


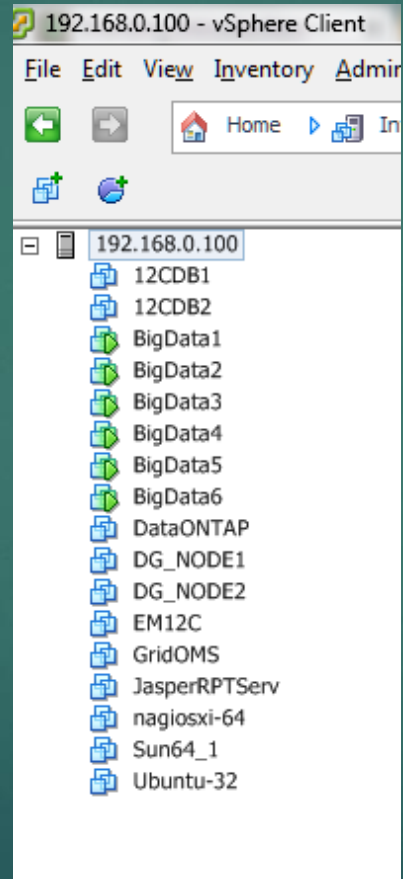


Installing and Setting up mongoDB replica set

PREPARED BY SUDHEER KONDLA
SOLUTIONS ARCHITECT

- 
- ▶ If you have multiple virtual machines, bring up VMs that you want to install MongoDB and set up a replica set.
 - ▶ The presentation is based on VMs created using VMWare's ESXi/Vsphere and RHEL/Oracle Linux.
 - ▶ This presentation is based on setting up replica set on local machine (Ubuntu 14.14) and
 - ▶ with 6 VMs in the MongoDB cluster (RedHat Linux).
 - ▶ Each VM consists of 2 vCPUs and 4 GB of RAM
 - ▶ Each VM is created with 80 GB of disk space. No special mounts/ file systems are used.
 - ▶ Linux version used: 6.5

Bring up VIRTUAL MACHINES



ESXi Hardware configuration

Getting Started Summary Virtual Machines Resource Allocation Performance Configuration Local Users & Groups Events Perm

ESXi Shell for the host has been enabled
SSH for the host has been enabled

General

Manufacturer:	empty
Model:	empty
CPU Cores:	8 CPUs x 2.266 GHz
Processor Type:	Intel(R) Xeon(R) CPU E5520 @ 2.27GHz
License:	Evaluation Mode -
Processor Sockets:	2
Cores per Socket:	4
Logical Processors:	16
Hyperthreading:	Active
Number of NICs:	4
State:	Connected
Virtual Machines and Templates:	17
vMotion Enabled:	N/A
VMware EVC Mode:	Disabled
vSphere HA State	Ⓜ N/A
Host Configured for FT:	N/A
Active Tasks:	
Host Profile:	N/A
Image Profile:	ESXi-5.1.0-20130402001-st...
Profile Compliance:	ⓘ N/A
DirectPath I/O:	Supported ⓘ

Resources

CPU usage: **3546 MHz** Capacity 8 x 2.266 GHz

Memory usage: **12063.00 MB** Capacity 24567.17 MB

Storage	Drive Type	Capacity
VMSTORE1	Non-SSD	1.36 TB 54%
VMSTORE2	Non-SSD	1.36 TB 41%
VMSTORE3	Non-SSD	1.36 TB 34%

Network Type

VM Network	Standard port group
------------	---------------------

Fault Tolerance

Fault Tolerance Version: 4.0.0-4.0.0-4.0.0 [Refresh Virtual Machine Counts](#)

Total Primary VMs:	0
Powered On Primary VMs:	0
Total Secondary VMs:	0
Powered On Secondary VMs:	0

Host Management

[Manage this host through VMware vCenter.](#)

Commands

VM Configuration

The screenshot displays the 'BigData1 - Virtual Machine Properties' window. The 'Hardware' tab is selected, showing a list of hardware components and their configurations. The 'Memory' component is highlighted, showing 4096 MB. The 'Memory Configuration' section on the right shows a memory size of 4 GB, with a scale from 1 GB to 1011 GB. Recommendations for memory size are provided: 1011 GB (Maximum recommended), 24568 MB (Maximum recommended for best performance), 2 GB (Default recommended), and 512 MB (Minimum recommended).

