MongoDB Cloud Manager (formerly MMS) and MongoDB Atlas

Contents

- Architecture & Components
  - Automation
  - Monitoring
  - Backup
- Common issues
- More details on Cloud Manager
- Where are we going?

What, When and Why

- What:
  - "MongoDB Monitoring Service", "MongoDB Management Service", "Cloud/Ops Manager", "Atlas"
  - Tool to manage MongoDB deployment
  - Cloud and Ops versions
- When (Monitoring, Backup, Automation):
  - Cloud: 9/2011, 4/2013, 10/2014 (attach 2/2015)
  - On-Prem: 7/2013, 4/2014, 3/2015
  - Atlas: 6/2016
- Why:
  - Ops are ready to pay for a better experience

CloudManager vs OpsManager

CloudManager

- free or low cost
- buy few additional hosts
- backups at $2.50/GB/month
- easy to install
- maintained by experts (us)
- data is outside org

OpsManager

- must have a MongoDB Advanced Enterprise Subscription
- buy many additional hosts
- backups at $X
- hard to install
- maintenance by local IT
- data is inside org

Classic versus newer CloudManager group

Free Tier

- monitoring –limited to 5 min granularity for last 24 hours
- backup, pay per space used and per oplog
- no automation
- no API

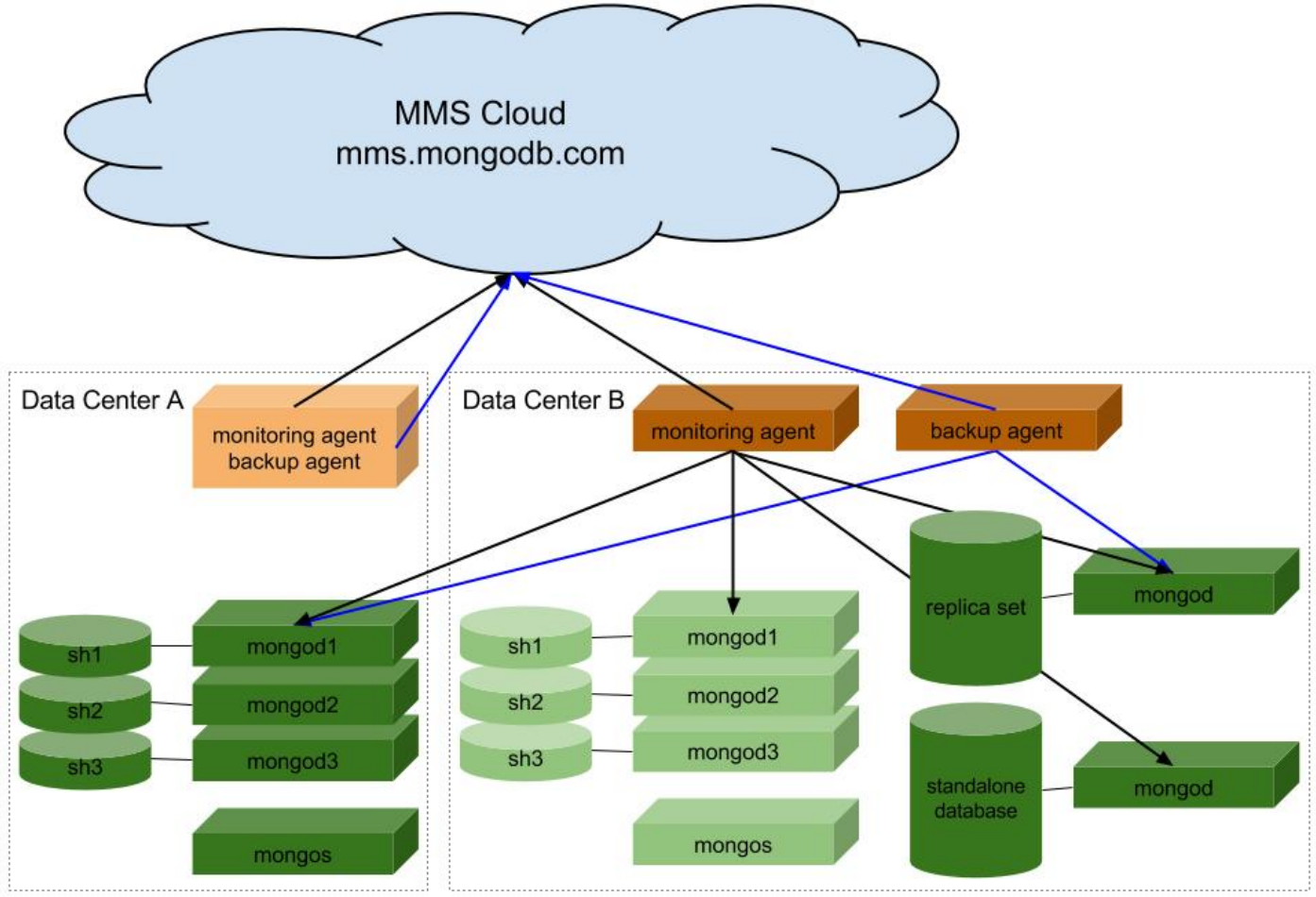
Premium

- Monitoring – free 30 days, $39/month/node after
- $2.50/GB per month for database size
- automation
- API
- manage new cluster or import existing
- Index analysis

