

Installation Process

Step 1: First you need to update and upgrade your system repository in order to install MongoDB. Type the following command in your terminal and then press Enter.

```
$ sudo apt update && sudo apt upgrade
```

A terminal window titled 'rishabh@rishabh-Lenovo-V130-15IKB: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'sudo apt update && sudo apt upgrade' being executed. It prompts for the password for 'rishabh'. The output shows several repository updates: 'Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease', 'Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease', 'Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease', 'Ign:4 http://dl.google.com/linux/chrome/deb stable InRelease', 'Hit:5 http://ppa.launchpad.net/obsproject/obs-studio/ubuntu bionic InRelease', 'Hit:6 http://dl.google.com/linux/chrome/deb stable Release', 'Get:7 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]', 'Fetched 88.7 kB in 2s (57.5 kB/s)', and 'Reading package lists... 32%'.

```
rishabh@rishabh-Lenovo-V130-15IKB: ~  
File Edit View Search Terminal Help  
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo apt update && sudo apt upgrade  
[sudo] password for rishabh:  
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Ign:4 http://dl.google.com/linux/chrome/deb stable InRelease  
Hit:5 http://ppa.launchpad.net/obsproject/obs-studio/ubuntu bionic InRelease  
Hit:6 http://dl.google.com/linux/chrome/deb stable Release  
Get:7 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Fetched 88.7 kB in 2s (57.5 kB/s)  
Reading package lists... 32%
```

Step 2: Now, install the MongoDB package using ‘**apt**’. Type the following command and press Enter.

```
$ sudo apt install -y mongodb
```

```
rishabh@rishabh-Lenovo-V130-15IKB: ~  
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo apt install -y mongodb  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  efibootmgr libfwup1  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  libboost-program-options1.65.1 libgoogle-perftools4 libpcrecpp0v5  
  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  
  libtcmalloc-minimal4 libyaml-cpp0.5v5 mongo-tools mongodb mongodb-clients  
  mongodb-server mongodb-server-core  
0 upgraded, 10 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.  
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libboost-program-opti  
ons1.65.1 amd64 1.65.1+dfsg-0ubuntu5 [137 kB]  
Get:2 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libtcmalloc-minimal4  
amd64 2.5-2.2ubuntu3 [91.6 kB]
```

Step 3: Check the service status for MongoDB with the help of following command:

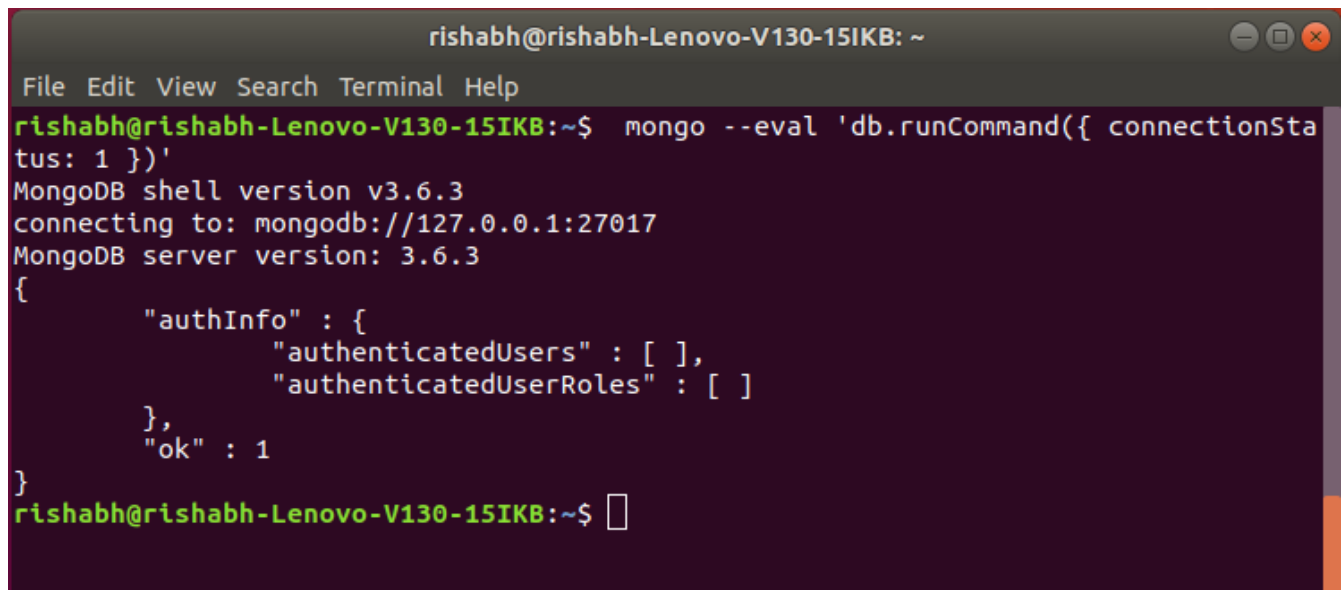
`$ sudo systemctl status mongodb`

```
rishabh@rishabh-Lenovo-V130-15IKB: ~  
File Edit View Search Terminal Help  
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo systemctl status mongodb  
● mongodb.service - An object/document-oriented database  
   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: e  
   Active: active (running) since Tue 2020-03-03 15:49:56 IST; 29s ago  
     Docs: man:mongod(1)  
   Main PID: 14151 (mongod)  
     Tasks: 23 (limit: 4915)  
    CGroup: /system.slice/mongodb.service  
           └─14151 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/  
  
Mar 03 15:49:56 rishabh-Lenovo-V130-15IKB systemd[1]: Started An object/document-  
lines 1-10/10 (END)
```