BigData1 - Virtual Machine Properties

Hardware | Options | Resources | Virtual Machine Version: 8

☐ Show All Devices Add... Remove

Hardware	Summary
Memory	4096 MB
CPUs	2
Video card	Video card
VMCI device	Restricted
SCSI controller 0	Paravirtual
Hard disk 1	Virtual Disk
CD/DVD drive 1	[VMSTORE1] V41362-0...
Network adapter 1	VM Network
Network adapter 2	VM Network
Floppy drive 1	Client Device

Memory Configuration

Memory Size: 4 GB

1011 GB
512 GB
256 GB
128 GB
64 GB
32 GB
16 GB
8 GB
4 GB
2 GB
1 GB

Maximum recommended for this guest OS: 1011 GB.
Maximum recommended for best performance: 24568 MB.
Default recommended for this guest OS: 2 GB.
Minimum recommended for this guest OS: 512 MB.

Installing mongodb

- ▶ The following steps guide you through installing mongodb software on Linux.
- ▶ set yum repository and download packages using yum package installer.
- ▶ `[root@bigdata1 ~]# cd /etc/yum.repos.d/`
- ▶ `[root@bigdata1 yum.repos.d]# ls -l`
- ▶ `-rw-r--r-- 1 root root 122 Dec 17 20:50 mongodb.repo`
- ▶ `[root@bigdata1 yum.repos.d]# scp mongodb.repo bigdata6:/etc/yum.repos.d/`
- ▶ Before you run “yum install”, be sure to check internet is working.
- ▶ `[root@bigdata6 yum.repos.d]# yum install -y mongodb-org-2.6.1 mongodb-org-server-2.6.1 mongodb-org-shell-2.6.1 mongodb-org-mongos-2.6.1 mongodb-org-tools-2.6.1`

Create yum repository for mongodb

```
[root@bigdata1 ~]# cd /etc/yum.repos.d/
[root@bigdata1 yum.repos.d]# ls -l
total 8
-rw-r--r-- 1 root root 122 Dec 17 20:50 mongodb.repo
drwxr-xr-x 2 root root 4096 Dec 17 20:50 tmp
[root@bigdata1 yum.repos.d]# cat mongodb.repo
[mongodb]
name=MongoDB Repository
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64/
gpgcheck=0
enabled=1
[root@bigdata1 yum.repos.d]#
```

Yum Install mongodb

```
[root@bigdata6 yum.repos.d]# yum install -y mongodb-org-2.6.1 mongodb-org-server-2.6.1 mongodb-org-shell-2.6.1 mongodb-org-mongos-2.6.1 mongodb-org-tools-2.6.1
Loaded plugins: security
mongodb                                | 951 B    00:00
mongodb/primary                        | 38 kB    00:00
mongodb                                245/245
```

Setting up Install Process

Resolving Dependencies

--> Running transaction check

---> Package mongodb-org.x86_64 0:2.6.1-2 will be installed

---> Package mongodb-org-mongos.x86_64 0:2.6.1-2 will be installed

---> Package mongodb-org-server.x86_64 0:2.6.1-2 will be installed

---> Package mongodb-org-shell.x86_64 0:2.6.1-2 will be installed

---> Package mongodb-org-tools.x86_64 0:2.6.1-2 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing:				
mongodb-org	x86_64	2.6.1-2	mongodb	4.6 k
mongodb-org-mongos	x86_64	2.6.1-2	mongodb	6.8 M
mongodb-org-server	x86_64	2.6.1-2	mongodb	8.9 M
mongodb-org-shell	x86_64	2.6.1-2	mongodb	4.2 M
mongodb-org-tools	x86_64	2.6.1-2	mongodb	89 M

Transaction Summary

Install 5 Package(s)

