# Ubuntu下MongoDB安装与使用教程

**一、MongoDB介绍**

MongoDB 是一个是一个基于分布式文件存储的数据库，介于关系数据库和非关系数据库之间，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似json的bson格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

**二、安装MongoDB（在线安装）**

MongoDB安装很简单，无需下载源文件，可以直接用apt-get命令进行安装。  
以下命令sudo apt-get install mongodb，可下载安装MongoDB，默认安装的版本是MongoDB 2.6.10，但目前MongoDB已经升级到3.2.8，这里将指导读者通过添加软件源的方式来安装3.2.8版本。  
首先打开终端，导入公共key到包管理器，输入以下命令：

1. sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv D68FA50FEA312927

创建MongoDB的文件列表

1. *#仅适用于Ubuntu14.04*
2. echo "deb http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list

 这是官方的源列表，官方MongoDB的软件源仓库可能对大陆用户不太友好， 建议使用阿里云的MongoDB软件源仓库， 执行以下命令：

1. echo"deb http://mirrors.aliyun.com/mongodb/apt/ubuntu xenial/mongodb-org/3.2 multiverse" | sudotee /etc/apt/sources.list.d/mongodb-org-3.2.list

更新包管理器，安装MongoDB,输入以下命令：

1. sudo apt-get update
2. sudo apt-get install -y mongodb-org

截图如下：  


注意：如果执行“sudo apt-get update”命令后出现如下错误：  
···  
update completed, but some metadata was ignored due to errors.  
E: 无法获得锁 /var/lib/dpkg/lock – open (11: 资源暂时不可用)  
E: 无法锁定管理目录(/var/lib/dpkg/)，是否有其他进程正占用它？  
···  
请按照如下方法解决错误，也就是输入以下三条命令：

1. sudo rm /var/cache/apt/archives/lock
2. sudo rm /var/lib/dpkg/lock
3. sudo apt-get update

运行上面三条命令以后，应该就可以顺利完成apt-get update了。

安装完成MongoDB以后，在终端输入以下命令查看MongoDB版本：

1. mongo -version

输出版本信息，表明安装成功，截图如下：  
  
启动和关闭mongodb命令如下：

1. sudo service mongodb start
2. sudo service mongodb stop

截图如下：  
启动关闭  
默认设置MongoDB是随Ubuntu启动自动启动的。  
输入以下命令查看是否启动成功：

1. pgrep mongo -l *#注意：-l是英文字母l，不是阿拉伯数字1*

截图如下：  
查看是否启动成功

 通过日志查看是否启动成功，cat /var/log/mongodb/mongod.log

[initandlisten] waiting for connections on port 27017

**出现安装错误的解决方案：**  
输入“sudo service mongodb start”启动mongodb的时候，如果报这个错误：Failed to start mongod.service: Unit not found  
请按照如下步骤解决该错误：  
（1）使用vim编辑器创建配置文件

1. sudo vim /etc/systemd/system/mongodb.service

（2）在该配置文件中添加如下内容：

[Unit]

Description=High-performance, schema-free document-oriented database

After=network.target

[Service]

User=mongodb

ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

[Install]