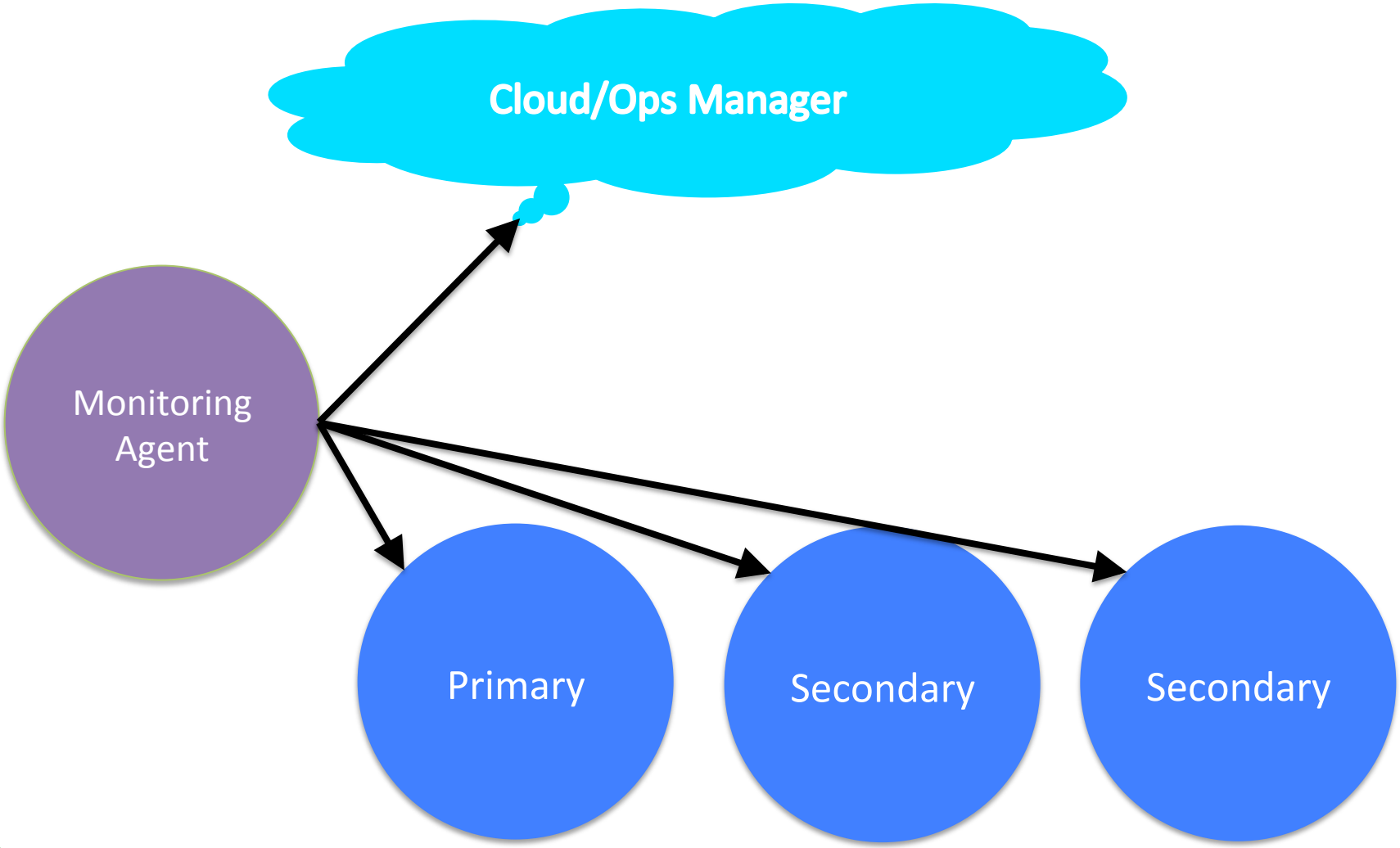
Using Cloud Manager

- Create
  - One MMS group for a set of clusters and users
- Install:
  - One automation agent per host
  - One or more monitoring agent per MMS group
  - One or more backup agent per MMS group
  - Additional monitoring/backup agents are usually in ‘stand-by’
- The agents always initiate the communication to the MMS Server

