

## 第 3 章

# MongoDB 数据库操作

### 学习目标

- 掌握 MongoDB 的部署
- 熟悉数据库和集合操作
- 掌握文档的插入、更新、删除以及查询操作
- 掌握使用 Java 操作 MongoDB
- 掌握使用 Python 操作 MongoDB
- 掌握使用 Robo 3T 操作 MongoDB

如果说理论知识是宝库,那么开启这个宝库的钥匙是实践操作。如果想要深入学习和掌握 MongoDB 数据库,除了学习 MongoDB 数据库的理论知识之外,还应掌握 MongoDB 数据库的实践操作。本章将针对 MongoDB 数据库操作的相关知识进行详细讲解。

## 3.1 MongoDB 部署

MongoDB 是一个开源、跨平台的数据库,它可以运行在 Windows 和 Linux 等多个平台上,为我们提供数据库服务。在不同的操作系统平台上,部署 MongoDB 也会有所不同。本节,我们将详细讲解 MongoDB 数据库基于 Windows 平台和 Linux 平台的部署。

### 3.1.1 基于 Windows 平台

MongoDB 提供了可用于 32 位系统和 64 位系统的预编译二进制安装包,其中,32 位的安装包不允许数据库文件(累积总和)超过 2GB,一般用于在 32 位的系统/平台上部署测试和开发,不可用于实际生产环境;而 64 位的安装包对数据库文件没有限制,因此受到了开发人员的青睐,因此,本书选择使用 64 位的 MongoDB 安装包(注意:选择 MongoDB 安装包之前,需要确认操作系统的位数,即 32 位或 64 位,其中在 32 位的系统上只能安装 32 位的 MongoDB,在 64 位系统上既可以安装 32 位,也可以安装 64 位的 MongoDB)。基于 Windows 平台的 MongoDB 部署的具体步骤如下。

#### 1. 下载 MongoDB 安装包

通过访问 MongoDB 官网 <https://www.mongodb.com/download-center/community> 进入 MongoDB 下载页面,如图 3-1 所示。

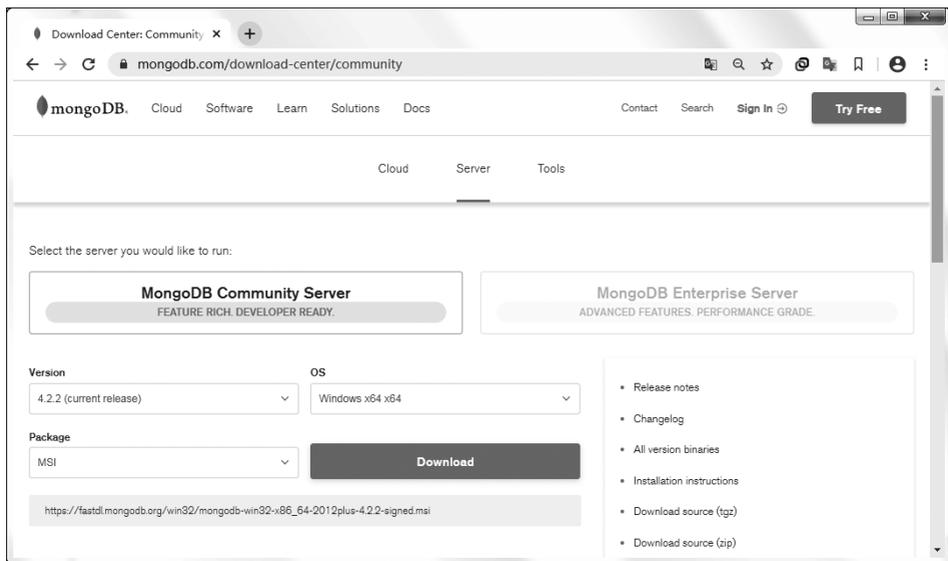


图 3-1 MongoDB 官网下载页面

在图 3-1 中,单击 Version 处的下拉框,选择需要安装的版本;单击 OS 处的下拉框选择要适配的系统或平台;单击 Package 处的下拉框,选择安装包的打包方式。关于 MongoDB 安装包版本、适配系统/平台以及打包方式的选择,具体介绍如下:

- Version 选择: MongoDB 的版本分为稳定版和开发版,其中,稳定版是经过充分测试的版本,具有稳定性和可靠性;而开发版是未得到充分测试的版本,不适合初学者使用。本书选择编写教材时的稳定版本 4.2.2。这里需要注意,稳定版和开发版的区别在于版本号(类似于 x.y.z),版本号中的第一位数字是主版本号;第二位数字是用于区分是稳定版还是开发版,若该数字为偶数,则说明该版本为稳定版,反之则为开发版;第三位数字为修订号。
- OS 选择: 由于我们是基于 Windows 平台,所以选择“Windows x64 x64”选项。
- Package 选择: 基于 Windows 平台的安装包打包方式有两种,分别是 MSI 和 ZIP,其中 MSI 安装包需要进行安装,而 ZIP 安装包只需要解压安装包,即可使用,因此这里选择 ZIP 安装包。

单击图 3-1 中的 Download 按钮,下载选择的 MongoDB 安装包。下载好 MongoDB 安装包,如图 3-2 所示。



图 3-2 下载好的 MongoDB 安装包

## 2. 解压 MongoDB 安装包

解压图 3-2 中的 MongoDB 安装包,解压完即可使用 MongoDB,读者也可以自行将文件夹重命名为 mongodb,然后再使用。解压后的 MongoDB,具体如图 3-3 所示。

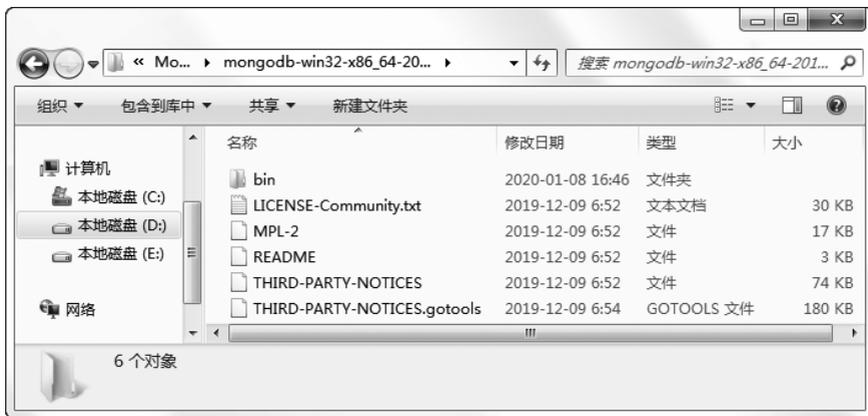


图 3-3 解压后的 MongoDB

从图 3-3 中可以看出,解压后的 MongoDB 包含一个 bin 文件夹,该文件夹中存放了很多 MongoDB 程序。下面,我们通过表 3-1 来介绍一下 MongoDB 程序。

表 3-1 MongoDB 程序

程 序	相 关 说 明
mongo.exe	用于启动 MongoDB Shell 客户端(Mongo Shell 命令行交互界面),在客户端里可执行相关命令对数据库进行增删改查等操作
mongod.exe	用于启动 MongoDB 服务
mongos.exe	用于启动 MongoDB 分片路由服务,可以处理来自应用层的查询操作并且识别所请求的数据位于分片群集的位置
bsondump.exe	用于将 bson 格式的文件转储为 json 格式的数据
mongodump.exe	用于备份 MongoDB 数据库中的数据
mongoexport.exe	用于导出 MongoDB 数据库中的数据
mongofiles.exe	用于管理 GridFS
mongoimport.exe	用于将数据导入 MongoDB 数据库中
mongorestore.exe	用于 MongoDB 的数据恢复
mongostat.exe	用于检测 MongoDB 数据库的状态
mongotop.exe	用于监控 MongoDB 数据库中数据的读写情况

## 3. 启动 MongoDB 服务

启动 MongoDB 服务共有两种不同的方式,即使用命令行参数的方式和使用配置文件

的方式,这两种启动方式的介绍如下。

(1) 使用命令行参数的方式启动 MongoDB 服务。

在使用命令行参数的方式启动 MongoDB 服务之前,需要在 MongoDB 的解压文件夹下创建一个文件夹,用于存放数据库文件和日志文件,因此本书创建了 data 文件夹,如图 3-4 所示。并在 data 文件夹下创建 db 和 log 子文件夹,其中 db 文件夹用于存储数据库文件,log 文件夹用于存储日志文件(便于在日志文件中查看 MongoDB 相关使用信息,不然关闭命令行窗口后,将无法再次查看 MongoDB 的日志),如图 3-5 所示。