WantedBy=multi-user.target

保存退出vim编辑器。  
（3）输入如下命令启动mongodb：

1. sudo systemctl start mongodb
2. sudo systemctl status mongodb

这时就可以启动成功了。

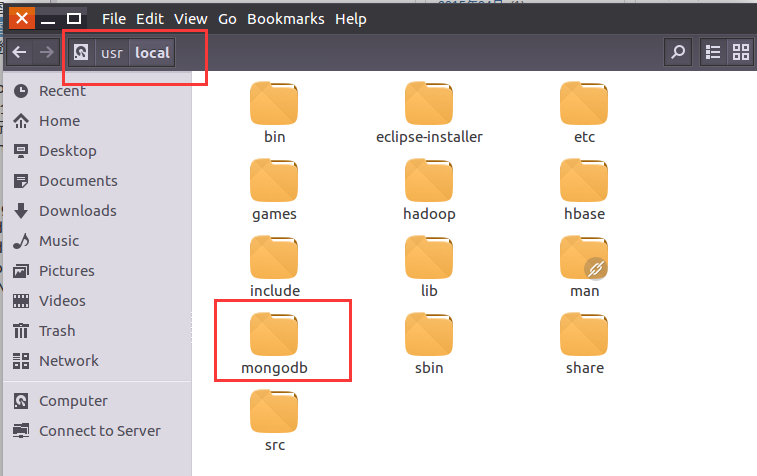
以后，每次启动和关闭MongoDB，就可以仍然使用如下命令：

1. sudo service mongodb start
2. sudo service mongodb stop

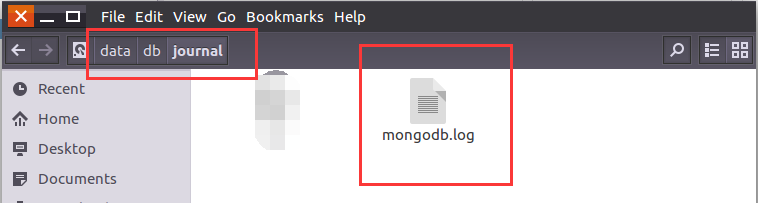
**二、安装MongoDB（离线安装）**

**1、将安装包解压到/usr/local，并将解压的文件夹重命名为mongodb**

1. sudo tar -zxf ~/Downloads/mongodb-linux-x86\_64-ubuntu1604-3.2.7.tgz -C /usr/local *#解压*
2. cd /usr/local
3. sudo mv mongodb-linux-x86\_64-ubuntu1604-3.2.7 mongodb *#更名为mongodb*
4. sudo chown -R dblab mongodb *#更改mongodb文件夹的属主。dblab是用户名*

文件目录结构截图如下：  
  
**2、创建文件夹/data/db/journal，创建文件mongodb.log**

1. sudo mkdir -p /data/db/journal *#创建文件夹*
2. sudo chown -R dblab /data *#更改/data文件夹的属主。dblab是用户名*
3. cd /data/db/journal
4. touch mongodb.log *#创建文件*

拟将/data/db作为数据保存目录，/data/db/journal/mongodb.log作为日志文件，截图如下：  
  
**3、创建并配置文件/data/mongodb.conf**

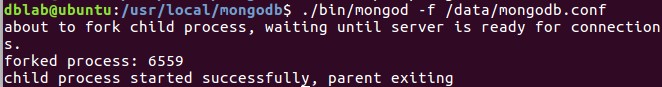
1. vim /data/mongodb.conf *#没有/data/mongodb.conf文件时用本条命令会自动创建/data/mongodb.conf*

