

Install MongoDB (Ubuntu server 20.04)

Import the public key used by the package management system.

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
```

Create a list file for MongoDB

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
```

```
sudo apt-get update
```

Install the MongoDB packages.

```
sudo apt-get install -y mongodb-org
```

```
rvv@usrv1:/var/log/mongodb$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("7643af92-190d-42d1-a6fe-9ee59dedd64f") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-10-22T10:03:14.443+0000 I STORAGE [initandlisten]
2022-10-22T10:03:14.443+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-10-22T10:03:14.443+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten]
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten]
> show collections
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
> use mydb
switched to db mydb
> db.books.insert({title:"MyBook", likes:100});
WriteResult({ "nInserted" : 1 })
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
mydb 0.000GB
> show collections
books
> -
```

Create collection books in mydb dbs. Insert 2 elemrnts in it. Find and Update and finally drop collection.

```
> db.books.find()
{ "_id" : ObjectId("6353caa4d748aeeb3feea704"), "title" : "MyBook", "likes" : 100 }
> db.books.insert({title:"Power Book", Author:"Vladimir", Pages:123});
WriteResult({ "nInserted" : 1 })
> db.books.find()
{ "_id" : ObjectId("6353caa4d748aeeb3feea704"), "title" : "MyBook", "likes" : 100 }
{ "_id" : ObjectId("6353cc7ed748aeeb3feea705"), "title" : "Power Book", "Author" : "Vladimir", "Page
s" : 123 }
> db.books.find({likes:100})
{ "_id" : ObjectId("6353caa4d748aeeb3feea704"), "title" : "MyBook", "likes" : 100 }
> db.books.find({Author:"Vlad"})
> db.books.find({Author:"Vladimir"})
{ "_id" : ObjectId("6353cc7ed748aeeb3feea705"), "title" : "Power Book", "Author" : "Vladimir", "Page
s" : 123 }
> db.books.find({Pages:{> 100}})
2022-10-22T10:59:02.430+0000 E QUERY [thread1] SyntaxError: missing : after property id @(shell):
1:26
> db.books.find({Pages:{>:100}})
{ "_id" : ObjectId("6353cc7ed748aeeb3feea705"), "title" : "Power Book", "Author" : "Vladimir", "Page
s" : 123 }
> db.books.updateOne({Author:"Vladimir"},{$set:{Pages:500}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.books.find({Pages:{>:100}})
{ "_id" : ObjectId("6353cc7ed748aeeb3feea705"), "title" : "Power Book", "Author" : "Vladimir", "Page
s" : 500 }
> db.books.deleteOne({likes:100})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.books.find()
{ "_id" : ObjectId("6353cc7ed748aeeb3feea705"), "title" : "Power Book", "Author" : "Vladimir", "Page
s" : 500 }
> db.books.drop()
true

> show.collections
2022-10-22T11:03:38.576+0000 E QUERY [thread1] ReferenceError: show is not defined :
@(shell):1:1
> show collections
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
> -
```

Create BackUp

```
> show collections
books
> db.books.find()
{ "_id" : ObjectId("6353d098d748aeeb3feea706"), "title" : "Cooking", "Author" : "Olga", "Pages" : 345 }
{ "_id" : ObjectId("6353d0add748aeeb3feea707"), "title" : "My path", "Author" : "Sveta", "Pages" : 200 }
{ "_id" : ObjectId("6353d0e1d748aeeb3feea708"), "title" : "Eatrh climate", "Recenzent" : "Ivan", "Author" : "Denis" }
> ^C
bye
rvv@usrv1:/var/log/mongodb$ cd ~
rvv@usrv1:~$ mongodump --out ~/
2022-10-22T11:18:03.701+0000    writing admin.system.version to
2022-10-22T11:18:03.703+0000    done dumping admin.system.version (1 document)
2022-10-22T11:18:03.704+0000    writing mydb.books to
2022-10-22T11:18:03.707+0000    done dumping mydb.books (3 documents)
rvv@usrv1:~$ ls -la
total 56
drwxr-xr-x  8 rvv  rvv  4096 Oct 22 11:18 .
drwxr-xr-x  3 root root  4096 Oct 22 09:35 ..
drwxrwxr-x  2 rvv  rvv  4096 Oct 22 11:18 admin
-rw-----  1 rvv  rvv   106 Oct 22 09:45 .bash_history
-rw-r--r--  1 rvv  rvv   220 Feb 25  2020 .bash_logout
-rw-r--r--  1 rvv  rvv  3771 Feb 25  2020 .bashrc
drwx-----  3 rvv  rvv  4096 Oct 22 10:05 .cache
drwx-----  4 rvv  rvv  4096 Oct 22 10:14 .config
-rw-----  1 rvv  rvv   846 Oct 22 11:17 .dbshell
drwx-----  3 rvv  rvv  4096 Oct 22 10:05 .local
-rw-----  1 rvv  rvv     0 Oct 22 10:23 .mongorc.js
drwxrwxr-x  2 rvv  rvv  4096 Oct 22 11:18 mydb
-rw-r--r--  1 rvv  rvv   807 Feb 25  2020 .profile
drwx-----  2 rvv  rvv  4096 Oct 22 09:35 .ssh
-rw-r--r--  1 rvv  rvv     0 Oct 22 09:37 .sudo_as_admin_successful
-rw-----  1 rvv  rvv    51 Oct 22 10:03 .Xauthority
rvv@usrv1:~$ _
> show collections
books
> db.books.drop()
true
> show collections
> _
```

Restore db

```
rvv@usrv1:~$ mongorestore --db=mydb mydb/
2022-10-22T11:35:11.489+0000 the --db and --collection args should only be used when restoring from
a BSON file. Other uses are deprecated and will not exist in the future; use --nsInclude instead
2022-10-22T11:35:11.490+0000 building a list of collections to restore from mydb dir
2022-10-22T11:35:11.492+0000 reading metadata for mydb.books from mydb/books.metadata.json
2022-10-22T11:35:11.514+0000 restoring mydb.books from mydb/books.bson
2022-10-22T11:35:11.518+0000 no indexes to restore
2022-10-22T11:35:11.519+0000 finished restoring mydb.books (3 documents)
2022-10-22T11:35:11.521+0000 done
rvv@usrv1:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongod://127.0.0.1:27017
Implicit session: session { "id" : UUID("96b8e385-687f-4d1b-9903-555b4f3c6704") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-10-22T10:03:14.443+0000 I STORAGE [initandlisten]
2022-10-22T10:03:14.443+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly
recommended with the WiredTiger storage engine
2022-10-22T10:03:14.443+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten]
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the
database.
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration
is unrestricted.
2022-10-22T10:03:20.559+0000 I CONTROL [initandlisten]
> use mydb
switched to db mydb
> show collections
books
```