

Ubuntu - Process for Upgrading MongoDB 3.4 Standalone to MongoDB 4.2 Standalone

Following the release notes for MongoDB you must upgrade to each version of MongoDB along the way.

NOTE: All Linux Commands in this document are run at root level: you need access to root via `su` or `sudo -i`

This document will follow the MongoDB upgrade path from 3.4 3.6 4.0 4.2, with the document structured as follows:

1. Prerequisites
2. Upgrade Preparation
3. Upgrade from MongoDB 3.4 to MongoDB 3.6
4. Upgrade from MongoDB 3.6 to MongoDB 4.0
5. Upgrade from MongoDB 4.0 to MongoDB 4.2
6. Services Restart
7. Clean-up

Table of Contents

- [Prerequisites for Upgrading MongoDB 3.4 to MongoDB 4.2](#)
 - [Supported Operating Systems](#)
 - [MongoDB 3.4 Installed and Running](#)
 - [MongoDB Storage Engine](#)
 - [Upgrade NMIS and Modules for Mongo 4.2](#)
- [Preparation](#)
 - [System Snapshot/Backup](#)
 - [Upgrade Ubuntu Packages](#)
 - [Shutdown All OMK Daemons](#)
- [Upgrade from MongoDB 3.4 to MongoDB 3.6](#)
 - [Prepare the Database for Upgrade](#)
 - [Ubuntu - Install MongoDB 3.6 Repo and Upgrade](#)
- [Upgrade from MongoDB 3.6 to MongoDB 4.0](#)
 - [Prepare the Database for Upgrade](#)
 - [Ubuntu - Install MongoDB 4.0 Repo and Upgrade](#)
- [Upgrade from MongoDB 4.0 to MongoDB 4.2](#)
 - [Prepare the Database for Upgrade](#)
 - [Ubuntu - Install MongoDB 4.2 Repo and Upgrade](#)
- [Restart All OMK Daemons](#)

Prerequisites for Upgrading MongoDB 3.4 to MongoDB 4.2

As MongoDB 3.4 is out of support customers are wanting to upgrade their NMIS installation to newer in support releases. This document outlines the process to bring MongoDB up to version 4.2. There are a series of prerequisites that determine the type of update required. This document outlines the process for MongoDB Standalone.

There are 4 prerequisites to confirm:

- Operating System
- MongoDB Version
- MongoDB Storage Engine
- NMIS and Modules for Mongo 4.2

Supported Operating Systems

MongoDB 3.4 through 4.2 are officially supported on the following Ubuntu platforms:

- 18.04 LTS ("Bionic")
- 16.04 LTS ("Xenial")

We have tested the upgrade process on:

- Ubuntu 20.04 LTS ("focal")

To determine the version of Linux running on your system use the following command in a terminal session:

```
lsb_release -rc
```

An example output of this command

```
Release:      20.04
Codename:     focal
```

MongoDB 3.4 Installed and Running

Verify by running (with your mongodb <username> and <password>)

This command does not do anything however, the output will show us the running shell and server that is running.

```
mongo -u <username> -p <password> --authenticationDatabase=admin --eval 'quit()' | grep MongoDB
```

Result will look like below, we are looking for the two version numbers to be 3.4.x

If the versions do not match there may be an issue with your running installation and you should not proceed until this is resolved.

```
MongoDB shell version v3.4.24
MongoDB server version: 3.4.24
```

MongoDB Storage Engine

Verify storage engine by running

```
mongo -u <username> -p <password> --authenticationDatabase=admin --quiet --eval 'db.serverStatus().
storageEngine'
```

The "name" line should have "wiredTiger" in order to proceed.

```
{
  "name" : "wiredTiger",
  "supportsCommittedReads" : true,
  "readOnly" : false,
  "persistent" : true
}
```

If this mongo cli command does not work you can try the following to determine the storage engine in use

```
grep -i "active storage" /var/log/mongodb/mongod.log
```

Example correct log output - notice the log entry declares the storage engine to be using "wiredTiger"

```
2022-06-28T07:53:33.568+0000 I - [initandlisten] Detected data files in /var/lib/mongodb created by the
'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
```

If the system is not using wiredTiger the data migration process should follow a different process. Refer to the following link.

<https://www.mongodb.com/docs/v4.2/tutorial/change-standalone-wiredtiger/>

Upgrade NMIS and Modules for Mongo 4.2

NMIS and Modules must be updated to versions that support MongoDB v4.2 before upgrading MongoDB to v4.2. These releases are backwards compatible with MongoDB v3.4.

Minimum releases for MongoDB 4.2 are:

- NMIS 9.4.0
- opAddress 2.2.0
- opCharts 4.4.0
- opConfig 4.4.0
- opEvents 4.2.0

- opHA 3.5.0

You can determine the versions you are running by browsing to:

<https://<yourserveraddress>/omk/>

You should follow the standard upgrade process for each of these packages, which is handled by [the Opmantek Installer](#).