在/data/mongodb.conf中写入如下配置信息：  
注意检查是否复制成功，用vim复制时有时会出现首行有几个字母丢失的问题。

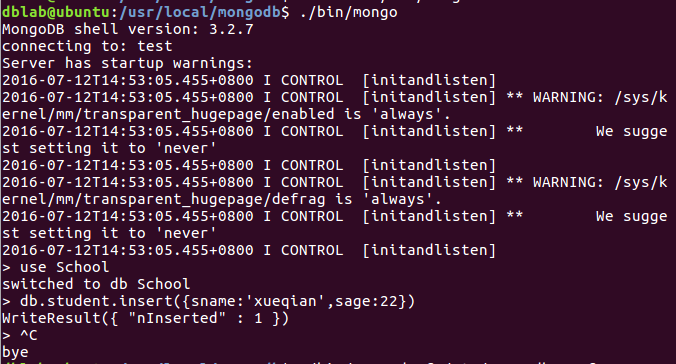
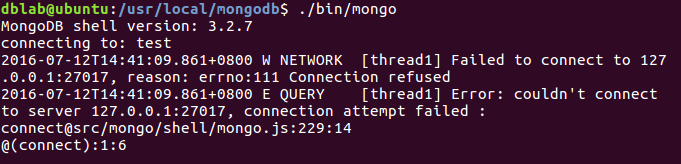
1. *#日志文件位置*
2. logpath=/data/db/journal/mongodb.log
4. *# 以追加方式写入日志*
5. logappend=true
7. *# 是否以守护进程方式运行*
8. fork=true
10. *# 默认27017*
11. *#port = 27017*
13. *# 数据库文件位置*
14. dbpath=/data/db
16. *# 启用定期记录CPU利用率和 I/O 等待*
17. *#cpu = true*
19. *# 是否以安全认证方式运行，默认是不认证的非安全方式*
20. *#noauth = true*
21. *#auth = true*
23. *# 详细记录输出*
24. *#verbose = true*
26. *# Inspect all client data for validity on receipt (useful for*
27. *# developing drivers)用于开发驱动程序时验证客户端请求*
28. *#objcheck = true*
30. *# Enable db quota management*
31. *# 启用数据库配额管理*
32. *#quota = true*
33. *# 设置oplog记录等级*
34. *# Set oplogging level where n is*
35. *# 0=off (default)*
36. *# 1=W*
37. *# 2=R*
38. *# 3=both*
39. *# 7=W+some reads*
40. *#diaglog=0*
42. *# Diagnostic/debugging option 动态调试项*
43. *#nocursors = true*
45. *# Ignore query hints 忽略查询提示*
46. *#nohints = true*
47. *# 禁用http界面，默认为localhost：28017*
48. *#nohttpinterface = true*
50. *# 关闭服务器端脚本，这将极大的限制功能*
51. *# Turns off server-side scripting. This will result in greatly limited*
52. *# functionality*
53. *#noscripting = true*
54. *# 关闭扫描表，任何查询将会是扫描失败*
55. *# Turns off table scans. Any query that would do a table scan fails.*
56. *#notablescan = true*
57. *# 关闭数据文件预分配*
58. *# Disable data file preallocation.*
59. *#noprealloc = true*
60. *# 为新数据库指定.ns文件的大小，单位:MB*
61. *# Specify .ns file size for new databases.*
62. *# nssize =*
64. *# Replication Options 复制选项*
65. *# in replicated mongo databases, specify the replica set name here*
66. *#replSet=setname*
67. *# maximum size in megabytes for replication operation log*
68. *#oplogSize=1024*
69. *# path to a key file storing authentication info for connections*
70. *# between replica set members*
71. *#指定存储身份验证信息的密钥文件的路径*
72. *#keyFile=/path/to/keyfile*

**4、脚本方式启动mongodb服务**

1. cd /usr/local/mongodb
2. ./bin/mongod -f /data/mongodb.conf

成功启动服务截图如下：  
  
输入pgrep mongo -l可以查看是否启动成功，截图如下：  
查看启动成功  
**5、进入mongodb shell模式**

1. cd /usr/local/mongodb
2. ./bin/mongo

截图如下：  
  
如果进入mongodb shell模式失败，输出如下截图信息，说明MongoDB服务未成功启动，请按步骤重新启动。  
  
**6、退出mongodb shell模式**  
输入exit或Ctrl+C退出shell模式  
**7、关闭mongodb服务**

1. use admin
2. db.shutdownServer()

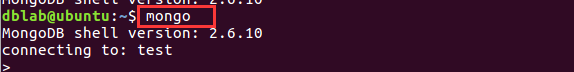
此时MongoDB服务就已经关闭了，然后输入exit或Ctrl+C退出shell模式。截图如下：  


