Introduction to MongoDB

MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.

## Document Database

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.



The advantages of using documents are:

* Documents (i.e. objects) correspond to native data types in many programming languages.
* Embedded documents and arrays reduce need for expensive joins.
* Dynamic schema supports fluent polymorphism.

## Key Features

### High Performance

MongoDB provides high performance data persistence. In particular,

* Support for embedded data models reduces I/O activity on database system.
* Indexes support faster queries and can include keys from embedded documents and arrays.

### Rich Query Language

MongoDB supports a rich query language to support [read and write operations (CRUD)](https://docs.mongodb.com/manual/crud/) as well as:

* [Data Aggregation](https://docs.mongodb.com/manual/core/aggregation-pipeline/)
* [Text Search](https://docs.mongodb.com/manual/text-search/) and [Geospatial Queries](https://docs.mongodb.com/manual/tutorial/geospatial-tutorial/).

### High Availability

MongoDB’s replication facility, called [replica set](https://docs.mongodb.com/manual/replication/), provides:

* automatic failover and
* data redundancy.

A [replica set](https://docs.mongodb.com/manual/replication/) is a group of MongoDB servers that maintain the same data set, providing redundancy and increasing data availability.

### Horizontal Scalability

MongoDB provides horizontal scalability as part of its core functionality:

* [Sharding](https://docs.mongodb.com/manual/sharding/#sharding-introduction) distributes data across a cluster of machines.
* Tag aware sharding allows for directing data to specific shards, such as to take into consideration geographic distribution of the shards.

# Install MongoDB

# Install MongoDB Community Edition on Ubuntu

## Overview

Use this tutorial to install MongoDB Community Edition on LTS Ubuntu Linux systems from .deb packages. While Ubuntu includes its own MongoDB packages, the official MongoDB Community Edition packages are generally more up-to-date.

**PLATFORM SUPPORT**

MongoDB only provides packages for 64-bit LTS (long-term support) Ubuntu releases. For example, 12.04 LTS (precise), 14.04 LTS (trusty), 16.04 LTS (xenial), and so on. These packages may work with other Ubuntu releases, however, they are not supported.

## Packages

MongoDB provides officially supported packages in their own repository. This repository contains the following packages:

|  |  |
| --- | --- |
| mongodb-org | A metapackage that will automatically install the four component packages listed below. |
| mongodb-org-server | Contains the [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "bin.mongod" \o "mongod) daemon and associated configuration and init scripts. |
| mongodb-org-mongos | Contains the [mongos](https://docs.mongodb.com/manual/reference/program/mongos/#bin.mongos) daemon. |
| mongodb-org-shell | Contains the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell. |
| mongodb-org-tools | Contains the following MongoDB tools: [mongoimport](https://docs.mongodb.com/manual/reference/program/mongoimport/#bin.mongoimport) [bsondump](https://docs.mongodb.com/manual/reference/program/bsondump/#bin.bsondump), [mongodump](https://docs.mongodb.com/manual/reference/program/mongodump/#bin.mongodump),[mongoexport](https://docs.mongodb.com/manual/reference/program/mongoexport/#bin.mongoexport), [mongofiles](https://docs.mongodb.com/manual/reference/program/mongofiles/#bin.mongofiles), [mongooplog](https://docs.mongodb.com/manual/reference/program/mongooplog/#bin.mongooplog), [mongoperf](https://docs.mongodb.com/manual/reference/program/mongoperf/#bin.mongoperf), [mongorestore](https://docs.mongodb.com/manual/reference/program/mongorestore/#bin.mongorestore),[mongostat](https://docs.mongodb.com/manual/reference/program/mongostat/#bin.mongostat), and [mongotop](https://docs.mongodb.com/manual/reference/program/mongotop/" \l "bin.mongotop" \o "mongotop). |

These packages conflict with the mongodb, mongodb-server, and mongodb-clients packages provided by Ubuntu.

The default /etc/mongod.conf configuration file supplied by the packages have bind\_ip set to127.0.0.1 by default. Modify this setting as needed for your environment before initializing a [replica set](https://docs.mongodb.com/manual/reference/glossary/#term-replica-set).