Preparation

The preparation phase includes 3 sections

1. System Snapshot/Backup
2. Upgrade to latest stable release packages
3. Shutdown all OMK Daemons

This is to be done in this specific sequence.

System Snapshot/Backup

For System Snapshots and Backups you will need to refer to the documentation for your environment.

If the information on your system, or stability of the system is critical to your business it is important to retain a working backup. It is important that you have confirmed that you backups can be restored.

Options:

- Full system backups, including Operating System and Data - Contact your IT department, Vendor Support, or Integrator
- Virtual Machine Snapshots - Refer to the solution guides for the environment
- Database Dump - MongoDB documentation: <https://www.mongodb.com/docs/database-tools/mongodump/>

Upgrade Ubuntu Packages

After completing the snapshot/backup follow Ubuntu's instructions for upgrading and checking the sanity of the apt package system.

Here is a link to an Ubuntu apt-get howto for further reference: <https://help.ubuntu.com/community/AptGet/Howto>

An example of the update/upgrade cycle is as follows:

```
apt-get update && apt-get upgrade -y
```

This command will execute all outstanding upgrades. Check for any errors and address as required.

Once this step is completed you may **optionally** reboot the system. This can clear out any memory leaks and instabilities in the system that occur due to long uptimes.

Shutdown All OMK Daemons

Prior to commencing into the MongoDB upgrade cycles it is required to shutdown all OMK processes.

The following commands are used to complete this task

```
/usr/local/omk/bin/checkomkdaemons.sh stop  
systemctl stop nmis9d
```

These commands offer minimal feedback, confirm that the nmis9d process has halted as follows:

Example Command and Output

```
user@servername:~/installers# systemctl status nmis9d
nmis9d.service - Opmantek NMIS9 Daemon
   Loaded: loaded (/etc/systemd/system/nmis9d.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Fri 2022-07-01 00:31:59 UTC; 1min 31s ago
     Process: 8852 ExecStart=/usr/local/nmis9/bin/nmisd (code=exited, status=0/SUCCESS)
    Main PID: 8856 (code=exited, status=0/SUCCESS)
       Tasks: 0 (limit: 9528)
      Memory: 26.4M
     CGroup: /system.slice/nmis9d.service

Jul 01 00:30:14 ferny systemd[1]: Starting Opmantek NMIS9 Daemon...
Jul 01 00:30:15 ferny systemd[1]: Started Opmantek NMIS9 Daemon.
Jul 01 00:31:58 ferny systemd[1]: Stopping Opmantek NMIS9 Daemon...
Jul 01 00:31:59 ferny systemd[1]: nmis9d.service: Succeeded.
Jul 01 00:31:59 ferny systemd[1]: Stopped Opmantek NMIS9 Daemon.
```

Upgrade from MongoDB 3.4 to MongoDB 3.6

Prepare the Database for Upgrade

Check the database compatibility mode is set to "3.4"

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'
```

The result should be as follows ensuring that the value 3.4 is set

```
{ "featureCompatibilityVersion" : "3.4", "ok" : 1 }
```

If MongoDB is on version 3.4 and the featureCompatibilityVersion is not "3.4" then set this with the following command

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
setFeatureCompatibilityVersion: "3.4" } )';
```

As before, the check the featureCompatibilityVersion again confirming the correct value

Example Command and Output

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'

{ "featureCompatibilityVersion" : "3.4", "ok" : 1 }
```

Ubuntu - Install MongoDB 3.6 Repo and Upgrade

Details available here:

<https://www.mongodb.com/docs/v3.6/tutorial/install-mongodb-on-ubuntu/>

Commands summarised here:

```
wget -qO - https://www.mongodb.org/static/pgp/server-3.6.asc | sudo apt-key add -
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/3.6 multiverse" | sudo tee /etc
/apt/sources.list.d/mongodb-org-3.6.list
sudo apt-get update
sudo apt-get upgrade -y
systemctl stop mongod
systemctl start mongod
systemctl daemon-reload
systemctl is-active mongod
```

Complete the upgrade with the following commands and retest. If you skip the stop/start sequence above the following command will fail.

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
setFeatureCompatibilityVersion: "3.6" } )'
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'
systemctl stop mongod
systemctl start mongod
```

Upgrade from MongoDB 3.6 to MongoDB 4.0

Prepare the Database for Upgrade

Check the database compatibility mode is set to "3.6"

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'
```

The result should be as follows ensuring that the value 3.6 is set

```
{ "featureCompatibilityVersion" : "3.6", "ok" : 1 }
```

If MongoDB is on version 3.6 and the featureCompatibilityVersion is not "3.6" then set this with the following command

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
setFeatureCompatibilityVersion: "3.6" } )';
```

