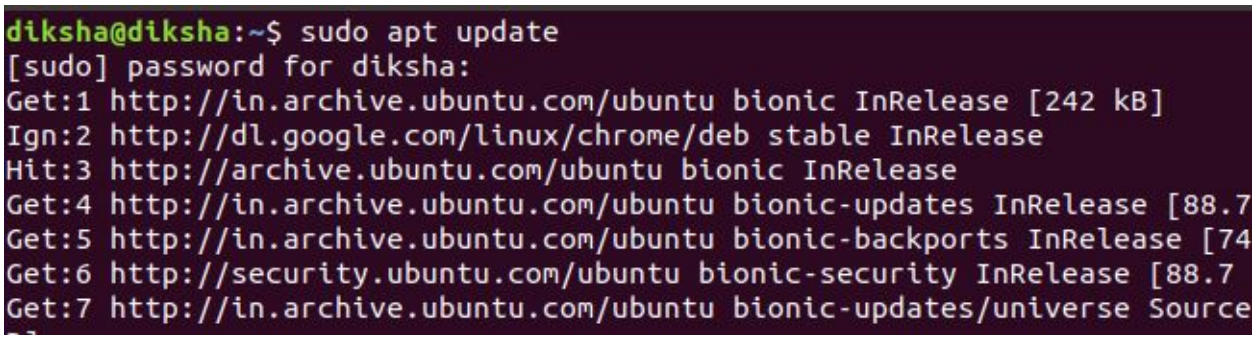
ASSESSMENT ON: MONGODB AND MYSQL

mongodb

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

Ans:

- First, update the packages list to have the most recent version of the repository listings



- Now install the MongoDB package itself:
- Next, let's verify that the server is running and works correctly.
- First, check the service's status:
- According to systemd, the MongoDB server is up and running.
- We can verify this further by actually connecting to the database server and executing a diagnos

  command
- A value of 1 for the ok field in the response indicates that the server is working properly.

```
diksha@diksha:~$ sudo systemctl status mongodb
● mongodb.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset:
   Active: active (running) since Sun 2020-02-16 12:28:39 IST; 1min 36s ago
     Docs: man:mongod(1)
 Main PID: 23407 (mongod)
    Tasks: 23 (limit: 4915)
   CGroup: /system.slice/mongodb.service
           └─23407 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc

Feb 16 12:28:39 diksha systemd[1]: Started An object/document-oriented database.

diksha@diksha:~$ 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" : [ ]
gle Chrome},
        "ok" : 1
}
diksha@diksha:~$
```

2. **Create a database student**

use DATABASE_NAME is used to create databasell

```
> use student
switched to db student
>
```

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

```
> db.student.insert({Name:"Jay", Contact:123 , City:"ddun", RollNo:1, Branch:"CS"})
WriteResult({ "nInserted" : 1 })
> db.student.insert({Name:"Joe", Contact:1256 , City:"delhi", RollNo:2, Branch:"BDA"
})
WriteResult({ "nInserted" : 1 })
> db.student.insert({Name:"Jim", Contact:125 , City:"Meerut", RollNo:7, Branch:"Clou
d"})
WriteResult({ "nInserted" : 1 })
> db.student.insert({Name:"Jam", Contact:12589 , City:"Nagpur", RollNo:10, Branch:"I
T"})
WriteResult({ "nInserted" : 1 })
> db.student.insert({Name:"Jack", Contact:1258 , City:"Delhi", RollNo:12, Branch:"CS
"})
oogle Chrome
WriteResult({ "nInserted" : 1 })
>
```

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

```
>
>
> db.student.find( { City: { $eq : "delhi" } } );
{ "_id" : ObjectId("5e4a43e1b5a5c9d445e3982c"), "Name" : "Joe", "Contact" : 1256, "C
ity" : "delhi", "RollNo" : 2, "Branch" : "BDA" }
> db.student.find( { City: { $eq : "Delhi" } } );
{ "_id" : ObjectId("5e4a4438b5a5c9d445e3982f"), "Name" : "Jack", "Contact" : 1258, "
City" : "Delhi", "RollNo" : 12, "Branch" : "CS" }
>
```