## Init Scripts

The mongodb-org package includes various [init scripts](https://docs.mongodb.com/manual/reference/glossary/#term-init-script), including the init script /etc/init.d/mongod. You can use these scripts to stop, start, and restart daemon processes.

The package configures MongoDB using the /etc/mongod.conf file in conjunction with the init scripts. See the [Configuration File](https://docs.mongodb.com/manual/reference/configuration-options/) reference for documentation of settings available in the configuration file.

As of version 3.2.9, there are no init scripts for [mongos](https://docs.mongodb.com/manual/reference/program/mongos/#bin.mongos). The [mongos](https://docs.mongodb.com/manual/reference/program/mongos/#bin.mongos) process is used only in [sharding](https://docs.mongodb.com/manual/sharding/). You can use the mongod init script to derive your own [mongos](https://docs.mongodb.com/manual/reference/program/mongos/#bin.mongos) init script for use in such environments. See the[mongos](https://docs.mongodb.com/manual/reference/program/mongos/#bin.mongos) reference for configuration details.

## Install MongoDB Community Edition

**NOTE**

To install a version of MongoDB prior to 3.2, please refer to that version’s documentation. For example, see version [3.0](https://docs.mongodb.com/v3.0/tutorial/install-mongodb-on-ubuntu/).

MongoDB only provides packages for 64-bit LTS (long-term support) Ubuntu releases. For example, 12.04 LTS (precise), 14.04 LTS (trusty), 16.04 LTS (xenial), and so on. These packages may work with other Ubuntu releases, however, they are not supported.

**1**

### Import the public key used by the package management system.

The Ubuntu package management tools (i.e. dpkg and apt) ensure package consistency and authenticity by requiring that distributors sign packages with GPG keys. Issue the following command to import the [MongoDB public GPG Key](https://www.mongodb.org/static/pgp/server-3.2.asc):

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927

**2**

### Create a list file for MongoDB.

Create the /etc/apt/sources.list.d/mongodb-org-3.2.list list file using the command appropriate for your version of Ubuntu:

Ubuntu 12.04

echo "deb http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list

Ubuntu 14.04

echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list

Ubuntu 16.04

echo "deb http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list

**3**

### Reload local package database.

Issue the following command to reload the local package database:

sudo apt-get update

**4**

### Install the MongoDB packages.

You can install either the latest stable version of MongoDB or a specific version of MongoDB.

#### Install the latest stable version of MongoDB.

Issue the following command:

sudo apt-get install -y mongodb-org

#### Install a specific release of MongoDB.

To install a specific release, you must specify each component package individually along with the version number, as in the following example:

sudo apt-get install -y mongodb-org=3.2.9 mongodb-org-server=3.2.9 mongodb-org-shell=3.2.9 mongodb-org-mongos=3.2.9 mongodb-org-tools=3.2.9

If you only install mongodb-org=3.2.9 and do not include the component packages, the latest version of each MongoDB package will be installed regardless of what version you specified.

#### Pin a specific version of MongoDB.

Although you can specify any available version of MongoDB, apt-get will upgrade the packages when a newer version becomes available. To prevent unintended upgrades, pin the package. To pin the version of MongoDB at the currently installed version, issue the following command sequence:

echo "mongodb-org hold" | sudo dpkg --set-selections

echo "mongodb-org-server hold" | sudo dpkg --set-selections

echo "mongodb-org-shell hold" | sudo dpkg --set-selections

echo "mongodb-org-mongos hold" | sudo dpkg --set-selections

echo "mongodb-org-tools hold" | sudo dpkg --set-selections

**5**

### (Ubuntu 16.04-only) Create systemd service file

**NOTE**

Follow this step ONLY if you are running Ubuntu 16.04.