As before, the check the featureCompatibilityVersion again confirming the correct value

Example Command and Output

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'

{ "featureCompatibilityVersion" : "3.6", "ok" : 1 }
```

Ubuntu - Install MongoDB 4.0 Repo and Upgrade

Details available here:

<https://www.mongodb.com/docs/v4.0/tutorial/install-mongodb-on-ubuntu/>

Commands summarised here:

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.0.asc | sudo apt-key add -
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse" | sudo tee /etc
/apt/sources.list.d/mongodb-org-4.0.list
sudo apt-get update
sudo apt-get upgrade -y
systemctl stop mongod
systemctl start mongod
systemctl daemon-reload
systemctl is-active mongod
```

Complete the upgrade with the following commands and retest. If you skip the stop/start sequence above the following command will fail.

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
setFeatureCompatibilityVersion: "4.0" } )'
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'
systemctl stop mongod
systemctl start mongod
```

As before, check the featureCompatibilityVersion output from above is "4.0"

Example Command and Output

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'

{ "featureCompatibilityVersion" : "4.0", "ok" : 1 }
```

Upgrade from MongoDB 4.0 to MongoDB 4.2

Prepare the Database for Upgrade

Check the database compatibility mode is set to "4.0"

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'
```

The result should be as follows ensuring that the value 4.0 is set

```
{ "featureCompatibilityVersion" : "4.0", "ok" : 1 }
```

If MongoDB is on version 4.0 and the featureCompatibilityVersion is not "4.0" then set this with the following command

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
setFeatureCompatibilityVersion: "4.0" } )';
```

As before, the check the featureCompatibilityVersion again confirming the correct value

Example Command and Output

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'

{ "featureCompatibilityVersion" : "4.0", "ok" : 1 }
```

Ubuntu - Install MongoDB 4.2 Repo and Upgrade

Details available here:

<https://www.mongodb.com/docs/v4.2/tutorial/install-mongodb-on-ubuntu/>

Commands summarised here:

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc
/apt/sources.list.d/mongodb-org-4.2.list
sudo apt-get update
sudo apt-get upgrade -y
systemctl stop mongod
systemctl start mongod
systemctl daemon-reload
systemctl is-active mongod
```

Complete the upgrade with the following commands and retest. If you skip the stop/start sequence above the following command will fail.

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
setFeatureCompatibilityVersion: "4.2" } )'
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'
systemctl stop mongod
systemctl start mongod
```

As before, check the featureCompatibilityVersion output from above is "4.2"

Example Command and Output

```
mongo -u <YOUR USERNAME> -p <YOUR PASSWORD> --authenticationDatabase=admin --quiet --eval 'db.adminCommand( {
getParameter: 1, featureCompatibilityVersion: 1 } )'

{ "featureCompatibilityVersion" : "4.2", "ok" : 1 }
```

This completes the in-situ upgrade of MongoDB 3.4 to 4.2 Standalone

Restart All OMK Daemons

Now the MongoDB upgrade is complete, you can start all the NMIS and OMK processes.

The following commands are used to complete this task


```
/usr/local/omk/bin/checkomkdaemons.sh start
systemctl start nmis9d
```

You can now browse to your instance of NMIS:

<https://<yourserver>/omk>

You will get a screen similar to the following which will give you a status on all installed services:

Opmantek Applications

Database Status: 

All configured databases are ok.

Welcome to the Opmantek Application Stack

All Opmantek applications are flexible and adaptable. It's easy: first add devices (and access credentials) so data can be collected, then use the Opmantek applications to access the information you need to gain visibility of your IT environment.

[Add Nodes](#)**First Step**[Application Administration](#)**Second Step**[Getting Started with Opmantek Applications Guide](#)**Documentation**

Need Support?

Extensive product documentation, guides and community questions are available on the Opmantek Community.

[Opmantek Community Wiki](#)[Community Questions](#)[Open Support Ticket](#)**NMIS**

9.4.0

NMIS provides visibility of an IT environment, providing valuable information about infrastructure performance and faults.

Your version is up to date.

**opConfig**

4.4.0

Provides configuration backup, archiving and change detection.

Your version is up to date.

**opReports**

4.3.1

Enhanced NMIS reports with engineering and business related reports.

Your version is up to date.

**opAddress**

2.1.1

IP address management as it should be, simplified and automated.

Your version is up to date.

**opEvents**

4.2.0

Event Management processing syslog, SNMP trap, NMIS events and others.

Your version is up to date.

**Open-Audit**

4.3.3

Open-Audit intelligently scans your network for everything.

Your version is up to date.

**opCharts**

4.4.0

Delivers interactive charts, custom dashboards and network diagrams.

Your version is up to date.

**opHA**

3.4.3

Manage large and geographically dispersed environments.

Your version is up to date.