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

```
> db.student.update( {},{ $set: { Branch: "CSE" } }, { multi: true } )
WriteResult({ "nMatched" : 6, "nUpserted" : 0, "nModified" : 6 })
>
```

```
> db.student.update( {},{ $set: { Branch: "CSE" } }, { multi: true } )
WriteResult({ "nMatched" : 6, "nUpserted" : 0, "nModified" : 6 })
> db.student.find()
{ "_id" : ObjectId("5e4a4254b5a5c9d445e3982a"), "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a4353b5a5c9d445e3982b"), "Name" : "Jay", "Contact" : 123, "Ci
ty" : "ddun", "RollNo" : 1, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a43e1b5a5c9d445e3982c"), "Name" : "Joe", "Contact" : 1256, "C
ity" : "delhi", "RollNo" : 2, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a43ffb5a5c9d445e3982d"), "Name" : "Jim", "Contact" : 125, "Ci
ty" : "Meerut", "RollNo" : 7, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a441eb5a5c9d445e3982e"), "Name" : "Jam", "Contact" : 12589, "
City" : "Nagpur", "RollNo" : 10, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a4438b5a5c9d445e3982f"), "Name" : "Jack", "Contact" : 1258, "
City" : "Delhi", "RollNo" : 12, "Branch" : "CSE" }
>
```

6. **Take dump of the database**

```
> show collections
student
> show databases
admin    0.000GB
config   0.000GB
local    0.000GB
student  0.000GB
> ^C
bye
diksha@diksha:~$ mongodump --db student -o dump_directory
2020-02-17T14:50:39.790+0530    writing student.student to
2020-02-17T14:50:39.792+0530    done dumping student.student (6 documents)
diksha@diksha:~$
```

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

```
> db.student.remove( { RollNo: { $gt:2 } } )
WriteResult({ "nRemoved" : 3 })
> db.student.find()
{ "_id" : ObjectId("5e4a4254b5a5c9d445e3982a"), "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a4353b5a5c9d445e3982b"), "Name" : "Jay", "Contact" : 123, "Ci
ty" : "ddun", "RollNo" : 1, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a43e1b5a5c9d445e3982c"), "Name" : "Joe", "Contact" : 1256, "C
ity" : "delhi", "RollNo" : 2, "Branch" : "CSE" }
>
```

8. **Drop the database**

```
> db.student.drop()
true
> db.student.find()
>
```

9. **Restore the database again to have the full data**

```
diksha@diksha:~/dump_directory/student$ mongorestore --db student student.bson
2020-02-17T15:49:09.550+0530    checking for collection data in student.bson
2020-02-17T15:49:09.550+0530    reading metadata for student.student from student.metadata.json
2020-02-17T15:49:09.576+0530    restoring student.student from student.bson
2020-02-17T15:49:09.637+0530    no indexes to restore
2020-02-17T15:49:09.637+0530    finished restoring student.student (6 documents)
2020-02-17T15:49:09.637+0530    done
```

```
> show databases
admin    0.000GB
config   0.000GB
local    0.000GB
student  0.000GB
> use student
switched to db student
> db.student.find()
{ "_id" : ObjectId("5e4a4254b5a5c9d445e3982a"), "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a43e1b5a5c9d445e3982c"), "Name" : "Joe", "Contact" : 1256, "City" : "delhi", "RollNo" : 2, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a43ffb5a5c9d445e3982d"), "Name" : "Jim", "Contact" : 125, "City" : "Meerut", "RollNo" : 7, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a441eb5a5c9d445e3982e"), "Name" : "Jam", "Contact" : 12589, "City" : "Nagpur", "RollNo" : 10, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a4438b5a5c9d445e3982f"), "Name" : "Jack", "Contact" : 1258, "City" : "Delhi", "RollNo" : 12, "Branch" : "CSE" }
{ "_id" : ObjectId("5e4a4353b5a5c9d445e3982b"), "Name" : "Jay", "Contact" : 123, "City" : "ddun", "RollNo" : 1, "Branch" : "CSE" }
>
>
```

10. **Enable authentication on the Mongo**

```
diksha@diksha:~$ mongod --port 27017 --dbpath /data/db1
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] MongoDB starting : pid=29086 port=27017 dbpath=/data
/db1 64-bit host=diksha
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] db version v3.6.3
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] git version: 9586e557d54ef70f9ca4b43c26892cd55257e1a
5
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1  11 Sep 2018
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] allocator: tcmalloc
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] modules: none
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] build environment:
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten]     distarch: x86_64
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten]     target_arch: x86_64
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] options: { net: { port: 27017 }, storage: { dbPath:
"/data/db1" } }
2020-02-17T16:41:08.430+0530 I STORAGE  [initandlisten] exception in initAndListen: NonExistentPath: Data di
rectory /data/db1 not found., terminating
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] now exiting
2020-02-17T16:41:08.430+0530 I CONTROL  [initandlisten] shutting down with code:100
```

