

1. Install latest version of MongoDB from apt-get repository

```
ayush@ayush:~$ sudo apt install -y mongodb
[sudo] password for ayush:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libboost-program-options1.65.1 libgoogle-perftools4 libpcrecpp0v5
  libsnappy1v5 libtcmalloc-minimal4 libyaml-cpp0.5v5 mongo-tools
  mongodb-clients mongodb-server mongodb-server-core
The following NEW packages will be installed:
  libboost-program-options1.65.1 libgoogle-perftools4 libpcrecpp0v5
  libsnappy1v5 libtcmalloc-minimal4 libyaml-cpp0.5v5 mongo-tools mongodb
  mongodb-clients mongodb-server mongodb-server-core
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 53.4 MB of archives.
After this operation, 217 MB of additional disk space will be used.
```

2. Create a database student

```
> use student
switched to db student
> db
student
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> 
```

3. Insert operation : 5 students data (Name, Contact, City, Roll No, Branch)

```
> db.createCollection("studentdata")
{ "ok" : 1 }
> show collections
studentdata
> db.studentdata.insert({
... name: 'Ayush',
... contact: '7011288349',
... city: 'ghaziabad',
... roll_no: '2011702',
... branch: 'cs'
... })
WriteResult({ "nInserted" : 1 })
> db.studentdata.insert({ name: 'Aman', contact: '7011312111', city: 'delhi', roll_no: '2011745', branch: 'me' })
WriteResult({ "nInserted" : 1 })
> db.studentdata.insert({ name: 'Amit', contact: '7011444343', city: 'noida', roll_no: '2011722', branch: 'ce' })
WriteResult({ "nInserted" : 1 })
> db.studentdata.insert({ name: 'Akash', contact: '7011435353', city: 'gurgaon', roll_no: '2011767', branch: 'ec' })
WriteResult({ "nInserted" : 1 })
> db.studentdata.insert({ name: 'Akash', contact: '7011435353', city: 'dehradun', roll_no: '2011656', branch: 'el' })
WriteResult({ "nInserted" : 1 })
>
```

4. Read operation : All the students belong to a particular city

```
> db.studentdata.find({}, {
...   "name": 1,
...   "city": 1
... });
{ "_id" : ObjectId("5e4b599386da36da1385b734"), "name" : "Ayush", "city" : "ghaziabad" }
{ "_id" : ObjectId("5e4b59e086da36da1385b735"), "name" : "Aman", "city" : "delhi" }
{ "_id" : ObjectId("5e4b5a2686da36da1385b736"), "name" : "Amit", "city" : "noida" }
{ "_id" : ObjectId("5e4b5a5f86da36da1385b737"), "name" : "Akash", "city" : "gurgaon" }
{ "_id" : ObjectId("5e4b5aa886da36da1385b738"), "name" : "Akash", "city" : "dehradun" }
>
```

5. Update operation : Update the branch of all the students to CSE

```
> db.studentdata.update({},{$set:{branch: "cse"}},{multi:true});
WriteResult({ "nMatched" : 5, "nUpserted" : 0, "nModified" : 4 })
> db.studentdata.find({})
{ "_id" : ObjectId("5e4b599386da36da1385b734"), "name" : "Ayush", "contact" : "7011288349",
  "city" : "ghaziabad", "roll_no" : "2011702", "branch" : "cse" }
{ "_id" : ObjectId("5e4b59e086da36da1385b735"), "name" : "Aman", "contact" : "7011312111",
  "city" : "delhi", "roll_no" : "2011745", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a2686da36da1385b736"), "name" : "Amit", "contact" : "7011444343",
  "city" : "noida", "roll_no" : "2011722", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a5f86da36da1385b737"), "name" : "Akash", "contact" : "7011435353",
  "city" : "gurgaon", "roll_no" : "2011767", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5aa886da36da1385b738"), "name" : "Akash", "contact" : "7011435353",
  "city" : "dehradun", "roll_no" : "2011656", "branch" : "cse" }
> █
```

6. Take dump of the database

```
ayush@ayush:~$ mongodump --db student --collection studentdata
2020-02-18T09:53:15.452+0530   writing student.studentdata to
2020-02-18T09:53:15.453+0530   done dumping student.studentdata (5 documents)
ayush@ayush:~$ █
```