systemctl verifies that MongoDB server is up and running.

Step 4: Now check if the installation process is done correctly and everything is working fine. Go through the following command:

```
$ mongo --eval 'db.runCommand({ connectionStatus: 1 })'
```

A terminal window titled 'rishabh@rishabh-Lenovo-V130-15IKB: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'rishabh@rishabh-Lenovo-V130-15IKB:~\$'. The command 'mongo --eval 'db.runCommand({ connectionStatus: 1 })'' is entered. The output shows 'MongoDB shell version v3.6.3', 'connecting to: mongodb://127.0.0.1:27017', and 'MongoDB server version: 3.6.3'. The JSON response is: { "authInfo" : { "authenticatedUsers" : [], "authenticatedUserRoles" : [] }, "ok" : 1 }. The prompt returns to 'rishabh@rishabh-Lenovo-V130-15IKB:~\$' with a cursor.

```
rishabh@rishabh-Lenovo-V130-15IKB:~$ mongo --eval 'db.runCommand({ connectionStatus: 1 })'
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
{
  "authInfo" : {
    "authenticatedUsers" : [ ],
    "authenticatedUserRoles" : [ ]
  },
  "ok" : 1
}
rishabh@rishabh-Lenovo-V130-15IKB:~$
```

the value “1” in ok field indicates that the server is working properly with no errors.

Step 5: MongoDB services can be started and stopped with the use of following commands:

To stop running the MongoDB service, use command :

```
$ sudo systemctl stop mongod
```

MongoDB service has been stopped and can be checked by using the status command:

```
$ sudo systemctl status mongod
```

```
rishabh@rishabh-Lenovo-V130-15IKB: ~
File Edit View Search Terminal Help
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo systemctl stop mongodb
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo systemctl status mongodb
● mongod.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: e
   Active: inactive (dead) since Tue 2020-03-03 15:51:06 IST; 10s ago
     Docs: man:mongod(1)
   Process: 14151 ExecStart=/usr/bin/mongod --unixSocketPrefix=${SOCKETPATH} --con
 Main PID: 14151 (code=exited, status=0/SUCCESS)

Mar 03 15:49:56 rishabh-Lenovo-V130-15IKB systemd[1]: Started An object/document-
Mar 03 15:51:06 rishabh-Lenovo-V130-15IKB systemd[1]: Stopping An object/document
Mar 03 15:51:06 rishabh-Lenovo-V130-15IKB systemd[1]: Stopped An object/document-
lines 1-10/10 (END)
```

As it can be seen that the service has stopped, to start the service we can use :