Step 1: Start MongoDB without access control

```
diksha@diksha:~$ mongo --port 27017
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/
MongoDB server version: 3.6.3
Server has startup warnings:
2020-02-16T12:28:39.491+0530 I STORAGE  [initandlisten]
2020-02-16T12:28:39.491+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly rec
ommended with the WiredTiger storage engine
2020-02-16T12:28:39.491+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodn
otes-filesystem
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten]
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the da
tabase.
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten] **          Read and write access to data and config
uration is unrestricted.
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten]
```

Step 2: Create the user administrator

```
> use admin
switched to db admin
> db.createUser(
... {
...  user:  "root",
...  pwd: "diksha",
...  roles: [ { role: "userAdminAnyDatabase", db: "admin"} ]
... }
... )
Successfully added user: {
        "user" : "root",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ]
}
>
>
>
> ^C
bye
```

Step 4: Re-start the MongoDB instance with access control

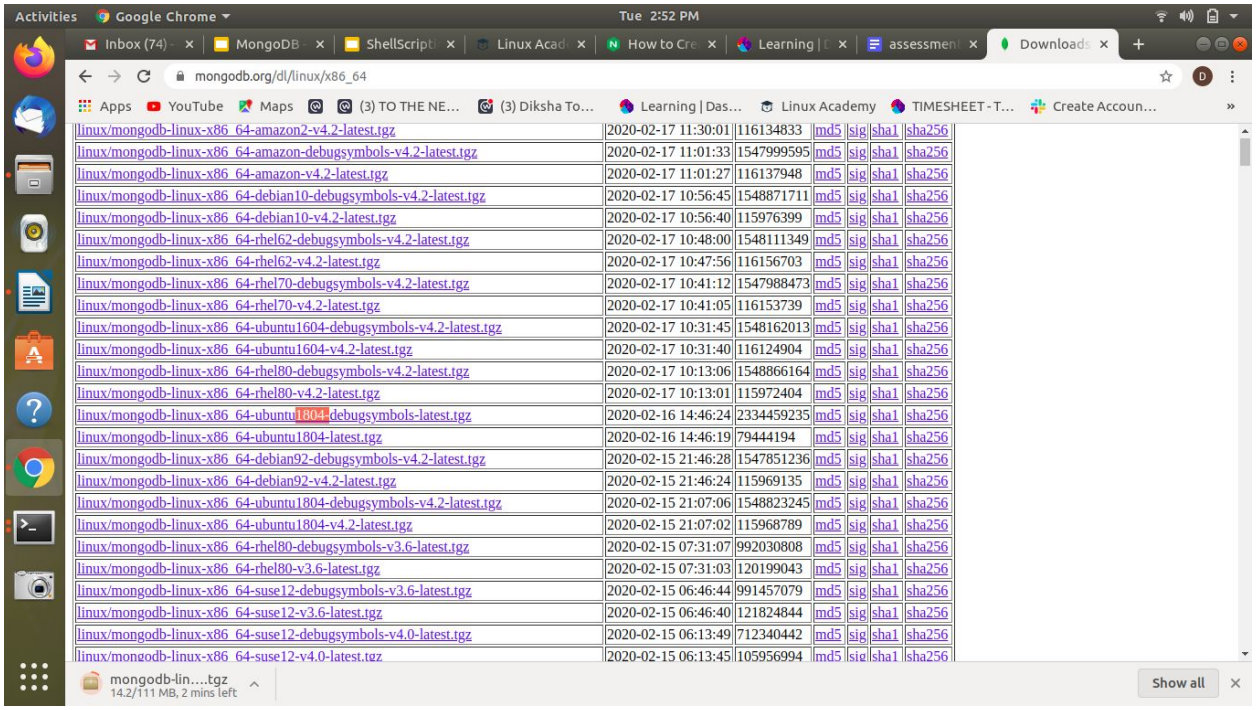
```
diksha@diksha:~$ mongod --auth --port 27017 --dbpath /data/db1
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] MongoDB starting : pid=29139 port=27017 dbpath=/data
/db1 64-bit host=diksha
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] db version v3.6.3
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] git version: 9586e557d54ef70f9ca4b43c26892cd55257e1a
5
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1  11 Sep 2018
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] allocator: tcmalloc
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] modules: none
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] build environment:
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten]     distarch: x86_64
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten]     target_arch: x86_64
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] options: { net: { port: 27017 }, security: { authori
zation: "enabled" }, storage: { dbPath: "/data/db1" } }
2020-02-17T16:44:02.464+0530 I STORAGE  [initandlisten] exception in initAndListen: NonExistentPath: Data di
rectory /data/db1 not found., terminating
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] now exiting
2020-02-17T16:44:02.464+0530 I CONTROL  [initandlisten] shutting down with code:100
```

