MongoDB University

Getting Started with MongoDB Ops Manager

MongoDB Ops Manager Workshop

Release 3.4

MongoDB, Inc.

May 30, 2017

Contents

1 MongoDB Ops Manager

1.1 Lab: Ops Manager Installation

Premise

Ops Manager is an On-Prem operational solution for the management of MongoDB clusters.

Enables features like:

- Automation
- Backup and Recovery
- Monitoring

Over the course of this lab we will be installing Ops Manager with high availability and scalability in mind.

Ops Manager HA

Ops Manager requires a number of servers for high availability (HA).

- Monitoring and backup/recovery are essential for production operations.
- Therefore, it's important to assure high availability for Ops Manager.
- For this we need to follow a specific deployment topology.

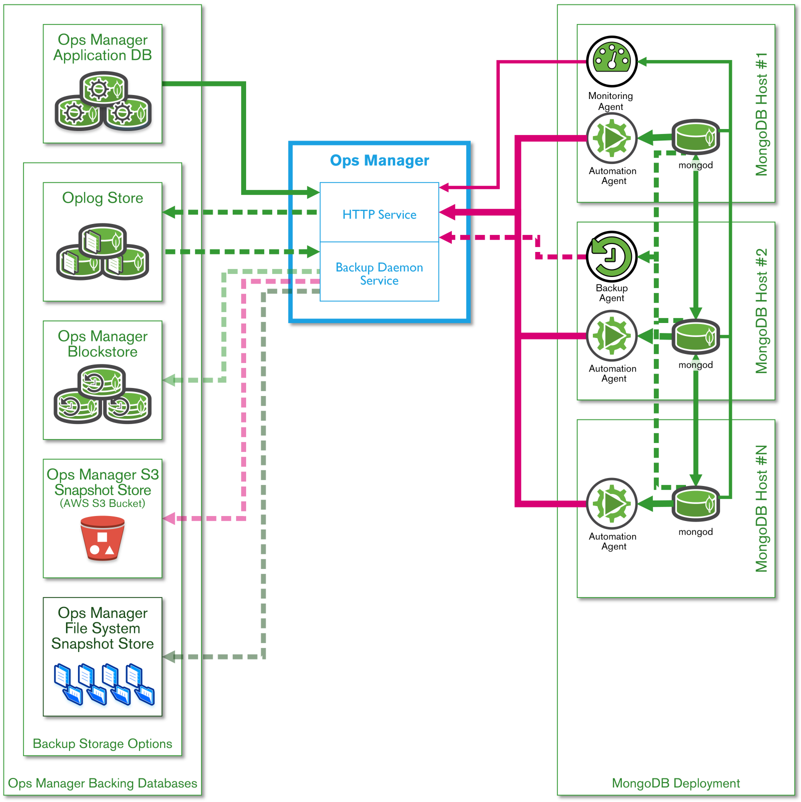
Ops Manager Scalability

Why do we need our operations tool to be scalable?

- The main reason is backup and recovery requirements
- The amount of data individual applications generate will grow
- The number of applications your Ops Manager deployment supports will grow
- Plan to accommodate both forms of growth

Ops Manager Architecture Review

Let's review the Ops Manager architecture[1] :



Exercise: Architect the Ops Manager Deployment

It's time to set up the our Ops Manager Deployment. As a team, make a plan for the following:

- Two replica sets of 3 nodes
  - Application Database replica set as **APPDB**
  - Backup Database replica set as **BACKUPDB**
- A redundant service of the Ops Manager Application
  - The hosts that will be supporting the OM App: `opsmgr1`, `opsmgr2` and `opsmgr3`
  - Load Balancer in front of those 3 instances
  - The load balancer is already set up. The name is in the info file

[1] https://docs.opsmanager.mongodb.com/current/core/system-overview/

Exercise: Configure Ops Manager Application Database

Ops Manager needs to store data:

- Configuration of nodes, groups, users
- Metrics for monitoring
- Backup metadata and job queries

Also consider relevant security settings[2] for this database.

From the available machines go ahead and set up a replica set to support the *Application Database*.

Name this replica set **APPDB**

You can install MongoDB by running:

```
yum install -y /share/downloads/mongodb_packages/mongodb-enterprise-3.4.2-1.el7.x86_
↪64.rpm
```

Exercise: Configure Ops Manager Backup Database

Ops Manager needs to store backup blocks/snaphots, either

- in database
- file system

From the available machines go ahead and set up a replica set to support the *Backup Database*.

Name this replica set **BACKUPDB**

Exercise: Install, Configure and Launch the Ops Manager Service

Habemus Replica Sets! Now it's time to launch the **Ops Manager** service. For this you will need to:

- Install Ops Manager
  - The files can be found in /share/downloads/opsmgr_packages
  - Follow the instructions to install from rpm[3]
- Edit Ops Manager configuration `conf-mms.properties`:
  - Point the config to the replica set: **APPDB**
- Launch the Ops Manager service
- Hint: there is a common keyfile shared by all 3 instances
- You can install Ops Manager by running:

  ```
  yum install -y /share/downloads/opsmgr_packages/mongodb-mms-3.4.3.402-1.x86_64.rpm
  ```

[2] https://docs.mongodb.com/manual/administration/security-checklist/

[3] https://docs.opsmanager.mongodb.com/current/tutorial/install-on-prem-with-rpm-packages/#install-the-onprem-package-on-each-server-being-used-for-onprem

Exercise: Install Ops Manager Automation Agents

At this point **Ops Manager** should be up and running. Now it's time to install our Automation Agents[4]:

- In the remaining VMs (node1, node2, etc) install the automation agent
- Make sure that all nodes are discoverable on the server's dashboard
- Validate that all agents are reporting pings correctly

1.2 Lab: Enable the Ops Manager Public API

Learning Objectives

Upon completing this lab, students will be able to:

- Understand the requirements for enabling Ops Manager Public API

Exercise: Enable Public API Access

Ops Manager, for most users, is primarily controlled via it's web UI, but it has an API that supports most of the operations that users perform.