\$ sudo systemctl start mongodb

```
rishabh@rishabh-Lenovo-V130-15IKB: ~
File Edit View Search Terminal Help
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo systemctl start mongodb
rishabh@rishabh-Lenovo-V130-15IKB:~$ sudo systemctl status mongodb
● mongod.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: e
   Active: active (running) since Tue 2020-03-03 15:51:32 IST; 14s ago
     Docs: man:mongod(1)
 Main PID: 14974 (mongod)
    Tasks: 23 (limit: 4915)
   CGroup: /system.slice/mongod.service
           └─14974 /usr/bin/mongod --unixSocketPrefix=/run/mongod --config /etc/

Mar 03 15:51:32 rishabh-Lenovo-V130-15IKB systemd[1]: Started An object/document-
lines 1-10/10 (END)
```

\$mongodb :for run mongod on terminal.

Crud operation In mongodb

CRUD:-create,read,update,delete.

Create:

db.collection.insertOne() It is used to insert a single document in the collection.

db.collection.insertMany() It is used to insert multiple documents in the collection.

db.createCollection() It is used to create an empty collection.

Create database

```
>use database_name
```

“eg:use aids”

```
>db.tablename.insertOne({ name:"Aaditya",age:20,branch:"AIDS"})
```

Or

```
db.tablename.insert({ name:"Aaditya",age:20,branch:"AIDS"})
```

Run successful: { “acknowledged:true,”

“insertedId”:ObjectId(“...”)

}

```
>db.tablename.insertMany([ {name:"Aaditya",age:20,branch:"AIDS"}, {name:"Anurag",age=21,branch="AIDS"}, {name:"Vaibhav",age=20,branch:"E&TC"} ])
```

Run successful: { “acknowledged:true,”

“insertedId”:

[ObjectId(“...”),

ObjectId(“...”)

ObjectId(“...”)]}

Read:

db.collection.find() It is used to retrieve documents from the collection

```
>db.tablename.find().pretty()
```

```
>db.tablename.find()
```

Update:

db.collection.updateOne() It is used to update a single document in the collection that satisfy the given criteria.

db.collection.updateMany() It is used to update multiple documents in the collection that satisfy the given criteria.

db.collection.replaceOne() It is used to replace single document in the collection that satisfy the given criteria.

```
> db.table_name.updateOne({ name:"Aaditya"},{$set:{age=21,name="adi"}})
```

Or

```
db.table_name.update({name:"Aaditya"},{$set:{age=21,name="adi"}})
```

```
>db.table_name.updateMany({},{$set:{year:2020 }})
```

Add year column

```
> db.empDetails.findOneAndUpdate({},{$set:{}})
```

It Used for update and show table

Delete:

It is used to delete a single document from the **db.collection.deleteOne()** collection that satisfy the given criteria.
It is used to delete multiple documents from the **db.collection.deleteMany()** collection that satisfy the given criteria.

```
>db.table_name.deleteOne({name:"Anurag"})
```

```
>db.table_name.deleteMany({})
```

remove():

```
db.table_name.remove({ name:"Anurag"})
```

Save Method:

The **db.collection.save()** method is used to updates an existing document or inserts a new document, depending on its document parameter

```
>db.table_name.save()
```

Logical Operator:

\$and It is used to join query clauses with a logical AND and return all documents that match the given conditions of both clauses.

\$or It is used to join query clauses with a logical OR and return all documents that match the given conditions of either clause.

\$not It is used to invert the effect of the query expressions and return documents that does not match the query expression.

\$nor It is used to join query clauses with a logical NOR and return all documents that fail to match both clauses.

And:

```
>db.table_name.find({$and: [{branch: "AIDS"}, {joiningYear: 2022}]}).pretty()
```

Or:

```
>db.contributor.find({$or: [{branch: "AIDS"}, {joiningYear: 2022}]}).pretty()
```

Not:

In this example, we are retrieving only those employee's documents whose salary is not greater than 2000

```
>db.table_name.find({salary: {$not: {$gt: 2000}}}).pretty()
```

Nor:

In this example, we are retrieving only those employee's documents whose salary is not 3000 and whose branch is not ECE

```
>db.table_name.find({$nor: [{salary: 3000}, {branch: "ECE"}]}).pretty()
```