Step 5: Connect and authenticate as the user administrator

```
diksha@diksha:~$ mongo  --port 27017 -u "root" -p "diksha" --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:
2020-02-16T12:28:39.491+0530 I STORAGE  [initandlisten]
2020-02-16T12:28:39.491+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly rec
ommended with the WiredTiger storage engine
2020-02-16T12:28:39.491+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodn
otes-filesystem
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten]
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the da
tabase.
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten] **          Read and write access to data and config
uration is unrestricted.
2020-02-16T12:28:39.956+0530 I CONTROL  [initandlisten]
>
```

```
> show databases ;
admin    0.000GB
config   0.000GB
local    0.000GB
student  0.000GB
> use student
switched to db student
> show collections
student
```

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



linux/mongodb-linux-x86_64-amazon2-v4.2-latest.tgz	2020-02-17 11:30:01	116134833	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-amazon-debugsymbols-v4.2-latest.tgz	2020-02-17 11:01:33	1547999595	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-amazon-v4.2-latest.tgz	2020-02-17 11:01:27	116137948	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-debian10-debugsymbols-v4.2-latest.tgz	2020-02-17 10:56:45	1548871711	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-debian10-v4.2-latest.tgz	2020-02-17 10:56:40	115976399	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel62-debugsymbols-v4.2-latest.tgz	2020-02-17 10:48:00	1548111349	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel62-v4.2-latest.tgz	2020-02-17 10:47:56	116156703	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel70-debugsymbols-v4.2-latest.tgz	2020-02-17 10:41:12	1547988473	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel70-v4.2-latest.tgz	2020-02-17 10:41:05	116153739	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-ubuntu1604-debugsymbols-v4.2-latest.tgz	2020-02-17 10:31:45	1548162013	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-ubuntu1604-v4.2-latest.tgz	2020-02-17 10:31:40	116124904	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel80-debugsymbols-v4.2-latest.tgz	2020-02-17 10:13:06	1548866164	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel80-v4.2-latest.tgz	2020-02-17 10:13:01	115972404	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-ubuntu1804-debugsymbols-latest.tgz	2020-02-16 14:46:24	2334459235	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-ubuntu1804-latest.tgz	2020-02-16 14:46:19	79444194	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-debian92-debugsymbols-v4.2-latest.tgz	2020-02-15 21:46:28	1547851236	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-debian92-v4.2-latest.tgz	2020-02-15 21:46:24	115969135	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-ubuntu1804-debugsymbols-v4.2-latest.tgz	2020-02-15 21:07:06	1548823245	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-ubuntu1804-v4.2-latest.tgz	2020-02-15 21:07:02	115968789	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel80-debugsymbols-v3.6-latest.tgz	2020-02-15 07:31:07	992030808	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-rhel80-v3.6-latest.tgz	2020-02-15 07:31:03	120199043	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-suse12-debugsymbols-v3.6-latest.tgz	2020-02-15 06:46:44	991457079	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-suse12-v3.6-latest.tgz	2020-02-15 06:46:40	121824844	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-suse12-debugsymbols-v4.0-latest.tgz	2020-02-15 06:13:49	712340442	md5	sig	sha1	sha256
linux/mongodb-linux-x86_64-suse12-v4.0-latest.tgz	2020-02-15 06:13:45	105956994	md5	sig	sha1	sha256

```
diksha@diksha:/opt$ sudo cp ~/Downloads/mongodb-linux-x86_64-ubuntu1804-v4.2-
latest.tgz .
[sudo] password for diksha:
diksha@diksha:/opt$ sudo tar -xvzf mongodb-linux-x86_64-ubuntu1804-v4.2-lates
t.tgz
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/THIRD-PARTY-NOTICES.gotools
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/README
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/THIRD-PARTY-NOTICES
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/MPL-2
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/LICENSE-Community.txt
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongodump
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongorestore
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongoexport
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongoimport
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongostat
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongotop
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/bsondump
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongofiles
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongoreplay
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongod
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongos
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mongo
mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/install_compass
diksha@diksha:/opt$
```

```
diksha@diksha:/opt$ mkdir -p ~/data/db
diksha@diksha:/opt$
diksha@diksha:/opt$ ls
google      mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe
lost+found  mongodb-linux-x86_64-ubuntu1804-v4.2-latest.tgz
diksha@diksha:/opt$ cd mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/
```

