Data Structuring and NoSQL Databases
Larbi Boubchir

Tutorial MongoDB Install

Option 1: Official Tutorials (For Lab PCs, go directly to Option 2).

https://docs.mongodb.com/manual/installation/#mongodb-community-edition

Option 2: Quick/Easy Tutorial

Download (Note: Before clining download from the page select the right version of your OS)

1) Go on the following link: https://www.mongodb.com/download-center/community
2) Select the MongoDB version. If the one you're looking for is not in the list, go to "All version binaries" on the right menu.
3) Select your OS version
4) Select TGZ in Package
5) Click on Download

Extract it in a folder. We'll call the path to Mongodb folder: /path/to/mongo/

3. **Launch Mongo Server**: In a terminal, run these commands:

```
cd /path/to/mongo/
cd bin
mkdir dataFolder
./mongod --dbpath dataFolder
```

Don't close the terminal window or you'll kill the server !

4. **Launch Mongo Client**: Open another terminal window. Launch following commands:

```
cd /path/to/mongo/
cd bin
./mongo
```

Note: You can also have client with graphical interface. i.e.:
https://stackoverflow.com/a/6691013

LAUNCHING :

Server launching : $ ./mongod

```
hydra:bin larbi$ ./mongod
2019-10-04T20:24:43.136+0200 I STORAGE  [main] Max cache overflow file size custom option: 0
2019-10-04T20:24:43.137+0200 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] MongoDB starting : pid=4942 port=27017 dbpath=/data/db 64-bit host=hydra.local
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] db version v4.0.12
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] git version: 5776e3cbf9e7afe86e6b29e22520ffb6766e95d4
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] allocator: system
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] modules: none
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] build environment:
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten]     distarch: x86_64
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten]     target_arch: x86_64
2019-10-04T20:24:43.149+0200 I CONTROL  [initandlisten] options: {}
2019-10-04T20:24:43.208+0200 W STORAGE  [initandlisten] Detected unclean shutdown - /data/db/mongod.lock is not empty.
2019-10-04T20:24:43.249+0200 I STORAGE  [initandlisten] Detected data files in /data/db created by the 'wiredTiger' storage engine, so setting the activ
e storage engine to 'wiredTiger'.
2019-10-04T20:24:43.250+0200 W STORAGE  [initandlisten] Recovering data from the last clean checkpoint.
2019-10-04T20:24:43.250+0200 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=7680M,cache_overflow=(file_max=0M),session_max=20000,e
viction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(c
lose_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-10-04T20:24:43.969+0200 I STORAGE  [initandlisten] WiredTiger message [1570213483:969617][4942:0x11819c5c0], txn-recover: Main recovery loop: start
ing at 1/62848 to 2/256
2019-10-04T20:24:43.971+0200 I STORAGE  [initandlisten] WiredTiger message [1570213483:971002][4942:0x11819c5c0], txn-recover: Recovering log 1 through
2
2019-10-04T20:24:44.021+0200 I STORAGE  [initandlisten] WiredTiger message [1570213484:21263][4942:0x11819c5c0], txn-recover: Recovering log 2 through 2
2019-10-04T20:24:44.060+0200 I STORAGE  [initandlisten] WiredTiger message [1570213484:60294][4942:0x11819c5c0], txn-recover: Set global recovery timest
amp: 0
2019-10-04T20:24:44.117+0200 I RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
2019-10-04T20:24:44.217+0200 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory '/data/db/diagnostic.data'
2019-10-04T20:24:44.286+0200 I NETWORK  [initandlisten] waiting for connections on port 27017
2019-10-04T20:24:45.064+0200 I FTDC     [ftdc] Unclean full-time diagnostic data capture shutdown detected, found interim file, some metrics may have be
en lost. OK
2019-10-04T20:25:03.709+0200 I NETWORK  [listener] connection accepted from 127.0.0.1:55116 #1 (1 connection now open)
2019-10-04T20:25:03.709+0200 I NETWORK  [conn1] received client metadata from 127.0.0.1:55116 conn1: { application: { name: "MongoDB Shell" }, driver: {
 name: "MongoDB Internal Client", version: "4.0.12" }, os: { type: "Darwin", name: "Mac OS X", architecture: "x86_64", version: "18.7.0" } }
```

Client launching : $ /bin/mongo

```
hydra:bin larbi$ ./mongo
MongoDB shell version v4.0.12
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("341cc1d3-e3a8-4493-baf9-d79082a80212") }
MongoDB server version: 4.0.12
Server has startup warnings:
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten]
2019-10-04T20:24:44.186+0200 I CONTROL  [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

>
```

To familiarize yourself with the environment try these few commands
> help
> db. help()

« Hello World » EXAMPLE

```
> db
test
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use persons
switched to db persons
> p1={name:"person1",zip:75000};
{ "name" : "person1", "zip" : 75000 }
> p2={name:"person2",zip:92000};
{ "name" : "person2", "zip" : 92000 }
> p3={name:"person3",zip:94000};
{ "name" : "person3", "zip" : 94000 }
> p4={name:"person4",zip:91000};
{ "name" : "person4", "zip" : 91000 }
> db.location.save(p1);
WriteResult({ "nInserted" : 1 })
> db.location.save(p2);
WriteResult({ "nInserted" : 1 })
> db.location.save(p3);
WriteResult({ "nInserted" : 1 })
> db.location.save(p4);
WriteResult({ "nInserted" : 1 })
>
```

QUERYING

Display data

> db.location.find()

```
> db.location.find()
{ "_id" : ObjectId("5d97911d1f338632e0237280"), "name" : "person1", "zip" : 75000 }
{ "_id" : ObjectId("5d97912b1f338632e0237281"), "name" : "person2", "zip" : 92000 }
{ "_id" : ObjectId("5d97912e1f338632e0237282"), "name" : "person3", "zip" : 94000 }
{ "_id" : ObjectId("5d9791311f338632e0237283"), "name" : "person4", "zip" : 91000 }
>
```

- ObjectID: Unique identifier of each document. Declared explicitly by the developer or implicitly by mongoDB
- Format: BSON (binary JSON): binary serialization of "JSON-Like" documents with an extension for other types (date, binary data, etc.)
- Specification of BSON: http://bsonspec.org/

Add records to the collection « location »

```
> p5={name:"person5",zip:95000};
{ "name" : "person5", "zip" : 95000 }
> p6={name:"person6",zip:77000};
{ "name" : "person6", "zip" : 77000 }
> db.location.save(p5);
WriteResult({ "nInserted" : 1 })
> db.location.save(p6);
WriteResult({ "nInserted" : 1 })
>
```

Find record with zip code: 75000 (Idem for name :''person1'')

```
> db.location.find({zip:75000});
{ "_id" : ObjectId("5d97911d1f338632e0237280"), "name" : "person1", "zip" : 75000 }
>
```