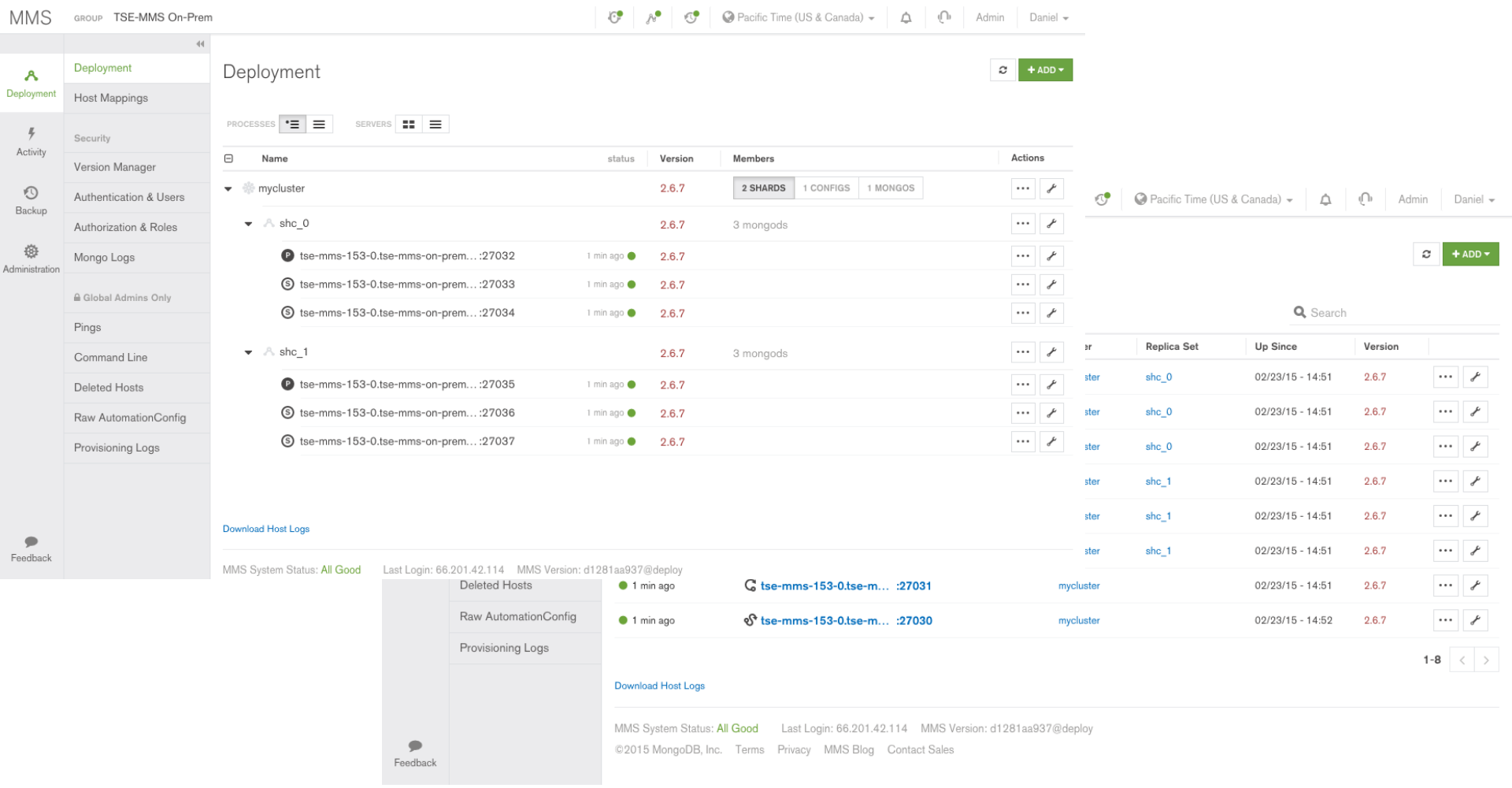
Monitoring and Backup Agents