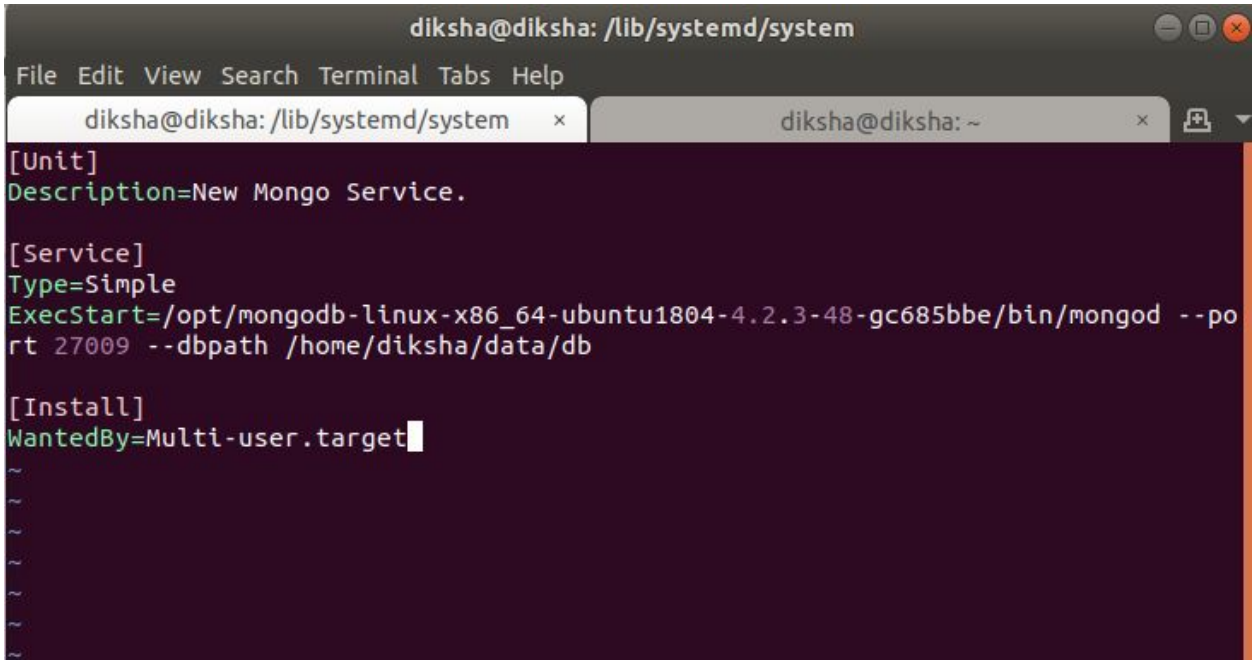
```
diksha@diksha:/opt/mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe$ cd bin/
diksha@diksha:/opt/mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin$ ./m
ongod --port 27009 --dbpath ~/data/db/
2020-02-18T15:09:30.771+0530 I  CONTROL  [main] Automatically disabling TLS 1
.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2020-02-18T15:09:30.775+0530 W  ASIO     [main] No TransportLayer configured
during NetworkInterface startup
2020-02-18T15:09:30.776+0530 I  CONTROL  [initandlisten] MongoDB starting : p
id=27668 port=27009 dbpath=/home/diksha/data/db/ 64-bit host=diksha
```

```
diksha@diksha:~$ sudo netstat -nltp | grep mongo
[sudo] password for diksha:
tcp        0      0 127.0.0.1:27009         0.0.0.0:*               LISTEN      27896/./mongod
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      23407/mongod
diksha@diksha:~$
```

