondary)
Backup Blockstore Database (secondary)
Server 3

More replica sets to backup


HTTP Load Balancer
Ops Manager Application
MMS HTTP Service
Backup Alert Service
Application Database (primary)
Server 1
Ops Manager Application
MMS HTTP Service
Backup Alert Service
Application Database (secondary)
Server 2
Application Database (secondary)
Backup Blockstore Database #1 (secondary)
Server 7
Backup Daemon
HEAD DB
Backup Blockstore Database #1 (primary)
HEAD DB
Server 3
Backup Daemon
HEAD DB
Backup Blockstore Database #1 (secondary)
HEAD DB
Server 4
Backup Daemon
HEAD DB
Backup Blockstore Database #2 (primary)
HEAD DB
Server 5
Backup Daemon
HEAD DB
Backup Blockstore Database #2 (secondary)
HEAD DB
Server 6
Backup Blockstore Database #2 (secondary)
Server 8

Ops Manager API

- Only approved way to programmatically access the Ops Manager Databases
  - DB Schema is likely to change
- RESTful
- Get a set of keys to access the API

Common Issues – restrictions

- Healthy Deployment
- All replica sets within the same group must have different names

Common Issues – Getting Started

- Monitoring topology incorrect or incomplete
  - E.g. hosts deactivated, no mongos, topology wrong
- Undersize hardware for backups
- Automation hanging
- Deployment unstable

Common Issues – Backup Agent

- Backup agent can't connect to mongods
  - E.g. firewall/connectivity problem, wrong auth
- Agent can't keep up
  - Underpowered machine
  - Oplog window too small
- Running multiple agents
  - Only one should be primary

