config files::

```
options: { config: "/etc/mongod.conf", net: { port: 27017 }, storage: {
dbPath: "/var/lib/mongodb" }, systemLog: { destination: "file", logAppend:
true, path: "/var/log/mongodb/mongod.log" } }
```

-->> Status of mongod process

- sudo service mongod status
- sudo service mongod start/stop

change localhost and port number if you installed in diff location..

/etc/mongod.conf

connect to mongo data bases using mongo shell::

- default it connect to test data base

```
rajesh@ubuntu-vm-05:/var/log/mongodb$ mongo
MongoDB shell version: 2.6.6
connecting to: test
```

- db command is used to see in which db currently is in use

```
> db
test
```

- show dbs command displays all the available db

```
> show dbs
admin  (empty)
local  0.078GB
mydb   0.078GB
test   0.078GB
```

- use command is used to switch the database schemas

```
> use mydb
switched to db mydb
> db
mydb
```

- to display the existing collections.

```
> show collections
emp
system.indexes
```

insert some sample data ::

```
db.emp.insert({fname : "pavan", lname : "naganna", emp_id : 2 })
db.emp.insert({fname : “aaa", lname : “ww", emp_id : 3 })
db.emp.insert({fname : “bbb", lname : “eeee", emp_id : 4 })
db.emp.insert({fname : “ccc", lname : "", emp_id : 5 })
db.emp.insert({fname : “dddd", lname : “yyy", emp_id : 6 })
db.emp.insert({fname : “eee", lname : “zzzz", emp_id : 6 })
db.emp.insert({fname : “fff", lname : “ppp", emp_id : 7 })
```

===================================

Read :-

read data from a collection with db.emp.find() command.

```
> db.emp.find();
{ "_id" : ObjectId("549d5d6a905deb248db1dae6"), "fname" : "rajesh", "lname" :
"dommati", "emp_id" : 1 }
{ "_id" : ObjectId("549d5d84905deb248db1dae7"), "fname" : "pavan", "lname" :
"naganna", "emp_id" : 2 }
{ "_id" : ObjectId("549d5f14905deb248db1dae9"), "fname" : "qwerty", "lname" :
"mongo", "emp_id" : 3 }
{ "_id" : ObjectId("549d6736905deb248db1daea"), "fname" : "qwerty", "lname" :
"mongo", "emp_id" : 3 }
{ "_id" : ObjectId("549d6752905deb248db1daeb"), "fname" : "qwerty", "lname" :
"mongo", "emp_id" : 3 }
{ "_id" : ObjectId("549e0b70d4c67f68b0a4598e"), "fname" : "fff", "lname" :
"asdhlaksdl", "emp_id" : 10 }
{ "_id" : ObjectId("549e0bdcd4c6d32dd312f6b6"), "fname" : "fff", "lname" :
"asdhlaksdl", "emp_id" : 10 }
{ "_id" : ObjectId("549e0c38d4c6a3612476b9a7"), "fname" : "fff", "lname" :
"asdhlaksdl", "emp_id" : 10 }
{ "_id" : ObjectId("549e0c84d4c6d9a81e412938"), "fname" : "fff", "lname" :
"asdhlaksdl", "emp_id" : 10 }
{ "_id" : ObjectId("54a0a532d4c655135f16623f"), "fname" : "fdsf", "lname" :
"mklopo", "emp_id" : 11 }
```

===================================

Write :-

```
switched to db zzzz
> db.testcollection.insert({user_id : 100})
WriteResult({ "nInserted" : 1 })
> db.testcollection.insert({user_id : 101, fn: "rajesh" })
WriteResult({ "nInserted" : 1 })
> db.testcollection.find()
{ "_id" : ObjectId("54a10c7b1d093e81881ac737"), "user_id" : 100 }
{ "_id" : ObjectId("54a10cad1d093e81881ac738"), "user_id" : 101, "fn" : "rajesh" }
> db.testcollection.insert({user_id : 102, fn: "aaaa" })
WriteResult({ "nInserted" : 1 })
> db.testcollection.find()
{ "_id" : ObjectId("54a10c7b1d093e81881ac737"), "user_id" : 100 }
{ "_id" : ObjectId("54a10cad1d093e81881ac738"), "user_id" : 101, "fn" : "rajesh" }
{ "_id" : ObjectId("54a10d081d093e81881ac739"), "user_id" : 102, "fn" : "aaaa" }
> db.testcollection.insert({user_id : 102, fn: "bbbb" })
WriteResult({ "nInserted" : 1 })
> db.testcollection.find()
{ "_id" : ObjectId("54a10c7b1d093e81881ac737"), "user_id" : 100 }
{ "_id" : ObjectId("54a10cad1d093e81881ac738"), "user_id" : 101, "fn" : "rajesh" }
{ "_id" : ObjectId("54a10d081d093e81881ac739"), "user_id" : 102, "fn" : "aaaa" }
{ "_id" : ObjectId("54a10d231d093e81881ac73a"), "user_id" : 102, "fn" : "bbbb" }
```

Upsert::