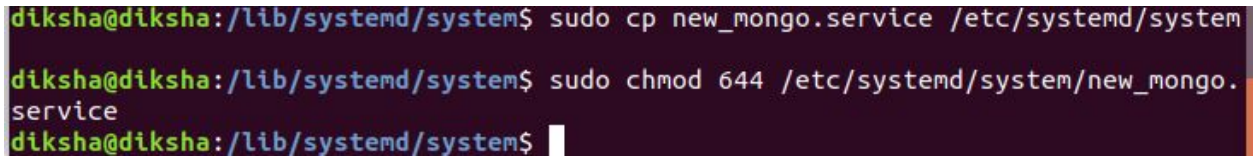
```
diksha@diksha:~$ sudo netstat -nltp | grep mongo
[sudo] password for diksha:
tcp        0      0 127.0.0.1:27009         0.0.0.0:*               LISTEN      27896/./mongod
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      23407/mongod
diksha@diksha:~$
diksha@diksha:~$ cd /opt/mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/
diksha@diksha:/opt/mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin$ ./mongo --port 27009
MongoDB shell version v4.2.3-48-gc685bbe
connecting to: mongodb://127.0.0.1:27009/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("019da100-33de-4e4f-8cf9-e169adeeaebe") }
MongoDB server version: 4.2.3-48-gc685bbe
Server has startup warnings:
2020-02-18T15:12:45.445+0530 I  STORAGE  [initandlisten]
2020-02-18T15:12:45.445+0530 I  STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongl
y recommended with the WiredTiger storage engine
2020-02-18T15:12:45.445+0530 I  STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/
prodnotes-filesystem
2020-02-18T15:12:47.080+0530 I  CONTROL  [initandlisten]
2020-02-18T15:12:47.081+0530 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for t
he database.
2020-02-18T15:12:47.081+0530 I  CONTROL  [initandlisten] **          Read and write access to data and c
onfiguration is unrestricted.
```

12. **Create init service of Mongo installed later***

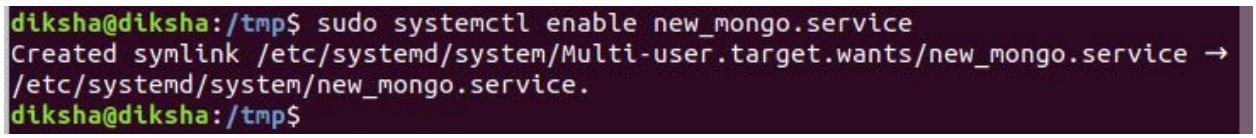
    A. Create unit file to define a systemd service at /lib/systemd/system/newmongo.service



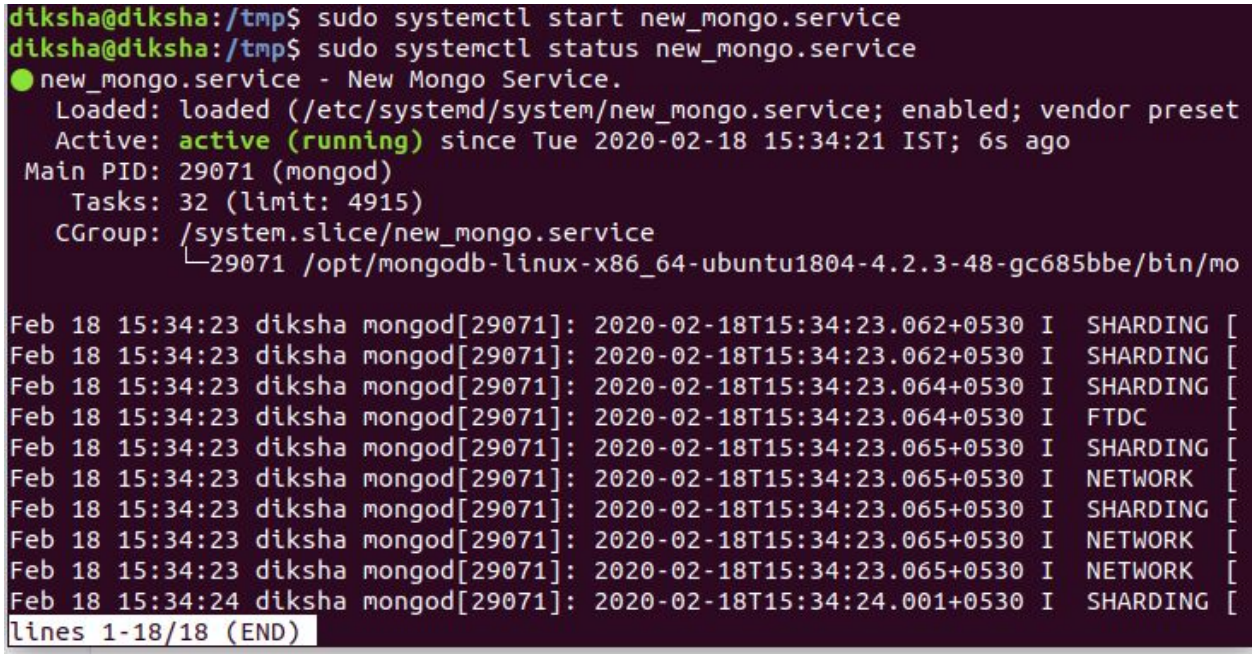
B. Copy new_mongo.service to /etc/systemd/system/ and set 644 permission to that file