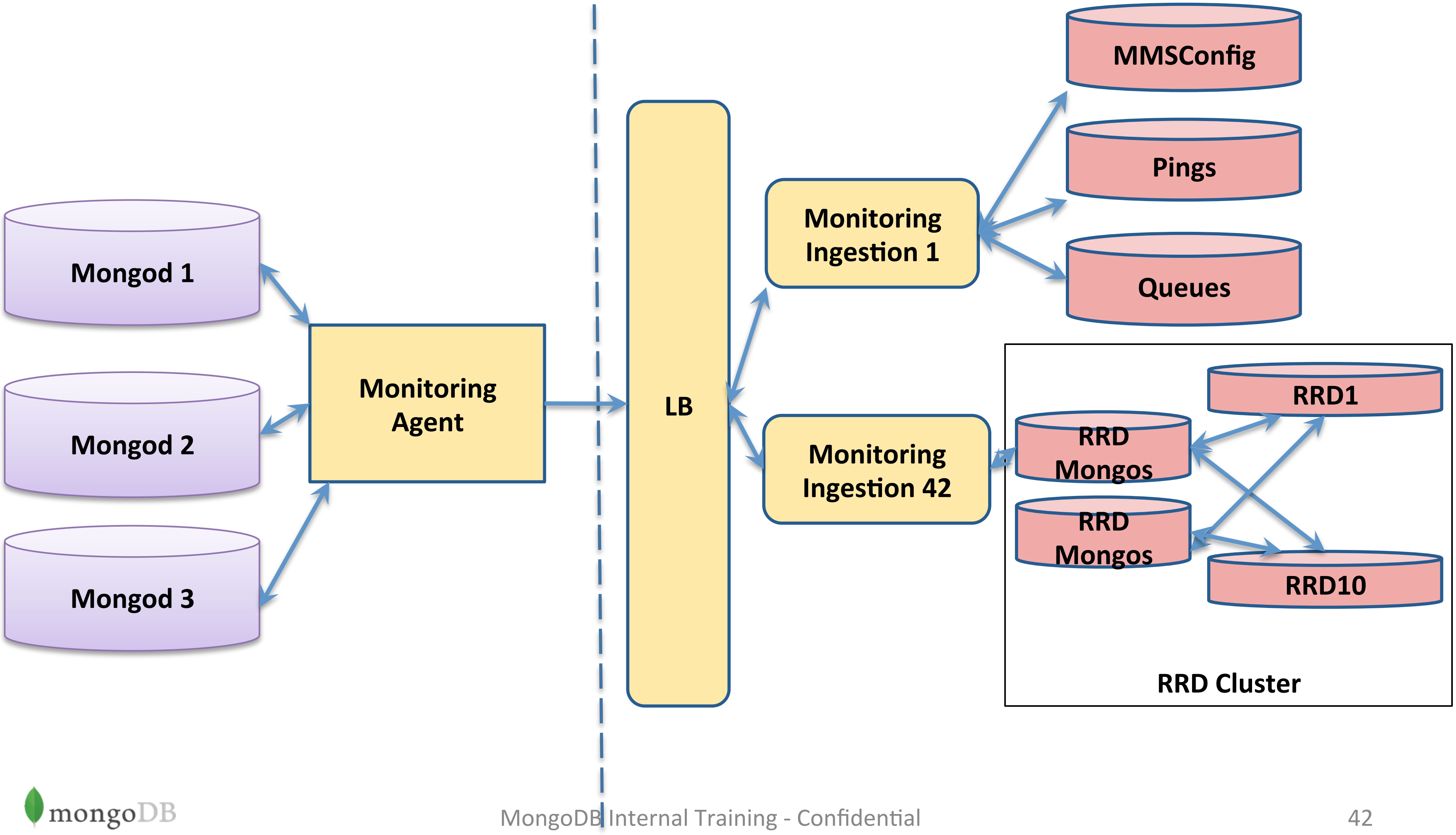
Common Issues - Daemon

- Missing appropriate MongoDB version to apply logs
- Daemon needs a lot of space, however only the Blockstore got enough
- Daemon can’t keep up with schedule
  - Applying oplogs takes too long
  - Storing Snapshot takes too long
- Blockstore grows faster than is Groomed
- Standard Mongod scaling issues

Common Issues - Restores

- SCP (tar.gz) – Host is not reachable
- SCP (individual files) – Not enough disk space
- Pull restore – connection terminated before completion
- No seedSecondary script for some restores