Enable your deployment of Ops Manager to allow API calls.

- Generate an API Key called "generic"

To verify that you've done this properly you can make the following request:

```
curl -u "$EMAIL:$APIKEY" --digest \
  -i "$OPSMGRURL/api/public/v1.0/groups"
```

1.3 Lab: Ops Manager User Administration

Learning Objectives

Upon completing this lab, students will be able to:

- Administer Ops Manager groups
- Identify the differences between Ops Manager user roles
- Create and define Ops Manager users

[4] https://docs.opsmanager.mongodb.com/current/tutorial/nav/install-automation-agent/

Exercise: Create Group

Connect to your Ops Manager instance and create the following `group`:

- **CIRCUS_MAXIMUS**

Exercise: Create Users

Using the Ops Manager API[5] , create the following users:

- **aediles@localhost.com** :
  - password: "123ABCabc!"
  - role: Owner[6]
- **patrician@localhost.com** :
  - password: "DAxN3ZpM6U!"
  - role: Monitoring Admin[7]
- **consus@localhost.com** :
  - password: "&o7chac0v3r3d"
  - role: Backup Admin[8]

Exercise: Create Global Users

In various different situations, we will need users with global roles.

Please create, either through the API or web UI, the following users:

- **automater@localhost.com** :
  - password: "84hjdpx%ea3m"
  - role: Global Automation Admin[9]
- **masterchef@localhost.com** :
  - password: "c6ny3n4x*8"
  - role: Global User Admin[10]

After creating these users, connect with the most appropriate user to change the password of the **CIRCUS_MAXIMUS** *Owner* user.

The new password should be *"$uperC00l"*

This last operation should be accomplished using the HTTP Rest API interface.

[5] https://docs.opsmanager.mongodb.com/current/api/

[6] https://docs.opsmanager.mongodb.com/current/reference/user-roles/#owner

[7] https://docs.opsmanager.mongodb.com/current/reference/user-roles/#monitoring-admin

[8] https://docs.opsmanager.mongodb.com/current/reference/user-roles/#backup-admin

[9] https://docs.opsmanager.mongodb.com/current/reference/user-roles/#global-automation-admin

[10] https://docs.opsmanager.mongodb.com/current/reference/user-roles/#global-user-admin

1.4 Lab: Secure Replica Set

Premise

- Setting up a MongoDB Replica set is quite easy and fast.
- Setting up a Secured MongoDB replica set requires a few extra steps.
- In this lab we will be exploring how to setup a secured Replica Set through Ops Manager.

X.509 Authentication Mechanism

We will be using X.509 certificates[11] for authentication and TLS/SSL network encryption.

Ops Manager Group SSL and Auth

To build secured MongoDB deployments you first need to enable Auth and SSL[12] on your group.

All VMs have a set of certificates that you will be using to configure your secured deployment.

In folder `/share/downloads/certs` (linked to `/etc/ssl/mongodb`) you will find:

- `ca.pem`: SSL CA certificate
- `automation.pem`: Automation agent certificate
- `backup.pem`: Backup agent certificate
- `monitor.pem`: Monitoring agent certificate
- `nodeX.pem`: Replica set member certificates (X)
- `dbadmin.pem`: MongoDB DB Admin certificate

Exercise: `VERYSAFE` Group

Let's start by creating a group called `VERYSAFE` that has SSL enabled.

- Using the existing certificates, configure the agents accordingly.
- You need to specify certificates for
  - Certificate Authority
  - Monitoring Agent
  - Backup Agent
  - Automation Agent
- **The existing certificates do not have a decryption password!**

[11] https://docs.mongodb.com/manual/core/security-x.509/

[12] https://docs.opsmanager.mongodb.com/current/tutorial/enable-ssl-for-a-deployment/

Exercise: Secure Replica Set Deployment

Once the automation agent has been reconfigured and servers are detected on your deployment, it's then time to deploy our secure replica set.

Create a replica set named **SECURE** with the following configuration:

- 3 Nodes:
  - **node1**, **node2** and **node3**
  - Port *27000*
- **sslMode**: requiredSSL
- **sslPEMKeyFile**: */etc/ssl/mongodb/nodeX.pem*

Exercise: X509 Users

Time to create users that will authenticate using an X.509 certificate.

- Go ahead and create a dbAdminAnyDatabase[13] user that authenticates using the `dbadmin.pem` certificate.
- To create users that authenticate using X509 certificates you should check the Certificate Subject as user[14] documentation.
- After the user has been created, connect to the *Primary* node of the replica set and create database "`allgood`".

[13] https://docs.mongodb.com/manual/reference/built-in-roles/#dbAdminAnyDatabase

[14] https://docs.mongodb.com/manual/tutorial/configure-x509-client-authentication/#add-x-509-certificate-subject-as-a-user

mongoDB

Find out more
mongodb.com | mongodb.org
university.mongodb.com

Having trouble?
File a JIRA ticket:
jira.mongodb.org

Follow us on twitter
@MongoDBInc
@MongoDB