Create a new file at /lib/systemd/system/mongod.service with the following contents:

**[Unit]**

Description=High-performance, schema-free document-oriented database

After=network.target

Documentation=https://docs.mongodb.org/manual

**[Service]**

User=mongodb

Group=mongodb

ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

**[Install]**

WantedBy=multi-user.target

## Run MongoDB Community Edition

The MongoDB instance stores its data files in /var/lib/mongodb and its log files in/var/log/mongodb by default, and runs using the mongodb user account. You can specify alternate log and data file directories in /etc/mongod.conf. See [systemLog.path](https://docs.mongodb.com/manual/reference/configuration-options/" \l "systemLog.path" \o "systemLog.path) and [storage.dbPath](https://docs.mongodb.com/manual/reference/configuration-options/" \l "storage.dbPath" \o "storage.dbPath) for additional information.

If you change the user that runs the MongoDB process, you **must** modify the access control rights to the/var/lib/mongodb and /var/log/mongodb directories to give this user access to these directories.

**1**

### Start MongoDB.

Issue the following command to start [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "bin.mongod" \o "mongod):

sudo service mongod start

**2**

### Verify that MongoDB has started successfully

Verify that the [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "bin.mongod" \o "mongod) process has started successfully by checking the contents of the log file at/var/log/mongodb/mongod.log for a line reading

[initandlisten] waiting for connections on port <port>

where <port> is the port configured in /etc/mongod.conf, 27017 by default.

**3**

### Stop MongoDB.

As needed, you can stop the [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "bin.mongod" \o "mongod) process by issuing the following command:

sudo service mongod stop

**4**

### Restart MongoDB.

Issue the following command to restart [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "bin.mongod" \o "mongod):

sudo service mongod restart

**5**

### Begin using MongoDB.[¶](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/#begin-using-mongodb)

To help you start using MongoDB, MongoDB provides [Getting Started Guides](https://docs.mongodb.com/manual/#getting-started) in various driver editions. See [Getting Started](https://docs.mongodb.com/manual/#getting-started) for the available editions.

Before deploying MongoDB in a production environment, consider the [Production Notes](https://docs.mongodb.com/manual/administration/production-notes/) document.

Later, to stop MongoDB, press Control+C in the terminal where the [mongod](https://docs.mongodb.com/manual/reference/program/mongod/" \l "bin.mongod" \o "mongod) instance is running.

# The Mongo Shell:

## Introduction

The [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell is an interactive JavaScript interface to MongoDB. You can use the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell to query and update data as well as perform administrative operations.

The [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell is a component of the [MongoDB distributions](http://www.mongodb.org/downloads). Once you have [installed and have started MongoDB](https://docs.mongodb.com/manual/installation/), connect the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell to your running MongoDB instance.

Most examples in the [MongoDB Manual](https://docs.mongodb.com/manual/) use the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell; however, many [drivers](https://docs.mongodb.com/manual/applications/drivers/) provide similar interfaces to MongoDB.

## Start the mongo Shell

**IMPORTANT**

Ensure that MongoDB is running before attempting to start the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell.

To start the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell and connect to your [MongoDB](https://docs.mongodb.com/manual/reference/program/mongod/) instance running on **localhost** with **default port**:

1. At a prompt in a terminal window (or a command prompt for Windows), go to your <mongodbinstallation dir>:
2. cd <mongodb installation dir>
3. Type ./bin/mongo to start [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo):
4. ./bin/mongo

If you have added the <mongodb installation dir>/bin to the PATH environment variable, you can just type mongo instead of ./bin/mongo.

### Options

When you run [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) without any arguments, the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell will attempt to connect to the MongoDB instance running on the localhost interface on port 27017. To specify a different host or port number, as well as other options, see [examples of starting up mongo](https://docs.mongodb.com/manual/reference/program/mongo/#mongo-usage-examples) and [mongo reference](https://docs.mongodb.com/manual/reference/program/mongo/) which provides details on the available options.