Yum Install mongodb

```
Total download size: 109 M
Installed size: 274 M
Downloading Packages:
(1/5): mongodb-org-2.6.1-2.x86_64.rpm
(2/5): mongodb-org-mongos-2.6.1-2.x86_64.rpm
(3/5): mongodb-org-server-2.6.1-2.x86_64.rpm
(4/5): mongodb-org-shell-2.6.1-2.x86_64.rpm
(5/5): mongodb-org-tools-2.6.1-2.x86_64.rpm
-----
Total
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : mongodb-org-mongos-2.6.1-2.x86_64
  Installing : mongodb-org-tools-2.6.1-2.x86_64
  Installing : mongodb-org-shell-2.6.1-2.x86_64
  Installing : mongodb-org-server-2.6.1-2.x86_64
  Installing : mongodb-org-2.6.1-2.x86_64
  Verifying  : mongodb-org-server-2.6.1-2.x86_64
  Verifying  : mongodb-org-2.6.1-2.x86_64
  Verifying  : mongodb-org-shell-2.6.1-2.x86_64
  Verifying  : mongodb-org-tools-2.6.1-2.x86_64
  Verifying  : mongodb-org-mongos-2.6.1-2.x86_64

Installed:
  mongodb-org.x86_64 0:2.6.1-2  mongodb-org-mongos.x86_64 0:2.6.1-2  mongodb-org-server.x86_64 0:2.6.1-2  mongodb-org-sh

Complete!
[root@bigdata6 yum.repos.d]#
```

Setting up mongod

- ▶ Make sure to create `/data/configdb` and `/data/db` directories on each servers
- ▶ The above directories should be owned by `mongod` user and `mongod` group.
- ▶ Change ownership to `mongod`
- ▶ As a root user run `“chown mongod:mongod /data/configdb”` and `“chown mongod:mongod /data/db”`
- ▶ Without above directories `mongod` process will not start
- ▶ When you start `mongo` daemon process it will create `“/data/configdb/mongod.lock”` file
- ▶ You can also start `mongod` process with service option as root. For example `“service mongod start”`
- ▶ You can also configure `mongo` daemon to start at system boot.

Creating a replica set

- ▶ This procedure describes deploying a replica set in a development or test environment.
- ▶ Three member [replica sets](#) provide enough redundancy to survive most network partitions and other system failures.
- ▶ These sets also have sufficient capacity for many distributed read operations.
- ▶ Replica sets should always have an odd number of members.
- ▶ This ensures that [elections](#) will proceed smoothly.
- ▶ Before you can deploy a replica set, you must install MongoDB on each system that will be part of your [replica set](#).
- ▶ For test and development systems, you can run your mongod instances on a local system.
- ▶ The examples in this procedure create a new replica set named rs0.
- ▶ If your application connects to more than one replica set, each set should have a distinct name.
- ▶ Some drivers group replica set connections by replica set name.

Setting up local replica sets

- ▶ Make 3 directories to set up 3 replicas on local node.
- ▶ `mkdir -p /data/mongodb/rs0-0 /data/mongodb/rs0-1 /data/mongodb/rs0-2`
- ▶ Start 3 mongo daemons on local node with 3 different available ports.
 - ▶ `mongod --port 27017 --dbpath /data/mongodb/rs0-0 --replSet rs0 --smallfiles --oplogSize 128`
 - ▶ `mongod --port 27018 --dbpath /data/mongodb/rs0-1 --replSet rs0 --smallfiles --oplogSize 128`
 - ▶ `mongod --port 27019 --dbpath /data/mongodb/rs0-2 --replSet rs0 --smallfiles --oplogSize 128`
- ▶ Check the mongo daemons running with “`ps -ef | grep mongo`” command

```
root    11613    1  0 Dec18 ?        00:14:17 /home/hadoop/mongodb/bin/mongod --port 27017 --dbpath /srv/mongodb/rs0-0 --replSet rs0 --smallfiles --oplogSize 128
root    11614    1  0 Dec18 ?        00:12:45 /home/hadoop/mongodb/bin/mongod --port 27018 --dbpath /srv/mongodb/rs0-1 --replSet rs0 --smallfiles --oplogSize 128
root    11615    1  0 Dec18 ?        00:12:48 /home/hadoop/mongodb/bin/mongod --port 27019 --dbpath /srv/mongodb/rs0-2 --replSet rs0 --smallfiles --oplogSize 128
```

Initialize and add replica sets on local node.

