

# MongoDB Installation

- [What is MongoDB?](#)
- [Which MongoDB version should be used?](#)
- [Installing MongoDB](#)
- [Upgrading MongoDB](#)
- [Configuring MongoDB Authentication/Authorization with the setup\\_mongodb.pl Script](#)
- [Resetting Databases with the setup\\_mongodb.pl Script](#)
- [Deprecated - Historic Installation Instructions](#)

## What is MongoDB?

MongoDB is a [free and open-source big-data document database](#) that stores data in flexible, JSON-like documents. Opmantek chose MongoDB as our back-end database because it is fast and reliable, all with relatively low overhead.

## Which MongoDB version should be used?

As of 2018 we recommend [version 3.4 of MongoDB](#), and the [WiredTiger storage engine](#).

It is possible to continue to use MongoDB 2.6 for certain products, but the 3.x versions bring in a number of important performance improvements and [some products](#) strictly require these capabilities. MongoDB 3.6 is not supported by our applications at this time. Version 2.4 is no longer supported.

Please see our [Product Compatibility](#) page for details regarding what product releases work (best) with which MongoDB versions.

## Installing MongoDB

The installer in our recent product releases offers to install version 3.4 of MongoDB for you if **none** is present on your server, and no manual configuration should be required.

## Upgrading MongoDB

If you do already have MongoDB 3.x on your system, then the installer (for opmantek application releases after Jan 2018) will offer to upgrade MongoDB to version 3.4 for you.

If you have MongoDB 2.x, then you will have to upgrade manually; we **HIGHLY RECOMMEND** that you follow the official documentation for MongoDB upgrades

(for [upgrades to 3.0](#), for [upgrades to 3.2](#) and finally for [upgrades to 3.4](#)) - the two most critical aspects are

- that you **cannot** skip major versions for upgrades (i.e. from 2.4 you **must** upgrade to 2.6 first, then to 3.0, then to 3.2 and finally to 3.4) ,
- and that you **must** upgrade the MongoDB authorization scheme before 3.0 will work, as documented [on this page](#).

Besides these caveats we've not encountered any major problems with upgrades in our lab tests (tested on CentOS/RedHat 6, Debian 7, Ubuntu 14).

Step by step procedures for upgrading MongoDB from 3.4 to 4.2 are available [here](#)

## Configuring MongoDB Authentication/Authorization with the setup\_mongodb.pl Script

All Opmantek applications that require MongoDB ship with a configuration tool for MongoDB, named `bin/setup_mongodb.pl`. It's recommended that you use this tool for setting up MongoDB authentication, as it can deal with separate per-product MongoDB instances and simplifies the procedure substantially. The relevant product installers already offer to run the tool during installation and upgrades, hence you rarely need to run it manually.

To use the tool manually, simply run it as root and follow the prompts (if any):

```
$ sudo /usr/local/omk/bin/setup_mongodb.pl
setup_mongodb.pl version 1.0.3

Reading configuration file /usr/local/omk/conf/opCommon.nmis...
Checking authentication status for db_server localhost...
MongoDB on localhost:27017 is running in non-authenticated mode.
INFO: adding user opUserRW to database admin
INFO: adding user opUserRW to database nmis
INFO: adding user opUserRW to database optrend
...
```

If required the tool will prompt you for database credentials and advise on remote-vs-local operation, as well as authenticated-vs-nonauthenticated modes. The MongoDB setup helper can be run repeatedly with no adverse effects.

See how authentication is enabled in this mongod.conf on the NMIS9 VM that we ship.  
See 'security:' >> 'authorization: enabled':

#### /etc/mongod.conf

```
net:
  bindIp: 127.0.0.1
  port: 27017
processManagement:
  fork: true
  pidFilePath: /var/run/mongodb/mongod.pid
security:
  authorization: enabled
storage:
  dbPath: /data/mongo
  journal:
    enabled: true
  wiredTiger:
    engineConfig:
      cacheSizeGB: 1.7
systemLog:
  destination: file
  logAppend: true
  logRotate: reopen
  path: /var/log/mongodb/mongod.log
```

See paragraph **Setting cacheSizeGB appropriately when MongoDB table type is wiredtiger (the default)** in this wiki page:

[Reducing a Server's Memory Footprint when OMK applications are installed](#)

for the calculation we use to calculate 'cacheSizeGB: 1.7' in the above mongod.conf file.