**三、使用MongoDB**

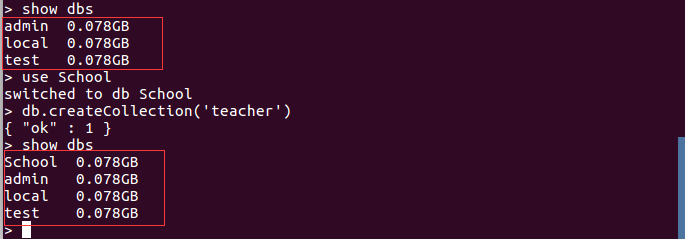
**shell命令模式**

输入如下命令进入MongoDB的shell命令模式：

1. mongo

默认连接的数据库是test数据库，在此之前一定要确保你已经启动了MongoDB，否则会出现错误，启动之后运行成功，如下截图：  


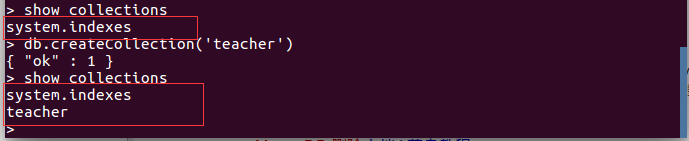
**常用操作命令：**

数据库相关  
show dbs:显示数据库列表  
show collections：显示当前数据库中的集合（类似关系数据库中的表table）  
show users：显示所有用户  
use yourDB：切换当前数据库至yourDB  
db.help() ：显示数据库操作命令  
db.yourCollection.help() ：显示集合操作命令，yourCollection是集合名  
MongoDB没有创建数据库的命令，如果你想创建一个“School”的数据库，先运行use School命令，之后做一些操作（如：创建聚集集合db.createCollection('teacher')）,这样就可以创建一个名叫“School”的数据库。截图如下：  
  
下面以一个School数据库为例，在School数据库中创建两个集合teacher和student，并对student集合中的数据进行增删改查基本操作（集合Collection相当于关系型数据库中的表table）。  
1、切换到School数据库

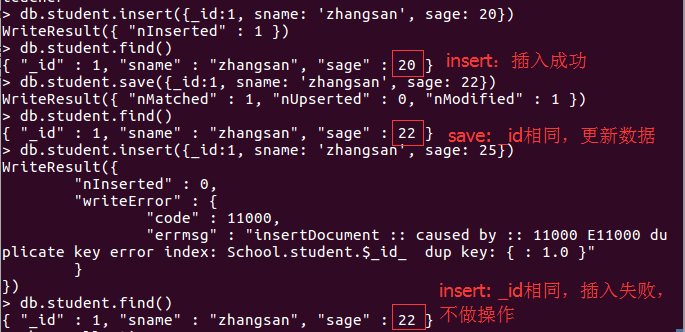
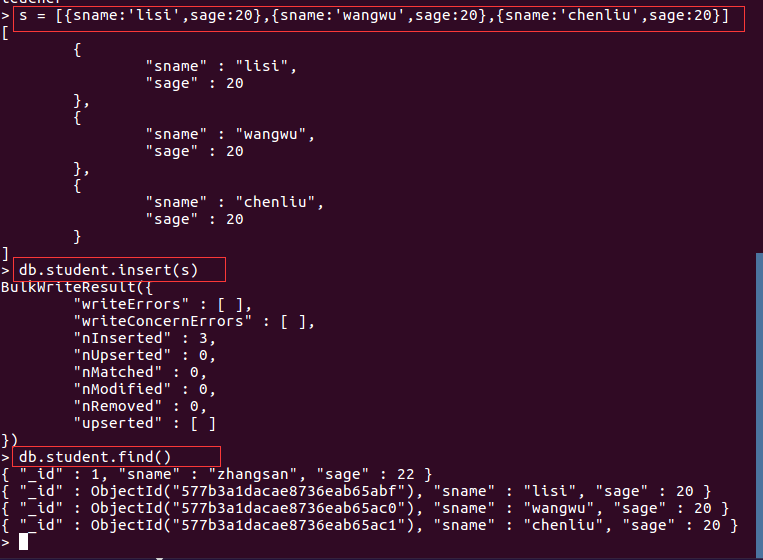
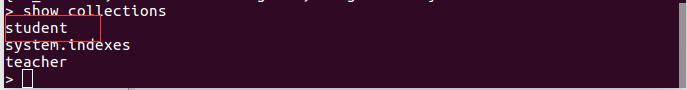
1. use School *#切换到School数据库。MongoDB 无需预创建School数据库，在使用时会自动创建*

2、创建Collection

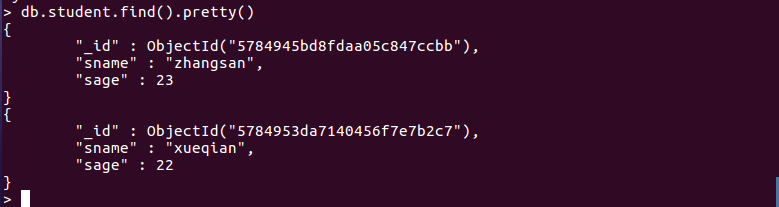
1. db.createCollection('teacher') *#创建一个聚集集合。MongoDB 其实在插入数据的时候，也会自动创建对应的集合，无需预定义集合*