- Connect to first mongo process and initialize mongoDB replica.

```
hadoop@hadoop:~/admin$ mongo --port 27017
```

```
MongoDB shell version: 2.6.4
```

```
connecting to: 127.0.0.1:27017/test
```

```
> rs.initiate()
```

```
{
  "info2" : "no configuration explicitly specified -- making one",
  "me" : "hadoop:27017",
  "info" : "Config now saved locally. Should come online in about a minute.",
  "ok" : 1
}
```

```
> rs.conf()
```

```
{
  "_id" : "rs0",
  "version" : 1,
  "members" : [
    {
      "_id" : 0,
      "host" : "hadoop:27017"
    }
  ]
}
```

Adding replicas

- ▶ rs0:PRIMARY> rs.add("hadoop:27018")
- ▶ { "ok" : 1 }
- ▶ rs0:PRIMARY> rs.add("hadoop:27019")
- ▶ { "ok" : 1 }

```
rs0:PRIMARY> rs.conf()
{
  "_id" : "rs0",
  "version" : 3,
  "members" : [
    {
      "_id" : 0,
      "host" : "hadoop:27017"
    },
    {
      "_id" : 1,
      "host" : "hadoop:27018"
    },
    {
      "_id" : 2,
      "host" : "hadoop:27019"
    }
  ]
}
```

Replica status

```
hadoop@hadoop:~$ mongo --host hadoop --port 27017
MongoDB shell version: 2.6.4
connecting to: hadoop:27017/test
rs0:PRIMARY> rs.status()
{
  "set" : "rs0",
  "date" : ISODate("2014-12-20T18:58:44Z"),
  "myState" : 1,
  "members" : [
    {
      "_id" : 0,
      "name" : "hadoop:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 139942,
      "optime" : Timestamp(1418962120, 1),
      "optimeDate" : ISODate("2014-12-19T04:08:40Z"),
      "electionTime" : Timestamp(1418962078, 2),
      "electionDate" : ISODate("2014-12-19T04:07:58Z"),
      "self" : true
    },
    {
      "_id" : 1,
      "name" : "hadoop:27018",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 139809,
      "optime" : Timestamp(1418962120, 1),
      "optimeDate" : ISODate("2014-12-19T04:08:40Z"),
      "lastHeartbeat" : ISODate("2014-12-20T18:58:43Z"),
      "lastHeartbeatRecv" : ISODate("2014-12-20T18:58:44Z"),
      "pingMs" : 0,
      "syncingTo" : "hadoop:27017"
    },
    {
      "_id" : 2,
      "name" : "hadoop:27019",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 139804,
      "optime" : Timestamp(1418962120, 1),
      "optimeDate" : ISODate("2014-12-19T04:08:40Z"),
      "lastHeartbeat" : ISODate("2014-12-20T18:58:44Z"),
      "lastHeartbeatRecv" : ISODate("2014-12-20T18:58:43Z"),
      "pingMs" : 0,
      "syncingTo" : "hadoop:27017"
    }
  ],
  "ok" : 1
}
```


Connect to local replicas

```
hadoop@hadoop:~/admin$ mongo --port 27017
MongoDB shell version: 2.6.4
connecting to: 127.0.0.1:27017/test
rs0:PRIMARY> exit
bye
hadoop@hadoop:~/admin$ mongo --port 27018
MongoDB shell version: 2.6.4
connecting to: 127.0.0.1:27018/test
rs0:SECONDARY> exit
bye
hadoop@hadoop:~/admin$ mongo --port 27019
MongoDB shell version: 2.6.4
connecting to: 127.0.0.1:27019/test
rs0:SECONDARY>
```


Setting 3 node mongo replica

- ▶ Install mongod software on all nodes in the cluster.
- ▶ Start mongod, mongoes and config servers on all nodes.