## Resetting Databases with the setup\_mongodb.pl Script

Should you want to reset one or more product databases to their initial empty state, you will need version 2.54 (or newer) of setup\_mongodb.pl. With that version you may pass the extra arguments "drop=<dbname1>,<dbname2> confirm=YES", and setup\_mongodb will completely empty the named databases. Please note that this is an **irrevocable operation** and may lead to substantial data loss if you pick the wrong database to drop!

You can find the database names in the standard configuration file `conf/opCommon.nmis`, under the keys `<product>_db_name` and `db_name`.

Please note that the database named by the key `db_name` (default name: "nmis") is shared and used by multiple products and therefore dropping this database would affect multiple products!

## Deprecated - Historic Installation Instructions

**As of September 2015 the installation instructions below are no longer applicable; the installer will take care of installation and initial configuration of MongoDB for you.**

### Debian:

Debian 7.0 (aka Wheezy) ships with a much older version, which *should* work but very likely won't provide adequate performance. However, the mongodb packages in wheezy-backports are fine, as are the versions in Testing/Jessie and Unstable. First make sure that [you have Backports or Testing/Unstable enabled](#), then use `sudo` or `su` to run the following command as `root`:

```
apt-get install mongodb-clients mongodb-server
```

As an alternative you can also download Debian-ready packages from [the MongoDB site](#), which do however use a different package name.

### Ubuntu:

Recent versions of Ubuntu come with suitable MongoDB packages. Use `sudo` or `su` to become the `root` user, then run the following command:

```
apt-get install mongodb-clients mongodb-server
```

Installing the mongodb-server package will result in a working, automatically started MongoDB with no authentication.

## Other Systems:

Download MongoDB from the Website at <https://www.mongodb.org/downloads>.

```
curl http://fastdl.mongodb.org/linux/mongodb-linux-x86_64-2.2.3.tgz -o mongodb-linux-x86_64-2.2.3.tgz
tar xvf mongodb-linux-x86_64-2.2.3.tgz
mv mongodb-linux-x86_64-2.2.3 /usr/local/
cd /usr/local/
ln -s mongodb-linux-x86_64-2.2.3 mongodb
ln -s mongodb/bin/mongo bin/mongo
```

## Opmantek Setup for MongoDB

For Debian-derived distributions the default configuration is fine, but you might still want to adjust the database storage area. The configuration file is `/etc/mongodb.conf`, and the init script is in `/etc/init.d/mongodb`.

For other systems, check that the location in the provided init script is where you want the database to store its files (the default is `mongodbpath=/var/mongodb` for the original version):

NOTE - this file will only exist AFTER you have run an installer which requires mongodb.

```
vi /usr/local/omk/install/mongod.init.d

### you will see
### two suggestions provided

mongodbpath=/var/mongodb
## mongodbpath=/data/mongodb
```

After you have the location you prefer, the next set of commands will put a start-up script in the correct location, register the script and then start mongo (before doing this make sure to consider how much data you plan to store. Each application has different needs and how you plan to use the application will have an enormous impact on the space required by the database). The last command here starts MongoDB, the first time it runs it can take some time to do its pre-allocation of database and journal files. This will depend on the performance of your storage.

```
# as root
cp /usr/local/omk/install/mongod.init.d /etc/init.d/mongod
chkconfig mongod on
service mongod start
```

If this is a Debian-derived system and you made config changes, then your mongod will already be running and you need to restart it like this:

```
#use su or sudo to become root
service mongodb restart
# or /etc/init.d/mongodb restart
# or invoke-rc.d mongodb restart
```

## Manual MongoDB Authentication Configuration

**Please note that these instructions are applicable only for MongoDB 2.4 and as such are of historic interest only.**

Should using the `setup_mongodb.pl` helper not be an option for you, then you can still prime the database(s) manually for Opmantek use.

Load the mongo CLI, create the user for the admin DB, authenticate, then do the same for the NMIS db (change the username and password to your liking):

```
mongo
```

```
use admin;
db.addUser("opUserRW","op42flow42");
db.auth("opUserRW","op42flow42");

use nmis;
db.addUser("opUserRW","op42flow42");
db.auth("opUserRW","op42flow42");
```

NB: Now make sure that the user/pass matches the config in opCommon.nmis.

```
'database' => {
  'db_server' => 'localhost',
  'db_port' => '27017',
  'db_name' => 'nmis',
  'db_username' => 'opUserRW',
  'db_password' => 'op42flow42'
},
```

If you don't require authentication simply skip the steps above and set the username in the configuration file to "

```
# to disable authentication
'db_username' => '',
```