图 3-4 父文件夹 data

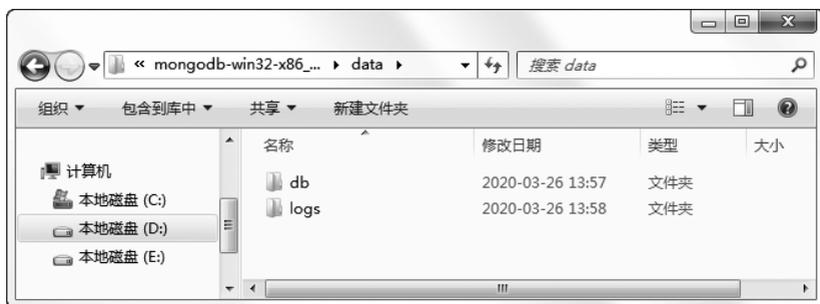


图 3-5 子文件夹 db 和 log

在 MongoDB 的 bin 文件夹下打开命令行窗口。进入 bin 文件夹,在目录栏中输入 cmd 提示符,如图 3-6 所示。

在图 3-6 中的目录栏处,按 Enter 键,在当前路径下打开命令行窗口,如图 3-7 所示。

在图 3-7 中,执行 `mongod --dbpath D:\MongoDB\mongodb-win32-x86_64-2012plus-4.2.2\data\db --logpath D:\MongoDB\mongodb-win32-x86_64-2012plus-4.2.2\data\logs\mongodb.log --logappend` 命令,启动 MongoDB 服务,命令行窗口的光标会一直闪动,效果如图 3-8 所示。然后查看日志文件 `mongodb.log`,若是日志文件中出现 MongoDB starting,则说明 MongoDB 服务启动成功,反之失败。日志文件 `mongodb.log` 的内容,如图 3-9 所示。

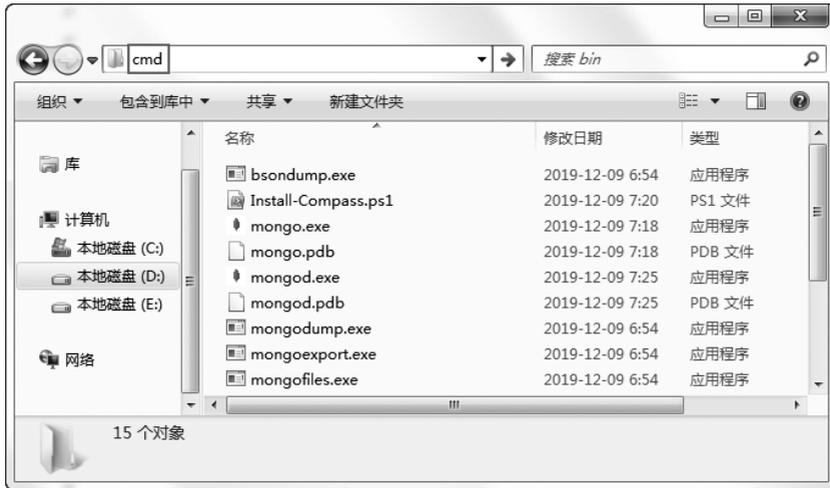


图 3-6 输入 cmd 提示符



图 3-7 命令行窗口



图 3-8 命令行窗口

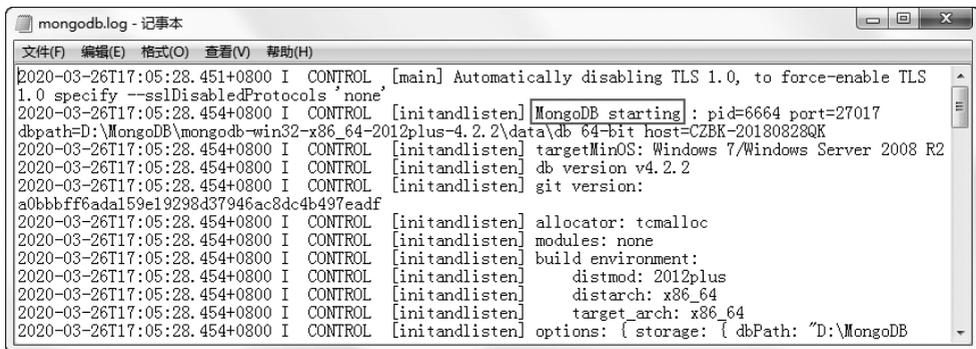


图 3-9 日志文件 mongodb.log 中的内容

从图 3-9 中可以看出,日志文件 mongodb.log 中出现了 MongoDB starting,因此说明我们成功启动了 MongoDB 服务。若是想要关闭 MongoDB 服务,只需要关闭命令行窗口即可。