### .mongorc.js File

When starting, [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) checks the user’s [HOME](https://docs.mongodb.com/manual/reference/program/mongo/#envvar-HOME) directory for a JavaScript file named [.mongorc.js](https://docs.mongodb.com/manual/reference/program/mongo/#mongo-mongorc-file). If found,[mongo](https://docs.mongodb.com/manual/reference/program/mongo/" \l "bin.mongo" \o "mongo) interprets the content of .mongorc.js before displaying the prompt for the first time. If you use the shell to evaluate a JavaScript file or expression, either by using the --eval option on the command line or by specifying [a .js file to mongo](https://docs.mongodb.com/manual/reference/program/mongo/#mongo-shell-file), [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) will read the .mongorc.js file after the JavaScript has finished processing. You can prevent .mongorc.js from being loaded by using the [--norc](https://docs.mongodb.com/manual/reference/program/mongo/#cmdoption--norc) option.

## Working with the mongo Shell

To display the database you are using, type db:

db

The operation should return test, which is the default database. To switch databases, issue the use <db>helper, as in the following example:

use <database>

To list the available databases, use the helper show dbs. See also [db.getSiblingDB()](https://docs.mongodb.com/manual/reference/method/db.getSiblingDB/" \l "db.getSiblingDB" \o "db.getSiblingDB()) method to access a different database from the current database without switching your current database context (i.e.db).

You can switch to non-existing databases. When you first store data in the database, such as by creating a collection, MongoDB creates the database. For example, the following creates both the databasemyNewDatabase and the [collection](https://docs.mongodb.com/manual/reference/glossary/#term-collection) myCollection during the [insert()](https://docs.mongodb.com/manual/reference/method/db.collection.insert/#db.collection.insert) operation:

use myNewDatabase

db.myCollection.insert( { x: 1 } );

The [db.myCollection.insert()](https://docs.mongodb.com/manual/reference/method/db.collection.insert/" \l "db.collection.insert" \o "db.collection.insert()) is one of the [methods available in the mongo shell](https://docs.mongodb.com/manual/reference/method/)

* db refers to the current database.
* myCollection is the name of the collection.

If the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell does not accept the name of the collection, for instance if the name contains a space, hyphen, or starts with a number, you can use an alternate syntax to refer to the collection, as in the following:

db["3test"].find()

db.getCollection("3test").find()

### Format Printed Results

The [db.collection.find()](https://docs.mongodb.com/manual/reference/method/db.collection.find/" \l "db.collection.find" \o "db.collection.find()) method returns a [cursor](https://docs.mongodb.com/manual/reference/glossary/#term-cursor) to the results; however, in the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell, if the returned cursor is not assigned to a variable using the var keyword, then the cursor is automatically iterated up to 20 times to print up to the first 20 documents that match the query. The [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell will prompt Typeit to iterate another 20 times.

To format the printed result, you can add the .pretty() to the operation, as in the following:

db.myCollection.find().pretty()

In addition, you can use the following explicit print methods in the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell:

* print() to print without formatting
* print(tojson(<obj>)) to print with [JSON](https://docs.mongodb.com/manual/reference/glossary/#term-json) formatting and equivalent to printjson()
* printjson() to print with [JSON](https://docs.mongodb.com/manual/reference/glossary/#term-json) formatting and equivalent to print(tojson(<obj>))

For more information and examples on cursor handling in the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell, see [Iterate a Cursor in the mongo Shell](https://docs.mongodb.com/manual/tutorial/iterate-a-cursor/). See also [Cursor Help](https://docs.mongodb.com/manual/tutorial/access-mongo-shell-help/#mongo-shell-help-cursor) for list of cursor help in the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell.

### Multi-line Operations in the mongo Shell

If you end a line with an open parenthesis ('('), an open brace ('{'), or an open bracket ('['), then the subsequent lines start with ellipsis ("...") until you enter the corresponding closing parenthesis (')'), the closing brace ('}') or the closing bracket (']'). The [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell waits for the closing parenthesis, closing brace, or the closing bracket before evaluating the code, as in the following example:

> **if** ( x > 0 ) {

... count++;

... print (x);

... }

You can exit the line continuation mode if you enter two blank lines, as in the following example:

> **if** (x > 0

...

...