1. If update is for the existing record then it updates.

```
> db.testcollection.update({fn : "aaaa"}, {user_id :2000, fn : "updatedname"}, {upsert : true})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.testcollection.find()
{ "_id" : ObjectId("54a10c7b1d093e81881ac737"), "user_id" : 100 }
{ "_id" : ObjectId("54a10cad1d093e81881ac738"), "user_id" : 101, "fn" : "rajesh" }
{ "_id" : ObjectId("54a10d081d093e81881ac739"), "user_id" : 2000, "fn" : "updatedname" }
{ "_id" : ObjectId("54a10d231d093e81881ac73a"), "user_id" : 102, "fn" : "bbbb” }
```

2. If the update is for unknown record it inserts data into the collection.

```
> db.testcollection.update({fn : "michel"}, {user_id :200, fn : "upsertname"}, {upsert : true})
```

```
WriteResult({
"nMatched" : 0,
"nUpserted" : 1,
"nModified" : 0,
"_id" : ObjectId("54a112780611c902ee9c7f7f")
})
> db.testcollection.find()
{ "_id" : ObjectId("54a10c7b1d093e81881ac737"), "user_id" : 100 }
{ "_id" : ObjectId("54a10cad1d093e81881ac738"), "user_id" : 101, "fn" : "rajesh" }
{ "_id" : ObjectId("54a10d081d093e81881ac739"), "user_id" : 2000, "fn" : "updatedname" }
{ "_id" : ObjectId("54a10d231d093e81881ac73a"), "user_id" : 102, "fn" : "bbbb" }
{ "_id" : ObjectId("54a112780611c902ee9c7f7f"), "user_id" : 200, "fn" : "upsertname” }
```

Duplicate Key error::

If we insert try to insert a record with same _id value it throws duplicate error as below.

```
> db.testcollection.insert({"_id" : ObjectId("54a10c7b1d093e81881ac737") ,user_id : 100, fn:
"bbbb" })
WriteResult({
"nInserted" : 0,
"writeError" : {
"code" : 11000,
"errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error index:
zzzz.testcollection.$_id_  dup key: { : ObjectId('54a10c7b1d093e81881ac737') }"
}
})
```

Bulk write operations:

```
> var bulk = db.testcollection.initializeUnorderedBulkOp();
> bulk.({_id: 500, fname : “sdfdsff", lname :"sdadappp", emp_id : 232 });
2014-12-29T04:58:42.683-0600 SyntaxError: Unexpected token (
> bulk.insert({_id: 500, fname : “sdfdsff", lname :"sdadappp", emp_id : 232 });
2014-12-29T04:59:13.178-0600 SyntaxError: Unexpected token ILLEGAL
> bulk.insert({ _id: 500, fname : "sdfdsff", lname :"sdadappp", emp_id : 232 });
> bulk.insert({ _id: 501, fname : "sdfdsff", lname :"sdadappp", emp_id : 2332 });
> bulk.insert({ _id: 502, fname : "dnakldn", lname :"kjlkd", emp_id : 23 });
> bulk.execute(w : "majority", wtimeout : 5000);
2014-12-29T05:02:20.498-0600 SyntaxError: Unexpected token :
> bulk.execute({w : "majority", wtimeout : 5000});
BulkWriteResult({
"writeErrors" : [ ],
"writeConcernErrors" : [ ],
"nInserted" : 3,
"nUpserted" : 0,
"nMatched" : 0,
"nModified" : 0,
"nRemoved" : 0,
"upserted" : [ ]
})
```

Download java driver and connect to mango DB as below from Java program ::

Basic read and write operations:

```
MongoClient client = new MongoClient ("10.0.0.55", 27017);
DB db= client.getDB("mydb");
boolean auth= db.authenticate("rajesh", "Chicago2014".toCharArray()) ;
 // MongoClientURI uri  = new
MongoClientURI("mongodb://rajesh:Chicago2014@10.0.0.55:27017/mydb");
     // client = new MongoClient(uri);
     // DB db = client.getDB(uri.getDatabase());
System.out.println("connected to mydb data base -->> " + db.getName());

DBCollection emp = db.getCollection("emp");
System.out.println(" \n collection name -->> " + emp.getName());
System.out.println(" \n connectionpoint name -->> "
+client.getConnectPoint().toString() );
List<String> dbs =client.getDatabaseNames();
for(String databases : dbs){

System.out.println(" \n data base name -->> " + databases );

}

BasicDBObject nineties = new BasicDBObject();
nineties.put("fname", "fdsf");
nineties.put("lname", "mklopo");
nineties.put("emp_id", 11);
emp.insert(nineties);
System.out.println("\n --- inserted data into emp collection --- ");
```

```
BasicDBObject findquery = new BasicDBObject("emp_id", new
BasicDBObject("$gte",1));

DBCursor docs = emp.find(findquery);

System.out.println(" \n got the findquery results");


while(docs.hasNext()){

           DBObject doc = docs.next();

           System.out.println(

               doc.get("fname") + " | " + doc.get("lname") + " | " +
doc.get("emp_id") + " \n "

           );
```