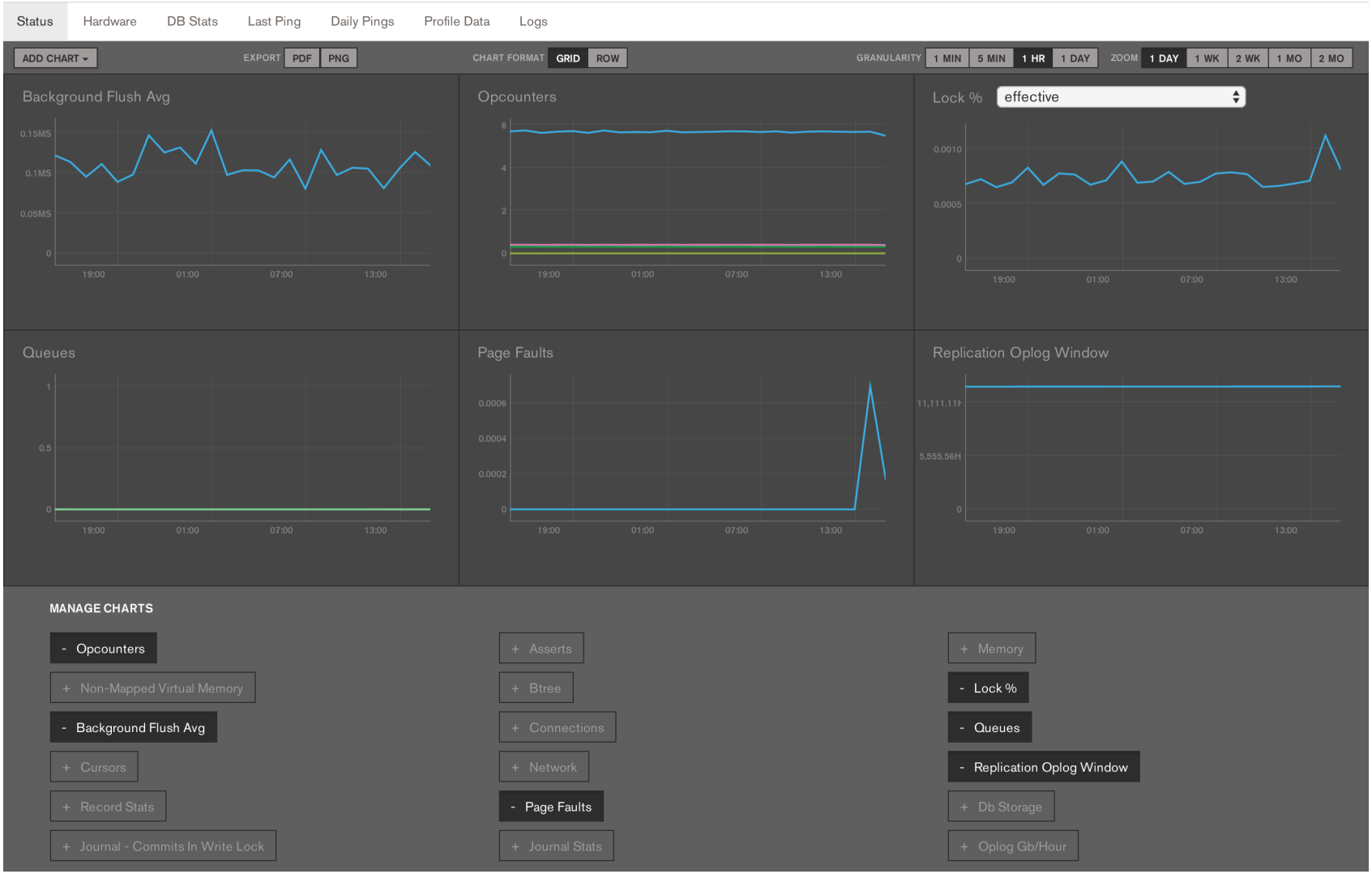
Cloud/Ops Manager Agents always initiate communications