>

## Tab Completion and Other Keyboard Shortcuts

The [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell supports keyboard shortcuts. For example,

* Use the up/down arrow keys to scroll through command history. See [.dbshell](https://docs.mongodb.com/manual/reference/program/mongo/#mongo-dbshell-file) documentation for more information on the .dbshell file.
* Use <Tab> to autocomplete or to list the completion possibilities, as in the following example which uses <Tab> to complete the method name starting with the letter 'c':
* db.myCollection.c<Tab>

Because there are many collection methods starting with the letter 'c', the <Tab> will list the various methods that start with 'c'.

# MongoDB CRUD Operations

# CRUD operations *create*, *read*, *update*, and *delete* [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format).

## Create Operations

Create or insert operations add new [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format) to a [collection](https://docs.mongodb.com/manual/core/databases-and-collections/#collections). If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

* [db.collection.insert()](https://docs.mongodb.com/manual/reference/method/db.collection.insert/#db.collection.insert)
* [db.collection.insertOne()](https://docs.mongodb.com/manual/reference/method/db.collection.insertOne/#db.collection.insertOne) New in version 3.2
* [db.collection.insertMany()](https://docs.mongodb.com/manual/reference/method/db.collection.insertMany/#db.collection.insertMany) New in version 3.2

In MongoDB, insert operations target a single [collection](https://docs.mongodb.com/manual/reference/glossary/#term-collection). All write operations in MongoDB are [atomic](https://docs.mongodb.com/manual/core/write-operations-atomicity/) on the level of a single [document](https://docs.mongodb.com/manual/core/document/).



For examples, see [Insert Documents](https://docs.mongodb.com/manual/tutorial/insert-documents/).

## Read Operations

Read operations retrieves [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format) from a [collection](https://docs.mongodb.com/manual/core/databases-and-collections/#collections); i.e. queries a collection for documents. MongoDB provides the following methods to read documents from a collection:

* [db.collection.find()](https://docs.mongodb.com/manual/reference/method/db.collection.find/#db.collection.find)

You can specify [query filters or criteria](https://docs.mongodb.com/manual/tutorial/query-documents/#read-operations-query-argument) that identify the documents to return.



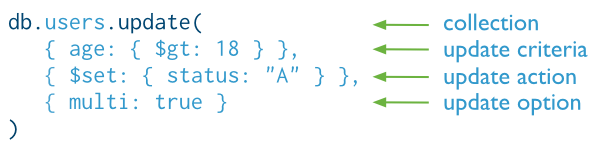
## Update Operations

Update operations modify existing [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format) in a [collection](https://docs.mongodb.com/manual/core/databases-and-collections/#collections). MongoDB provides the following methods to update documents of a collection:

* [db.collection.update()](https://docs.mongodb.com/manual/reference/method/db.collection.update/#db.collection.update)
* [db.collection.updateOne()](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#db.collection.updateOne) New in version 3.2
* [db.collection.updateMany()](https://docs.mongodb.com/manual/reference/method/db.collection.updateMany/#db.collection.updateMany) New in version 3.2
* [db.collection.replaceOne()](https://docs.mongodb.com/manual/reference/method/db.collection.replaceOne/#db.collection.replaceOne) New in version 3.2

In MongoDB, update operations target a single collection. All write operations in MongoDB are [atomic](https://docs.mongodb.com/manual/core/write-operations-atomicity/) on the level of a single document.

You can specify criteria, or filters, that identify the documents to update. These [filters](https://docs.mongodb.com/manual/core/document/#document-query-filter) use the same syntax as read operations.



## Delete Operations

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

* [db.collection.remove()](https://docs.mongodb.com/manual/reference/method/db.collection.remove/#db.collection.remove)
* [db.collection.deleteOne()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteOne/#db.collection.deleteOne) New in version 3.2
* [db.collection.deleteMany()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteMany/#db.collection.deleteMany) New in version 3.2

In MongoDB, delete operations target a single [collection](https://docs.mongodb.com/manual/reference/glossary/#term-collection). All write operations in MongoDB are [atomic](https://docs.mongodb.com/manual/core/write-operations-atomicity/) on the level of a single document.

You can specify criteria, or filters, that identify the documents to remove. These [filters](https://docs.mongodb.com/manual/core/document/#document-query-filter) use the same syntax as read operations.