Node1:

```
#!/bin/bash
mongod --configsvr --dbpath /data/configdb --port 27019 --bind_ip 192.168.0.131 -v &
sleep 5
mongos --configdb bigdata1:27019,bigdata2:27019,bigdata3:27019 &
sleep 5
mongod --replset "rs0" &
sleep 5
```

Node2:

```
#!/bin/bash
mongod --configsvr --dbpath /data/configdb --port 27019 --bind_ip 192.168.0.132 -v &
sleep 5
mongos --configdb bigdata1:27019,bigdata2:27019,bigdata3:27019 &
sleep 5
mongod --replset "rs0" &
sleep 5
```

Node3:

```
#!/bin/bash
mongod --configsvr --dbpath /data/configdb --port 27019 --bind_ip 192.168.0.133 -v &
sleep 5
mongos --configdb bigdata1:27019,bigdata2:27019,bigdata3:27019 &
sleep 5
mongod --replset "rs0" &
sleep 5
```

Add replica sets on other nodes

```
[hdfs@bigdata1 ~]$ mongo --host bigdata1 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata1:27017/test
rs0:PRIMARY> rs.add("bigdata2:27017")
{ "ok" : 1 }
rs0:PRIMARY> rs.add("bigdata3:27017")
{ "down" : [ "bigdata3:27017" ], "ok" : 1 }
```

```
rs0:PRIMARY> rs.add("bigdata4:27017")
{ "ok" : 1 }
rs0:PRIMARY> rs.add("bigdata5:27017")
{ "ok" : 1 }
rs0:PRIMARY> rs.conf()
{
  "_id" : "rs0",
  "version" : 5,
  "members" : [
    {
      "_id" : 0,
      "host" : "bigdata1:27017"
    },
    {
      "_id" : 1,
      "host" : "bigdata2:27017"
    },
    {
      "_id" : 2,
      "host" : "bigdata3:27017"
    },
    {
      "_id" : 3,
      "host" : "bigdata4:27017"
    },
    {
      "_id" : 4,
      "host" : "bigdata5:27017"
    }
  ]
}
```

Primary and Secondary replica

```
{
  "_id" : 1,
  "name" : "bigdata2:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1225,
  "optime" : Timestamp(1418965767, 1),
  "optimeDate" : ISODate("2014-12-19T05:09:27Z"),
  "infoMessage" : "syncing to: bigdata4:27017",
  "self" : true
},
{
  "_id" : 2,
  "name" : "bigdata3:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 990,
  "optime" : Timestamp(1418965767, 1),
  "optimeDate" : ISODate("2014-12-19T05:09:27Z"),
  "lastHeartbeat" : ISODate("2014-12-19T05:13:08Z"),
  "lastHeartbeatRecv" : ISODate("2014-12-19T05:13:09Z"),
  "pingMs" : 0,
  "lastHeartbeatMessage" : "syncing to: bigdata4:27017",
  "syncingTo" : "bigdata4:27017"
},
{
  "_id" : 3,
  "name" : "bigdata4:27017",
  "health" : 1,
  "state" : 1,
  "stateStr" : "PRIMARY",
  "uptime" : 290,
  "optime" : Timestamp(1418965767, 1),
  "optimeDate" : ISODate("2014-12-19T05:09:27Z"),
  "lastHeartbeat" : ISODate("2014-12-19T05:13:10Z"),
  "lastHeartbeatRecv" : ISODate("2014-12-19T05:13:08Z"),
  "pingMs" : 0,
  "electionTime" : Timestamp(1418965955, 1),
  "electionDate" : ISODate("2014-12-19T05:12:35Z")
},
}
```

Connections to Primary replica sets

```
[root@bigdata4 ~]# mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata4:27017/test
rs0:PRIMARY>
```

```
[root@bigdata1 ~]# mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata4:27017/test
rs0:PRIMARY>
```

```
[hdfs@bigdata2 admin]$ mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata4:27017/test
rs0:PRIMARY>
```

```
[root@bigdata5 ~]# mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata4:27017/test
rs0:PRIMARY>
```

```
remote machine: hadoop@hadoop:~/admin$ mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.4
connecting to: bigdata4:27017/test
rs0:PRIMARY>
```