Monitoring UI

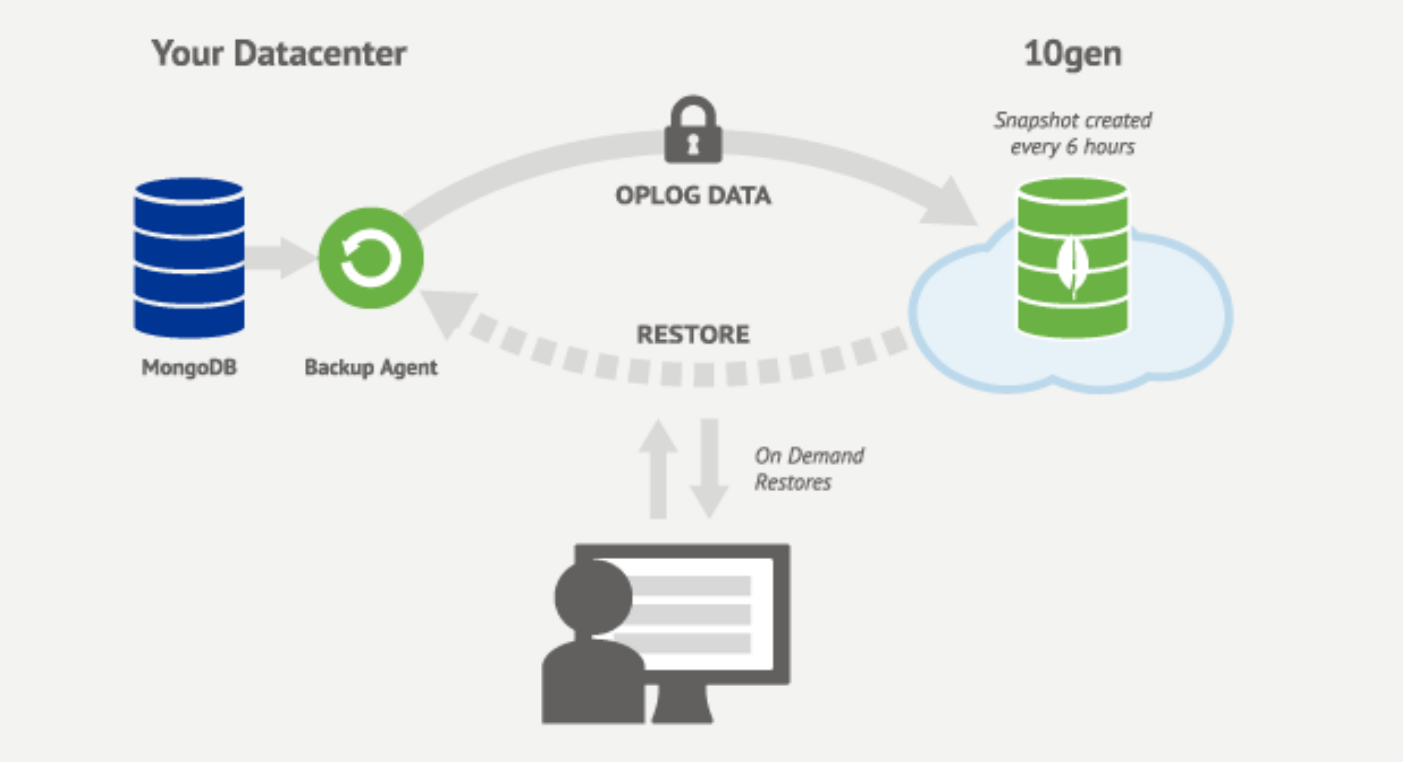


Monitoring UI



Backup with CloudManager

- Easy to install
- Slow restores for large datasets



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)
- 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

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

Backing up a 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

MMS API

- Only approved way to programmatically access the MMS 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 – 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 monitoring agents
  - Only one could connect to all mongod processes

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

Cloud Manager

- Let's look at our MMS deployment in the cloud.
- This is the largest MMS deployment known of man kind
- … and likely one of the largest MongoDB deployment too!