7. Delete operation : Delete the record of last 2 students according to the roll number


```

> db.studentdata.find().sort({roll_no:1});
{ "_id" : ObjectId("5e4b5aa886da36da1385b738"), "name" : "Akash", "contact" : "7011435353", "city" : "dehradun", "roll_no" : "2011656", "branch" : "cse" }
{ "_id" : ObjectId("5e4b599386da36da1385b734"), "name" : "Ayush", "contact" : "7011288349", "city" : "ghaziabad", "roll_no" : "2011702", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a2686da36da1385b736"), "name" : "Amit", "contact" : "7011444343", "city" : "noida", "roll_no" : "2011722", "branch" : "cse" }
{ "_id" : ObjectId("5e4b59e086da36da1385b735"), "name" : "Aman", "contact" : "7011312111", "city" : "delhi", "roll_no" : "2011745", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a5f86da36da1385b737"), "name" : "Akash", "contact" : "7011435353", "city" : "gurgoan", "roll_no" : "2011767", "branch" : "cse" }
> const roll = db.studentdata.find({}).sort({roll_no:-1}).limit(2).toArray().map(ele => ele.roll_no);
> roll
[ "2011767", "2011745" ]
> db.studentdata.remove({roll_no: {$in: roll}})
WriteResult({ "nRemoved" : 2 })
> db.studentdata.find().sort({roll_no:1});
{ "_id" : ObjectId("5e4b5aa886da36da1385b738"), "name" : "Akash", "contact" : "7011435353", "city" : "dehradun", "roll_no" : "2011656", "branch" : "cse" }
{ "_id" : ObjectId("5e4b599386da36da1385b734"), "name" : "Ayush", "contact" : "7011288349", "city" : "ghaziabad", "roll_no" : "2011702", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a2686da36da1385b736"), "name" : "Amit", "contact" : "7011444343", "city" : "noida", "roll_no" : "2011722", "branch" : "cse" }
>

```

8. Drop the database

```

> use student
switched to db student
> db.dropDatabase()
{ "dropped" : "student", "ok" : 1 }
> use student
switched to db student
> db.dropDatabase()
{ "ok" : 1 }
> dbs
2020-02-18T16:42:58.475+0530 E QUERY [thread1] ReferenceError: dbs is not defined :
@(shell):1:1
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB

```

9. Restore the database again to have the full data

```

ayush@ayush:~$ mongorestore --db student --verbose dump/student/studentdata.
studentdata.bson          studentdata.metadata.json
ayush@ayush:~$ mongorestore --db student --verbose dump/student/
2020-02-18T16:48:55.995+0530 using write concern: w='1', j=false, fsync=false, wtimeout=0
2020-02-18T16:48:55.995+0530 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 in
stead
2020-02-18T16:48:55.995+0530 building a list of collections to restore from dump/student dir
2020-02-18T16:48:55.995+0530 found collection student.studentdata bson to restore to student.s
tudentdata
2020-02-18T16:48:55.995+0530 found collection metadata from student.studentdata to restore to
student.studentdata
2020-02-18T16:48:55.995+0530 reading metadata for student.studentdata from dump/student/studen
tdata.metadata.json
2020-02-18T16:48:55.995+0530 creating collection student.studentdata using options from metada
ta
2020-02-18T16:48:56.026+0530 restoring student.studentdata from dump/student/studentdata.bson
2020-02-18T16:48:56.030+0530 no indexes to restore
2020-02-18T16:48:56.030+0530 finished restoring student.studentdata (5 documents)
2020-02-18T16:48:56.030+0530 done
ayush@ayush:~$

```

```

> use student
switched to db student
> db.studentdata.find();
{ "_id" : ObjectId("5e4b599386da36da1385b734"), "name" : "Ayush", "contact" : "7011288349", "city"
: "ghaziabad", "roll_no" : "2011702", "branch" : "cse" }
{ "_id" : ObjectId("5e4b59e086da36da1385b735"), "name" : "Aman", "contact" : "7011312111", "city"
: "delhi", "roll_no" : "2011745", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a2686da36da1385b736"), "name" : "Amit", "contact" : "7011444343", "city"
: "noida", "roll_no" : "2011722", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5a5f86da36da1385b737"), "name" : "Akash", "contact" : "7011435353", "city"
: "gurgaon", "roll_no" : "2011767", "branch" : "cse" }
{ "_id" : ObjectId("5e4b5aa886da36da1385b738"), "name" : "Akash", "contact" : "7011435353", "city"
: "dehradun", "roll_no" : "2011656", "branch" : "cse" }
>