C. Run sudo systemctl enable new_mongo.service to create symlin



D. Then start the service


diksha@diksha:/tmp$ sudo systemctl start new_mongo.service
diksha@diksha:/tmp$ sudo systemctl status new_mongo.service
new_mongo.service - New Mongo Service.
Loaded: loaded (/etc/systemd/system/new_mongo.service; enabled; vendor preset
Active: active (running) since Tue 2020-02-18 15:34:21 IST; 6s ago
Main PID: 29071 (mongod)
Tasks: 32 (limit: 4915)
CGroup: /system.slice/new_mongo.service
└─29071 /opt/mongodb-linux-x86_64-ubuntu1804-4.2.3-48-gc685bbe/bin/mo
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.062+0530 I SHARDING [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.062+0530 I SHARDING [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.064+0530 I SHARDING [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.064+0530 I FTDC [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.065+0530 I SHARDING [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.065+0530 I NETWORK [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.065+0530 I SHARDING [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.065+0530 I NETWORK [
Feb 18 15:34:23 diksha mongod[29071]: 2020-02-18T15:34:23.065+0530 I NETWORK [
Feb 18 15:34:24 diksha mongod[29071]: 2020-02-18T15:34:24.001+0530 I SHARDING [
lines 1-18/18 (END)

MySQL

1. **Install latest version of Mysql from apt-get repository**

```
diksha@diksha:~$ sudo apt install mysql-server
[sudo] password for diksha:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  galera-3 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libjemalloc1
  libmysqlclient20 libterm-readkey-perl mariadb-common socat
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  mysql-client-5.7 mysql-client-core-5.7 mysql-server-5.7
  mysql-server-core-5.7
Suggested packages:
  mailx tinyca
The following packages will be REMOVED:
  mariadb-client-10.1 mariadb-client-core-10.1 mariadb-server
  mariadb-server-10.1 mariadb-server-core-10.1
The following NEW packages will be installed:
  mysql-client-5.7 mysql-client-core-5.7 mysql-server mysql-server-5.7
  mysql-server-core-5.7
0 upgraded, 5 newly installed, 5 to remove and 3 not upgraded.
Need to get 19.0 MB of archives.
After this operation, 17.0 MB disk space will be freed.
Do you want to continue? [Y/n] y
```

2. **Create a database student**

```
mysql> CREATE DATABASE Student;
Query OK, 1 row affected (0.00 sec)

mysql> USE Student;
Database changed
```

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

```
Database changed
mysql> CREATE TABLE Student_data(NAME VARCHAR(20), CONTACT INT(10), CITY VARCHAR(20),
 ROLLNO INT(4), BRANCH VARCHAR(8) );
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql>
mysql>
mysql> INSERT INTO Student_data VALUES( "Diksha", 43535665,"Delhi",2,"CSE");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO Student_data VALUES( "Yash", 98535665,"Noida",21,"Cloud"),("Garima
",67475683,"Gurgao",45,"IT");
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> mysql> INSERT INTO Student_data VALUES( "Abhishek", 243535665,"Noida",43,"CSIT
vant",56464924,"Ddun",40,"IT");
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
```

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

```
mysql> SELECT * FROM Student_data WHERE CITY="Noida";
+----------+-----------+-------+--------+--------+
| NAME     | CONTACT   | CITY  | ROLLNO | BRANCH |
+----------+-----------+-------+--------+--------+
| Yash     |  98535665 | Noida |     21 | Cloud  |
| Abhishek | 243535665 | Noida |     43 | CSIT   |
+----------+-----------+-------+--------+--------+
2 rows in set (0.00 sec)

mysql>
```