Connections secondary replica set

```
[hdfs@bigdata1 ~]$ mongo --host bigdata1 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata1:27017/test
rs0:SECONDARY> 
```

```
[hdfs@bigdata2 ~]$ mongo --host bigdata2 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata2:27017/test
rs0:SECONDARY> 
```

```
[hdfs@bigdata3 ~]$ mongo --host bigdata3 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata3:27017/test
rs0:SECONDARY> 
```

```
[hdfs@bigdata4 ~]$ mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata4:27017/test
rs0:PRIMARY> 
```

```
[hdfs@bigdata5 ~]$ mongo --host bigdata5 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata5:27017/test
rs0:SECONDARY> 
```

CRUD operations on replica sets

```
[hdfs@bigdata2 ~]$ mongo --host bigdata1 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata1:27017/test
rs0:SECONDARY>

rs0:SECONDARY> db.inventory.insert(
...   {
...     item: "ABC1",
...     details: {
...       model: "14Q3",
...       manufacturer: "XYZ Company"
...     },
...     stock: [ { size: "S", qty: 25 }, { size: "M", qty: 50 } ],
...     category: "clothing"
...   }
... )
WriteResult({ "writeError" : { "code" : undefined, "errmsg" : "not master" } })
rs0:SECONDARY>

[root@bigdata5 ~]# mongo --host bigdata4 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata4:27017/test
rs0:PRIMARY> db.inventory.insert(
...   {
...     item: "ABC1",
...     details: {
...       model: "14Q3",
...       manufacturer: "XYZ Company"
...     },
...     stock: [ { size: "S", qty: 25 }, { size: "M", qty: 50 } ],
...     category: "clothing"
...   }
... )
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.inventory.find()
{ "_id" : ObjectId("549445df95923d76efcc5246"), "item" : "ABC1", "details" : { "model" : "14Q3",
"manufacturer" : "XYZ Company" }, "stock" : [ { "size" : "S", "qty" : 25 }, { "size" : "M", "qty" : 50 } ],
"category" : "clothing" }
rs0:PRIMARY> db.inventory.count()
1
```

Reading from secondary replica

```
[hdfs@bigdata5 ~]$ mongo --host bigdata5 --port 27017
MongoDB shell version: 2.6.1
connecting to: bigdata5:27017/test
rs0:SECONDARY> db.inventory.find()
error: { "$err" : "not master and slaveOk=false", "code" : 13435 }
rs0:SECONDARY> db.inventory.find().readPref( { mode: 'secondary'})
{ "_id" : ObjectId("549445df95923d76efcc5246"), "item" : "ABC1", "details" : { "model" : "14Q3", "manufacturer" : "XYZ Company" }, "stock" : [ {
"size" : "S", "qty" : 25 }, { "size" : "M", "qty" : 50 } ], "category" : "clothing" }
{ "_id" : ObjectId("54944ff863498a0e7f597612"), "item" : "ABC2", "details" : { "model" : "14Q3", "manufacturer" : "M1 Corporation" }, "stock" : [
{ "size" : "M", "qty" : 50 } ], "category" : "clothing" }
{ "_id" : ObjectId("54944ff863498a0e7f597613"), "item" : "MNO2", "details" : { "model" : "14Q3", "manufacturer" : "ABC Company" }, "stock" : [ {
"size" : "S", "qty" : 5 }, { "size" : "M", "qty" : 5 }, { "size" : "L", "qty" : 1 } ], "category" : "clothing" }
{ "_id" : ObjectId("54944ff863498a0e7f597614"), "item" : "IJK2", "details" : { "model" : "14Q2", "manufacturer" : "M5 Corporation" }, "stock" : [
{ "size" : "S", "qty" : 5 }, { "size" : "L", "qty" : 1 } ], "category" : "houseware" }
{ "_id" : ObjectId("54959ca12456c5f521c776cb"), "item" : "BE10", "details" : { "model" : "14Q2", "manufacturer" : "XYZ Company" }, "stock" : [ {
"size" : "L", "qty" : 5 } ], "category" : "clothing" }
{ "_id" : ObjectId("54959ca12456c5f521c776cc"), "item" : "ZYT1", "details" : { "model" : "14Q1", "manufacturer" : "ABC Company" }, "stock" : [ {
"size" : "S", "qty" : 5 }, { "size" : "M", "qty" : 5 } ], "category" : "houseware" }
rs0:SECONDARY> █
```