CloudManager by the numbers

- Monitoring (Feb 2015)
  - 56,000 Groups (7,000 Active)
    - 86,000/6500 (Aug 2015)
  - 54,000 Hosts Monitored
  - 63,000 Users (6,500 Active)
    - 93,000/8,000 (Aug 2015)
  - 9.5M Pings/Day (3M processed)
  - 14M Chart Requests / Day

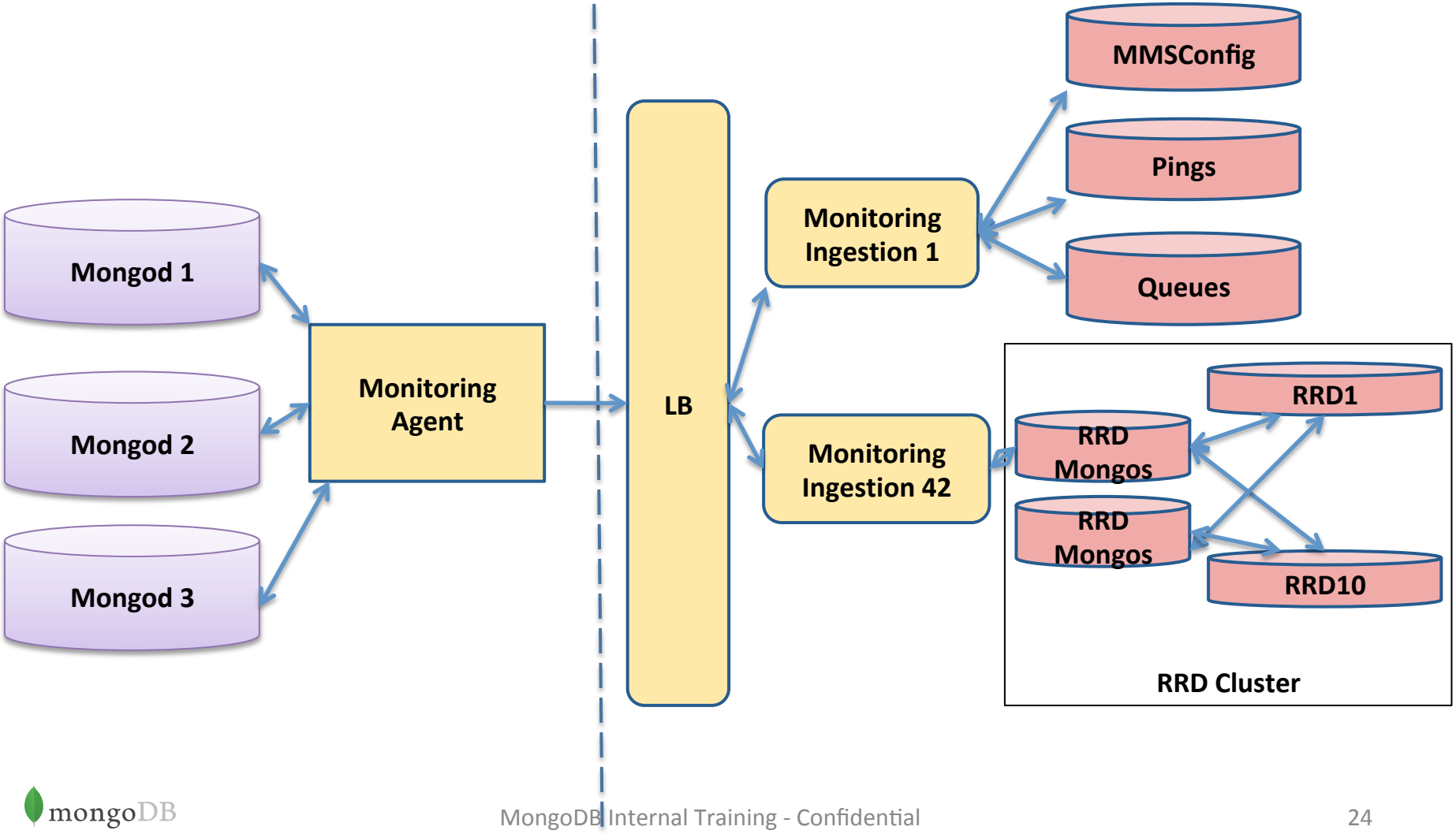
CloudManager Monitoring

- 2 Data Centers
- Databases
  - 3 replica sets: mmsconfig, pings, queues
  - RRD Cluster: 10 shard cluster
  - All are running on 3.0.0-rcX (as of Feb 2015)
- App Servers
  - 6 physical machines between 2 datacenters
  - 8 app servers per machine