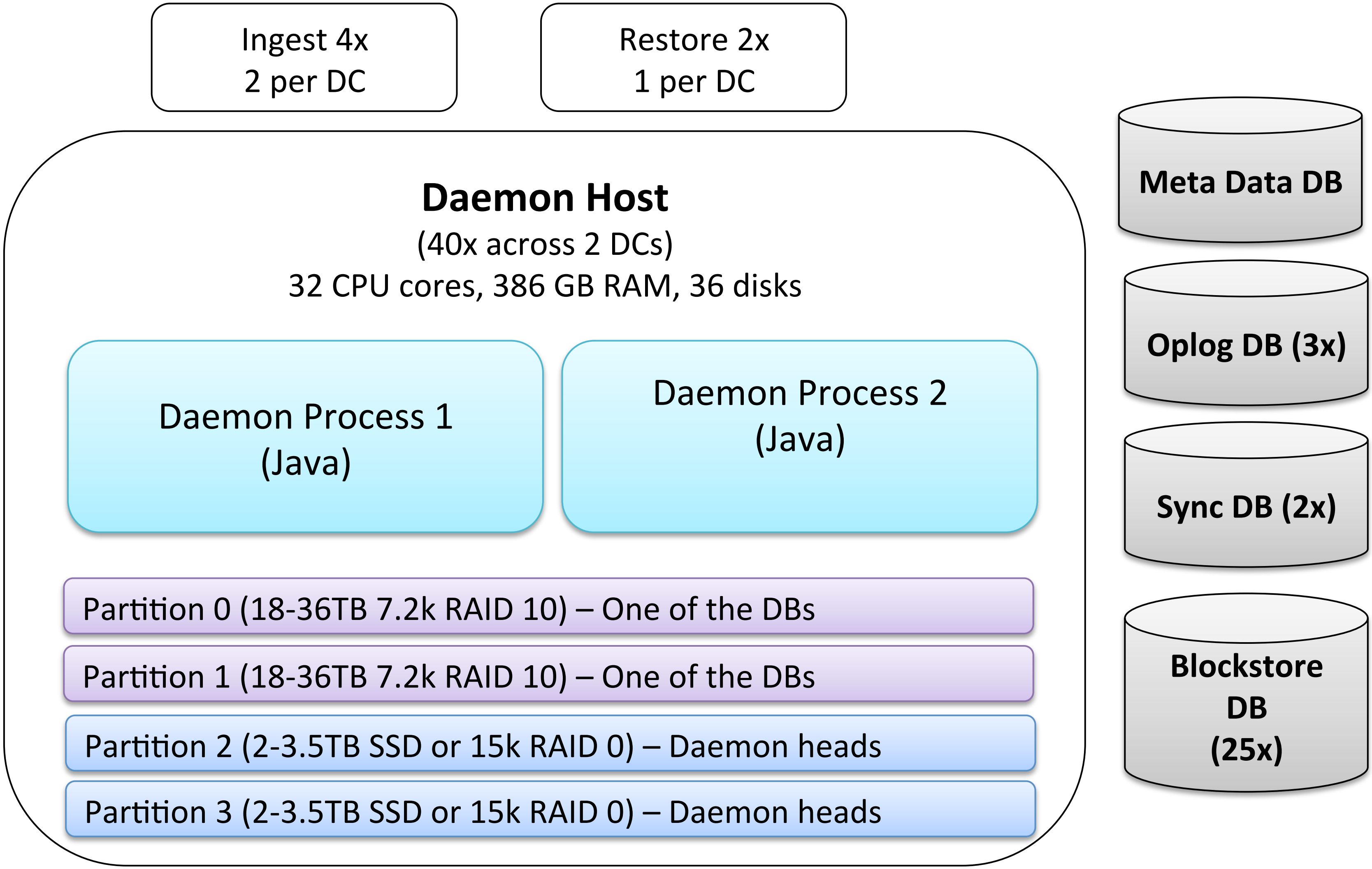
CloudManager – Monitoring Architecture



Throttling and Sampling

- Throttling based on customer type
  - MMS Classic: every 3rd ping is processed
  - MMS New (M5): every other ping is processed
- Additional automatic throttling when the system is under heavy load
- RRD data is sampled 30 min for daily resolution, every 10 min for hourly resolution
- Some stats collected less frequently (e.g. DB stats, logs, profiling)

CloudManager Deployment



Ingest 4x
2 per DC

Restore 2x
1 per DC

Daemon Host
(40x across 2 DCs)
32 CPU cores, 386 GB RAM, 36 disks

Daemon Process 1 (Java)

Daemon Process 2 (Java)

Partition 0 (18-36TB 7.2k RAID 10) – One of the DBs

Partition 1 (18-36TB 7.2k RAID 10) – One of the DBs

Partition 2 (2-3.5TB SSD or 15k RAID 0) – Daemon heads

Partition 3 (2-3.5TB SSD or 15k RAID 0) – Daemon heads

Meta Data DB

Oplog DB (3x)

Sync DB (2x)

Blockstore DB (25x)

Where is Ops Manager going?