Setting up and Working with MMS

- ▶ Mongo Management Service (MMS) automates , monitors, admisters your mongo instances.
- ▶ Compatible with mongoDB version 2.4 or later.
- ▶ To use Automation, you must have an agent running on every host where a managed MongoDB instance runs.
- ▶ Agents do *not* transmit any data from a MongoDB deployment. The agents only communicate cluster configuration information
- ▶ You can configure all MongoDB deployment types, including standalones, replica sets, and sharded clusters. You can also configure replica set-member types, such as arbiter, hidden, and delayed secondary, and can configure replica set options, such as priority level.
- ▶ You can deploy MongoDB to Amazon Web Services (AWS).
- ▶ MMS Monitoring will scale its request cycle to limit more expensive statistics gathering. The DB Stats information updates every 10 minutes, and the agent will throttle the frequency to reduce the impact on the database
- ▶ MMS Backup creates backups of MongoDB replica sets and sharded clusters.

Configure Monitor and Backup Agents

Agents

AWS Settings

Billing/Subscriptions


Payment History


Agent Downloads


Monitoring current: 2.8.0


Backup current: 2.9.1

Automation current: 1.4.3

 RHEL/CentOS (5.X, 6.X, 7.X), SUSE, and Amazon Linux - RPM

 Ubuntu (12.04 and above) - DEB

 Other Linux - TAR

 Mac OSX (10.8 and above) - TAR

Install or Update the Automation Agent on RHEL (6+) / CentOS (6+) / Amazon Linux

INSTALL UPDATE

1. Download the 32-bit or 64-bit rpm.

```
curl -OL https://mms.mongodb.com/download/agent/automation/mongodb-mms-automation-agent-manager-1.4.3.799-1.x86_64.rpm
```

2. Install the package

```
sudo rpm -U mongodb-mms-automation-agent-manager-1.4.3.799-1.x86_64.rpm
```

3. Edit the config file `sudo vi /etc/mongodb-mms/automation-agent.config` and enter your Group ID and API key, as shown below

```
mmsGroupId=549083d9e4b0adda320f3433
mmsApiKey=f455362957c74a64baff4d859c927c6
```

4. Prepare a directory in which to store your MongoDB data. This directory must be owned by the `mongod` user. Any directory is fine, but the default suggested by MMS is `/data`. This directory can be created with a command similar to below.

```
sudo mkdir /data
sudo chown mongod:mongod /data
```

5. Start the agent

```
sudo service mongodb-mms-automation-agent start
```

```
[root@bigdata1 ~]# curl -OL https://mms.mongodb.com/download/agent/automation/mongodb-mms-automation-agent-manager-1.4.3.799-1.x86_64.rpm
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 4018k 100 4018k    0     0 1708k      0  0:00:02  0:00:02 --:--:-- 2329k
[root@bigdata1 ~]# rpm -U mongodb-mms-automation-agent-manager-1.4.3.799-1.x86_64.rpm
package mongodb-mms-automation-agent-manager-1.4.3.799-1.x86_64 is already installed
[root@bigdata1 ~]# chown mongod:mongod /data
[root@bigdata1 ~]# service mongodb-mms-automation-agent start
mongodb-mms-automation-agent is already running
[root@bigdata1 ~]#
```

MMS Deployment Page

EDITING

Deployment

view mode | edit mode

+ ADD

TOPOLOGY VIEW

SERVER VIEW

bigdata1.localdomain

Automation Agent

1.4.3.799

Monitoring Agent

2.8.0

Backup Agent

2.9.1

Drag Processes Between Servers

bigdata4.localdomain

Automation Agent

1.4.3.799

Monitoring Agent

2.8.0

Backup Agent

2.9.1.235-1

Drag Processes Between Servers

bigdata5.localdomain

Automation Agent

1.4.3.799

Monitoring Agent

2.8.0

Backup Agent

2.9.1

Drag Processes Between Servers

bigdata6.localdomain

Automation Agent

1.4.3.799

Monitoring Agent

2.8.0

Backup Agent

2.9.1

Drag Processes Between Servers

hadoop

Automation Agent

1.4.3.799

Monitoring Agent

myMongo_0 27000

2.6.6

Drag Processes Between Servers

Monitor MongoDB server with MMS