CloudManager Backup

- Databases
  - Metadata DBs (isdb)
  - 3 oplog stores
  - 2 syncs stores
  - 25 blockstores (up from 6 in 4 months!)
- App Servers
  - 4 ingests
  - 2 restore “ingests”
  - ~50 daemon machines with 2 daemons each
    - 8 queryable restore daemons
    - 2 dedicated restore daemons

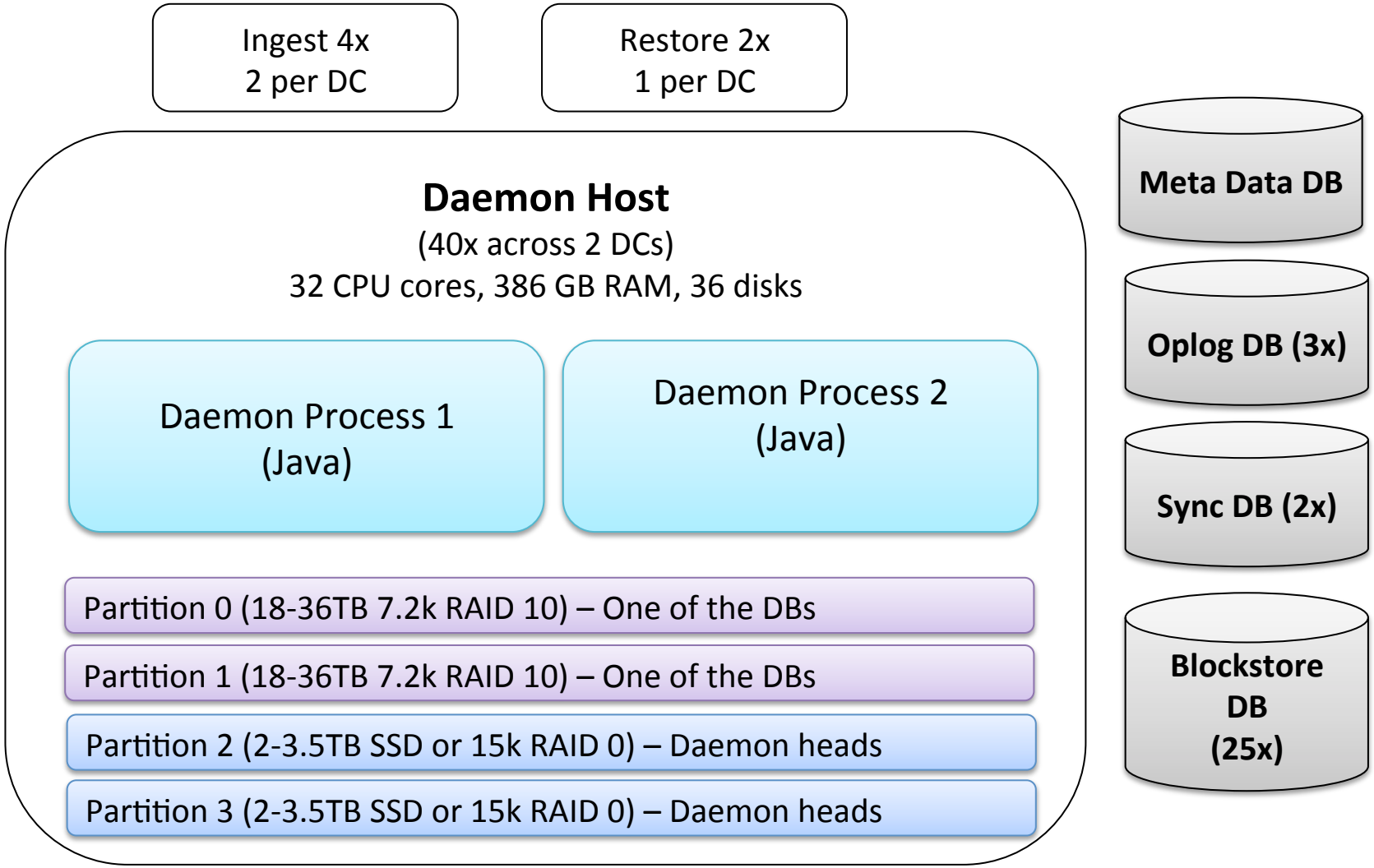
CloudManager – Monitoring Architecture



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



Where are CloudManager and OpsManager going?