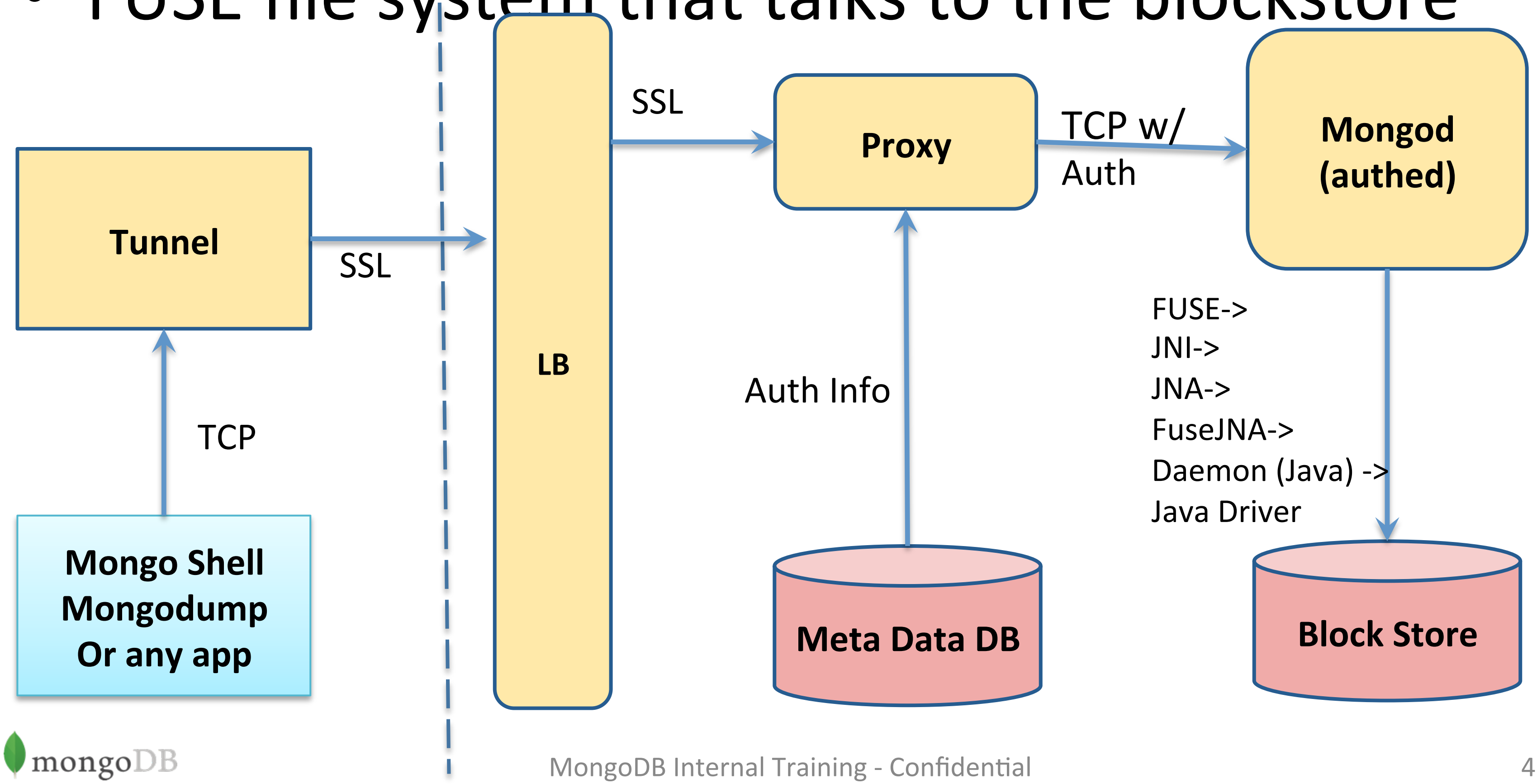
- CloudManager, on going releases every 3 weeks
  - Queryable backups
    - Based on FUSE
  - Backup snapshots to EC2 filesystem
- OpsManager 2.0, sometime this fall
  - Integration with existing backup tools (in 2.0?)
    - Dump to file system instead of using blockstore DB

Queryable snapshots

- Lazily load a snapshot with FUSE (Filesystem in UserSpacE[?]) to a directory.
- Launch MongoD with --dbpath against FUSE directory.
- Securely expose MongoD to customer for querying (complicated)

Queryable Restores

- Currently in Beta supporting Replica Sets only
- FUSE file system that talks to the blockstore



Summary

- Mental model
  - We developed MMS/CloudManager first
  - The agents always contact the server
  - The **active** monitoring agent must be able to reach all hosts/MongoDB processes
- CloudManager is:
  - one of the most extreme deployment of MongoDB
  - Always first Beta customer!!!
- Customer must have subscription to use OpsManager
  - Way to sell something else than support
- Need to make OpsManager easier to install and use

References

- MMS Backup Sizing Calculator
  - https://docs.google.com/a/10gen.com/spreadsheet/ccc?key=0AodCHiGt3A8ndDAxcVY0aHJzRHk5UVhxNGRDX0xtMEE&usp=sharing#gid=0
- Steve's training
  - https://drive.google.com/a/10gen.com/file/d/0B4f0mtJd4pkOdm82ZWVtSE5HU0U/view?usp=sharing_eid
- https://wiki.mongodb.com/display/MMS/MMS+System+Overview