截图如下：  
  
3、插入数据  
与数据库创建类似，插入数据时也会自动创建集合。  
插入数据有两种方式：insert和save。

1. db.student.insert({\_id:1, sname: 'zhangsan', sage: 20}) *#\_id可选*
2. db.student.save({\_id:1, sname: 'zhangsan', sage: 22}) *#\_id可选*

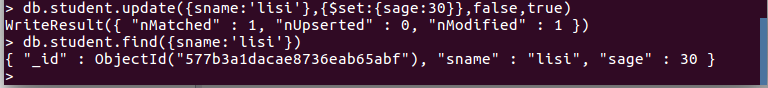
这两种方式，其插入的数据中\_id字段均可不写，会自动生成一个唯一的\_id来标识本条数据。而insert和save不同之处在于：在手动插入\_id字段时，如果\_id已经存在，insert不做操作，save做更新操作；如果不加\_id字段，两者作用相同都是插入数据。截图如下：  
  
添加的数据其结构是松散的，只要是bson格式均可，列属性均不固定，根据添加的数据为准。先定义数据再插入，就可以一次性插入多条数据，截图如下：  
  
运行完以上例子，student 已自动创建，这也说明 MongoDB 不需要预先定义 collection ，在第一次插入数据后，collection 会自动的创建。截图如下：  
  
3、查找数据  
db.youCollection.find(criteria, filterDisplay)  
criteria ：查询条件，可选  
filterDisplay：筛选显示部分数据，如显示指定列数据，可选（当选择时，第一个参数不可省略，若查询条件为空，可用{}做占位符，如下例第三句）

1. db.student.find() *#查询所有记录。相当于：select \* from student*
2. db.student.find({sname: 'lisi'}) *#查询sname='lisi'的记录。相当于： select \* from student where sname='lisi'*
3. db.student.find({},{sname:1, sage:1}) *#查询指定列sname、sage数据。相当于：select sname,sage from student。sname:1表示返回sname列，默认\_id字段也是返回的，可以添加\_id:0（意为不返回\_id）写成{sname: 1, sage: 1,\_id:0}，就不会返回默认的\_id字段了*
4. db.student.find({sname: 'zhangsan', sage: 22}) *#and 与条件查询。相当于：select \* from student where sname = 'zhangsan' and sage = 22*
5. db.student.find({$or: [{sage: 22}, {sage: 25}]}) *#or 条件查询。相当于：select \* from student where sage = 22 or sage = 25*

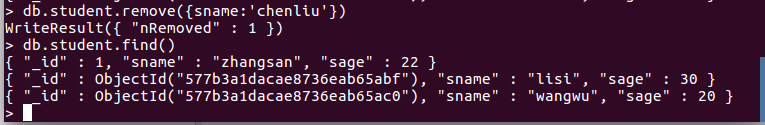
查询操作类似，这里只给出db.student.find({sname: 'lisi'})查询的截图，如下：  
查找  
db.youCollection.find(criteria, filterDisplay).pretty()表示格式化输出，截图如下：  


4、修改数据  
db.youCollection.update(criteria, objNew, upsert, multi )  
criteria: update的查询条件，类似sql update查询内where后面的  
objNew : update的对象和一些更新的操作符（如$set）等，也可以理解为sql update查询内set后面的。  
upsert : 如果不存在update的记录，是否插入objNew，true为插入，默认是false，不插入。  
multi: mongodb默认是false,只更新找到的第一条记录，如果这个参数为true,就把按条件查出来多条记录全部更新。默认false，只修改匹配到的第一条数据。  
其中criteria和objNew是必选参数，upsert和multi可选参数  
举例如下：

1. db.student.update({sname: 'lisi'}, {$set: {sage: 30}}, false, true) *#相当于：update student set sage =30 where sname = 'lisi';*

操作截图如下：  
  
5、删除数据

1. db.student.remove({sname: 'chenliu'}) *#相当于：delete from student where sname='chenliu'*

操作截图如下：  
  
6、删除集合

1. db.student.drop()

操作截图如下：  
  
7、退出shell命令模式  
输入exit或者Ctrl+C退出shell命令模式