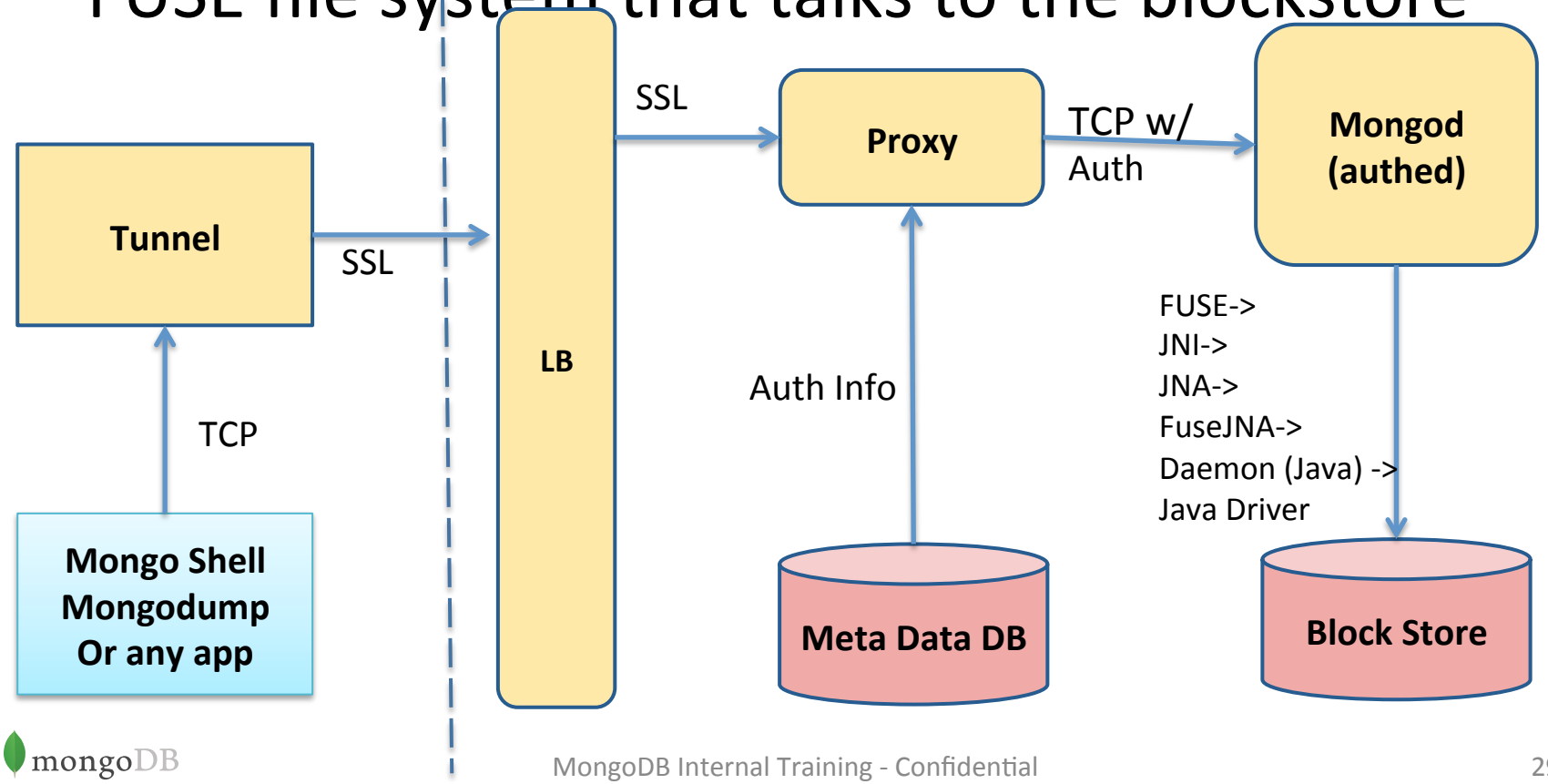
- CloudManager, on going releases every 3 weeks
  - Queryable backups
    - Based on FUSE
  - Backup snapshots to EC2 filesystem
- OpsManager 2.0, sometime this fall
  - Integration with existing backup tools (in 2.0?)
    - Dump to file system instead of using blockstore DB

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



Atlas

- Some restictions
  - db.runCommand({ setParameter: 1, … })

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

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