```

10. Enable authentication on the Mongo

```

> use admin
switched to db admin
> db.createUser({ user: "ayush", pwd: "ayush123", roles: [{role: "root", db: "admin"}]});
Successfully added user: {
  "user" : "ayush",
  "roles" : [
    {
      "role" : "root",
      "db" : "admin"
    }
  ]
}
>

```

```

ayush@ayush:~$ sudo vim /etc/mongodb.conf
[sudo] password for ayush:
ayush@ayush:~$ sudo vim /etc/mongodb.conf | grep auth
Vim: Warning: Output is not to a terminal
#noauth = true
auth = true

[1]+  Stopped                  sudo vim /etc/mongodb.conf | grep --color=auto auth
ayush@ayush:~$ exit
There are stopped jobs.
ayush@ayush:~$ sudo service mongod restart
Failed to restart mongod.service: Unit mongod.service not found.
ayush@ayush:~$ sudo service mongod restart
ayush@ayush:~$ mongo --port 27017 -u "ayush" -p "ayush123" --authenticationDatabase "admin"
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/
MongoDB server version: 3.6.3
Server has startup warnings:
2020-02-18T16:56:28.520+0530 I STORAGE [initandlisten]
2020-02-18T16:56:28.520+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is s
trongly recommended with the WiredTiger storage engine
2020-02-18T16:56:28.520+0530 I STORAGE [initandlisten] **          See http://dochub.mongodb.org
/core/prodnotes-filesystem

```

11. Install another version of MongoDB from source (Version 2.6.3) and run it on port 27009


```
ayush@ayush:~$ wget https://fastdl.mongodb.org/src/mongodb-src-r4.2.3.tar.gz
--2020-02-19 10:00:07-- https://fastdl.mongodb.org/src/mongodb-src-r4.2.3.tar.g
z
Resolving fastdl.mongodb.org (fastdl.mongodb.org)... 99.86.47.34, 99.86.47.53, 9
9.86.47.42, ...
Connecting to fastdl.mongodb.org (fastdl.mongodb.org)|99.86.47.34|:443... connec
ted.
HTTP request sent, awaiting response... 200 OK
Length: 61565615 (59M) [application/gzip]
Saving to: 'mongodb-src-r4.2.3.tar.gz'

mongodb-src-r4.2.3. 100%[=====>] 58.71M 332KB/s in 84s

2020-02-19 10:01:32 (717 KB/s) - 'mongodb-src-r4.2.3.tar.gz' saved [61565615/615
65615]
```

```
ayush@ayush:~/mongo$ ls
mongodb-linux-x86_64-ubuntu1804-4.2.3.tgz
ayush@ayush:~/mongo$ tar zxvf mongodb-linux-x86_64-ubuntu1804-4.2.3.tgz
mongodb-linux-x86_64-ubuntu1804-4.2.3/THIRD-PARTY-NOTICES.gotools
mongodb-linux-x86_64-ubuntu1804-4.2.3/README
mongodb-linux-x86_64-ubuntu1804-4.2.3/THIRD-PARTY-NOTICES
mongodb-linux-x86_64-ubuntu1804-4.2.3/MPL-2
mongodb-linux-x86_64-ubuntu1804-4.2.3/LICENSE-Community.txt
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongodump
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongorestore
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongoexport
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongoimport
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongostat
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongotop
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/bsondump
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongofiles
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongoreplay
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongod
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongos
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/mongo
mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/install_compass
```



```

ayush@ayush:~/mongo$ cd mongodb-linux-x86_64-ubuntu1804-4.2.3/bin/
ayush@ayush:~/mongo/mongodb-linux-x86_64-ubuntu1804-4.2.3/bin$ ls
bsondump          mongod            mongofiles        mongorestore      mongotop
install_compass   mongodump         mongoimport       mongos
mongo             mongoexport       mongoreplay       mongostat
ayush@ayush:~/mongo/mongodb-linux-x86_64-ubuntu1804-4.2.3/bin$ mkdir ../data
ayush@ayush:~/mongo/mongodb-linux-x86_64-ubuntu1804-4.2.3/bin$ mkdir ../../data
ayush@ayush:~/mongo/mongodb-linux-x86_64-ubuntu1804-4.2.3/bin$ mongod --port 270
09 --dbpath ~/mongo/data
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] MongoDB starting : pid=8
630 port=27009 dbpath=/home/ayush/mongo/data 64-bit host=ayush
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] db version v3.6.3
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] git version: 9586e557d54
ef70f9ca4b43c26892cd55257e1a5
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.1.1 11 Sep 2018
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] allocator: tcmalloc
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] modules: none
2020-02-19T10:16:59.533+0530 I CONTROL [initandlisten] build environment:

```

```

--bind_ip 127.0.0.1 to disable this warning.
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten]
2020-02-19T10:17:01.217+0530 I STORAGE [initandlisten] createCollection: admin.
system.version with provided UUID: b8786dde-b207-4c5a-8879-7a62d35ab268
2020-02-19T10:17:01.552+0530 I COMMAND [initandlisten] setting featureCompatibi
lityVersion to 3.6
2020-02-19T10:17:01.568+0530 I STORAGE [initandlisten] createCollection: local.
startup_log with generated UUID: 7143a68d-084c-46df-bf6c-5d62857f6bdf
2020-02-19T10:17:01.870+0530 I FTDC [initandlisten] Initializing full-time d
iagnostic data capture with directory '/home/ayush/mongo/data/diagnostic.data'
2020-02-19T10:17:01.871+0530 I NETWORK [initandlisten] waiting for connections
on port 27009

```

```

ayush@ayush:~/mongo/mongodb-linux-x86_64-ubuntu1804-4.2.3/bin$ ls
bsondump          mongod            mongofiles        mongorestore      mongotop
install_compass   mongodump         mongoimport       mongos
ayush@ayush:~/mongo/mongodb-linux-x86_64-ubuntu1804-4.2.3/bin$ mongo --port 27009
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27009/
MongoDB server version: 3.6.3
Server has startup warnings:
2020-02-19T10:16:59.533+0530 I STORAGE [initandlisten]
2020-02-19T10:16:59.533+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS
filesystem is strongly recommended with the WiredTiger storage engine
2020-02-19T10:16:59.533+0530 I STORAGE [initandlisten] ** See http://do
chub.mongodb.org/core/prodnotes-filesystem
2020-02-19T10:17:01.207+0530 I CONTROL [initandlisten]
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten] ** WARNING: Access contro
l is not enabled for the database.
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten] ** Read and writ
e access to data and configuration is unrestricted.
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten]
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten] ** WARNING: This server i
s bound to localhost.

```



```
agnostic data capture with directory '/home/ayush/mongo/data/diagnostic.data'
2020-02-19T10:17:01.871+0530 I NETWORK [initandlisten] waiting for connections on port 27009
2020-02-19T10:19:03.460+0530 I NETWORK [listener] connection accepted from 127.0.0.1:60100 #1 (1 connection now open)
2020-02-19T10:19:03.460+0530 I NETWORK [conn1] received client metadata from 127.0.0.1:60100 conn: { application: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "3.6.3" }, os: { type: "Linux", name: "Ubuntu", architecture: "x86_64", version: "18.04" } }
```

```
ver with --bind_ip <address> to specify which IP  
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten] **      addresses it  
should serve responses from, or with --bind_ip_all to  
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten] **      bind to all i  
nterfaces. If this behavior is desired, start the  
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten] **      server with -  
--bind_ip 127.0.0.1 to disable this warning.  
2020-02-19T10:17:01.208+0530 I CONTROL [initandlisten]  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
>  
  
> show dbs  
admin    0.000GB  
local    0.000GB
```

12. Create init service of Mongo installed later*

Do entry in rc-local file