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

```
mysql>
mysql>
mysql>
mysql> SELECT * FROM Student_data;
+----------+-----------+--------+--------+--------+
| NAME     | CONTACT   | CITY   | ROLLNO | BRANCH |
+----------+-----------+--------+--------+--------+
| Diksha   |  43535665 | Delhi  |      2 | CSE    |
| Yash     |  98535665 | Noida  |     21 | Cloud  |
| Garima   |  67475683 | Gurgao |     45 | IT     |
| Abhishek | 243535665 | Noida  |     43 | CSIT   |
| Revant   |  56464924 | Ddun   |     40 | IT     |
+----------+-----------+--------+--------+--------+
5 rows in set (0.00 sec)

mysql>
mysql>
mysql> UPDATE Student_data SET BRANCH="Civil" WHERE BRANCH="CS";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> UPDATE Student_data SET BRANCH="Civil" WHERE BRANCH="IT";
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT * FROM Student_data;
+----------+-----------+--------+--------+--------+
| NAME     | CONTACT   | CITY   | ROLLNO | BRANCH |
+----------+-----------+--------+--------+--------+
| Diksha   |  43535665 | Delhi  |      2 | CSE    |
| Yash     |  98535665 | Noida  |     21 | Cloud  |
| Garima   |  67475683 | Gurgao |     45 | Civil  |
| Abhishek | 243535665 | Noida  |     43 | CSIT   |
| Revant   |  56464924 | Ddun   |     40 | Civil  |
+----------+-----------+--------+--------+--------+
5 rows in set (0.00 sec)

mysql>
```

6. **Take dump of the database**

```
diksha@diksha:~$ sudo mysqldump -u root -pttn Student > Studentdump.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
diksha@diksha:~$ sudo vim Studentdump.sql
diksha@diksha:~$ ls -l Studentdump.sql
-rw-r--r-- 1 diksha diksha 2190 Feb 17 18:01 Studentdump.sql
diksha@diksha:~$
```

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

```
mysql> SELECT * FROM Student_data;
+----------+-----------+--------+--------+--------+
| NAME     | CONTACT   | CITY   | ROLLNO | BRANCH |
+----------+-----------+--------+--------+--------+
| Diksha   |  43535665 | Delhi  |      2 | CSE    |
| Yash     |  98535665 | Noida  |     21 | Cloud  |
| Garima   |  67475683 | Gurgao |     45 | Civil  |
| Abhishek | 243535665 | Noida  |     43 | CSIT   |
| Revant   |  56464924 | Ddun   |     40 | Civil  |
+----------+-----------+--------+--------+--------+
5 rows in set (0.00 sec)

mysql> DELETE FROM Student_data WHERE ROLLNO >3;
Query OK, 4 rows affected (0.01 sec)

mysql> SELECT * FROM Student_data;
+--------+----------+-------+--------+--------+
| NAME   | CONTACT  | CITY  | ROLLNO | BRANCH |
+--------+----------+-------+--------+--------+
| Diksha | 43535665 | Delhi |      2 | CSE    |
+--------+----------+-------+--------+--------+
1 row in set (0.00 sec)

mysql>
```

8. **Drop the database**

```
mysql> DROP DATABASE Student;
Query OK, 1 row affected (0.02 sec)

mysql>
```

9. **Restore the database again to have the full data**

```
mysql> create database Student;
Query OK, 1 row affected (0.00 sec)

mysql> use Student;
Database changed
mysql> source Studentdump.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from Student_data;
+----------+-----------+--------+--------+--------+
| NAME     | CONTACT   | CITY   | ROLLNO | BRANCH |
+----------+-----------+--------+--------+--------+
| Diksha   |  43535665 | Delhi  |      2 | CSE    |
| Yash     |  98535665 | Noida  |     21 | Cloud  |
| Garima   |  67475683 | Gurgao |     45 | Civil  |
| Abhishek | 243535665 | Noida  |     43 | CSIT   |
| Revant   |  56464924 | Ddun   |     40 | Civil  |
+----------+-----------+--------+--------+--------+
5 rows in set (0.00 sec)

mysql>
```

10. Enable authentication on the Mysql

```
mysql> CREATE DATABASE User1;
Query OK, 1 row affected (0.00 sec)

mysql> USE User1;
Database changed
mysql> CREATE TABLE TB( NAME VARCHAR(32));
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO TB VALUES("ABCD");
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM TB;
+------+
| NAME |
+------+
| ABCD |
+------+
1 row in set (0.00 sec)

mysql> CREATE USER 'test'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT, INSERT ON User1.* to 'test';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| Student            |
| User1              |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> USE User1;
Database changed
mysql> USE User1
Database changed
mysql> SHOW TABLES;
+-----------------+
| Tables_in_User1 |
+-----------------+
| TB              |
+-----------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT * FROM TB;
+------+
| NAME |
+------+
| ABCD |
+------+
1 row in set (0.00 sec)

mysql> INSERT INTO TB VALUES("LMN");
Query OK, 1 row affected (0.02 sec)

mysql> UPDATE TB SET NAME="JOE" WHERE  NAME="ABCD";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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

12. Create init service of Mysql installed later*