(2) 使用配置文件的方式启动 MongoDB 服务。

在使用配置文件的方式启动 MongoDB 服务之前,需要在 MongoDB 的解压文件夹下创建一个文件夹 conf,用于存放 MongoDB 数据库的配置文件,并在该文件夹下创建一个文件 mongod.conf,用于指定数据库文件的存储路径及 MongoDB 的相关配置信息,文件 mongod.conf 的内容如下所示:

```
storage:
  dbPath: D:\MongoDB\mongodb-win32-x86_64-2012plus-4.2.2\data\db
systemLog:
  destination: file
  path: D:\MongoDB\mongodb-win32-x86_64-2012plus-4.2.2\logs\mongodb.log
  logAppend: true
```

在图 3-7 所示的命令行窗口中,执行相关命令,启动 MongoDB 服务,具体命令如下:

```
mongod -f ..\conf\mongod.conf
```

或

```
mongod --config ..\conf\mongod.conf
```

执行上述两条命令中的任意一条均可以启动 MongoDB 服务,这里以执行第一条命令进行演示,启动 MongoDB 服务,命令行窗口的光标会一直闪动,效果如图 3-10 所示。然后查看日志文件 mongodb.log,若是日志文件中出现 MongoDB starting,则说明 MongoDB 服务启动成功,反之失败,日志文件 mongodb.log 中的内容,如图 3-11 所示。



图 3-10 命令行窗口

从图 3-11 中可以看出,日志文件 mongodb.log 中出现了 MongoDB starting,因此说明我们成功启动了 MongoDB 服务。若想关闭 MongoDB 服务,只需要关闭命令行窗口即可。

### 3.1.2 基于 Linux 平台

基于 Linux 平台部署 MongoDB 之前,我们需要搭建 Linux 平台,关于 Linux 平台的搭建步骤,请参考本书提供的环境配置文档(注意:为了便于后续章节的操作使用,读者需要

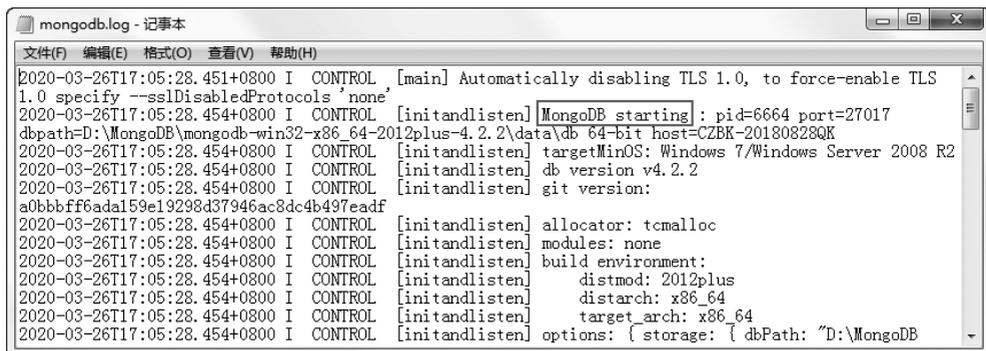


图 3-11 日志文件 mongod.log 中的内容

根据环境配置文档完成虚拟机 NoSQL\_1、NoSQL\_2、NoSQL\_3 的创建和配置以及系统目录结构的创建操作)。本章,我们将在虚拟机 NoSQL\_1(即 IP 地址为 192.168.121.134 的主机)中完成 MongoDB 数据库的部署和相关操作。

由于 root 用户拥有的权限很大,出于系统安全的考虑,需要新建一个普通用户操作 MongoDB 数据库,因此我们需要新建一个用户 user\_mongo。下面,我们详细介绍如何新建用户 user\_mongo。

### 1. 新建用户

打开 Linux 虚拟机并通过远程工具 Secure CRT 连接 Linux 平台,执行 useradd user\_mongo 命令,新建用户 user\_mongo;再执行 passwd user\_mongo 命令,初始化新用户 user\_mongo 的密码,具体如下:

```
[root@nosql01~]# useradd user_mongo
[root@nosql01~]# passwd user_mongo
Changing password for user user_mongo.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

从上述返回结果 successfully 可以看出,所有的身份验证令牌已经成功更新,即用户 user\_mongo 的密码初始化成功,这里设置的密码为 123456(New password 和 Retype new password 处均填密码 123456)。

### 2. 用户授权

首先执行 ls -l /etc/sudoers 命令,查看文件 sudoers 的操作权限,具体如下:

```
[root@nosql01 ~]# ls -l /etc/sudoers
-r--r----- .1 root root 4188 Jul  7  2015 /etc/sudoers
```

从上述返回结果可以看出,文件 sudoers 的操作权限为只读,不可进行编辑操作。因此

需要执行 `chmod -v u+w /etc/sudoers` 命令,将文件 `sudoers` 的权限修改为可编辑,然后执行查看 `ls -l /etc/sudoers` 命令,查看文件 `sudoers` 的权限是否变为可编辑,具体如下:

```
[root@nosql01 ~]# chmod -v u+w /etc/sudoers
mode of '/etc/sudoers' changed from 0440 (r--r-----) to 0640 (rw-r-----)
[root@nosql01 ~]# ls -l /etc/sudoers
-rw-r-----. 1 root root 4188 Jul  7 2015 /etc/sudoers
```

从上述返回结果可以看出,文件 `sudoers` 的操作权限为读写权限,说明我们已经成功将文件 `sudoers` 的操作权限修改为可编辑。需要注意,为了系统安全编辑完文件 `sudoers` 后,必须执行 `chmod -v u-w /etc/sudoers` 命令,将该文件的权限改为默认的只读权限。

执行 `vi /etc/sudoers` 命令,进入 `sudoers` 文件中,添加 `user_mongo ALL=(ALL) ALL` 内容,按 `ESC` 键,再执行 `:wq!` 命令,保存并退出 `sudoers` 文件。`sudoers` 文件添加的内容,具体如图 3-12 所示。

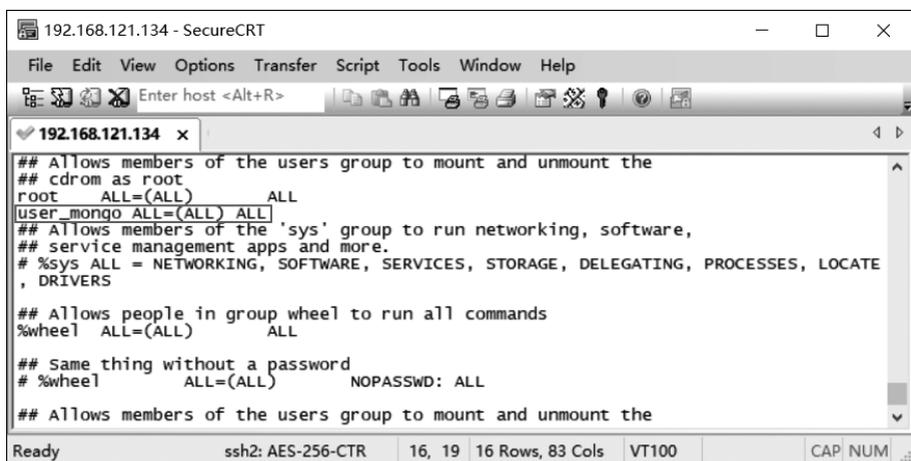


图 3-12 将 `user_mongo` 用户添加到 `sudoers` 文件中

执行 `su user_mongo` 命令,从 `root` 用户切换到 `user_mongo` 用户,效果如图 3-13 所示。

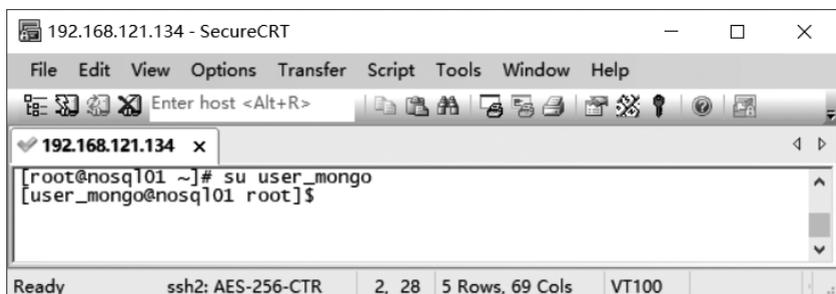


图 3-13 切换到用户 `user_mongo`

从图 3-13 中可以看出,我们成功切换到 `user_mongo` 用户。至此,我们完成了用户 `user_mongo` 的创建以及使用管理员命令的授权。

接下来,我们将详细讲解如何基于 Linux 平台部署 MongoDB,具体部署步骤如下。

## 1. 下载 MongoDB 安装包

通过访问 MongoDB 官网 <https://www.mongodb.com/download-center/community> 进入 MongoDB 下载页面,如图 3-14 所示。

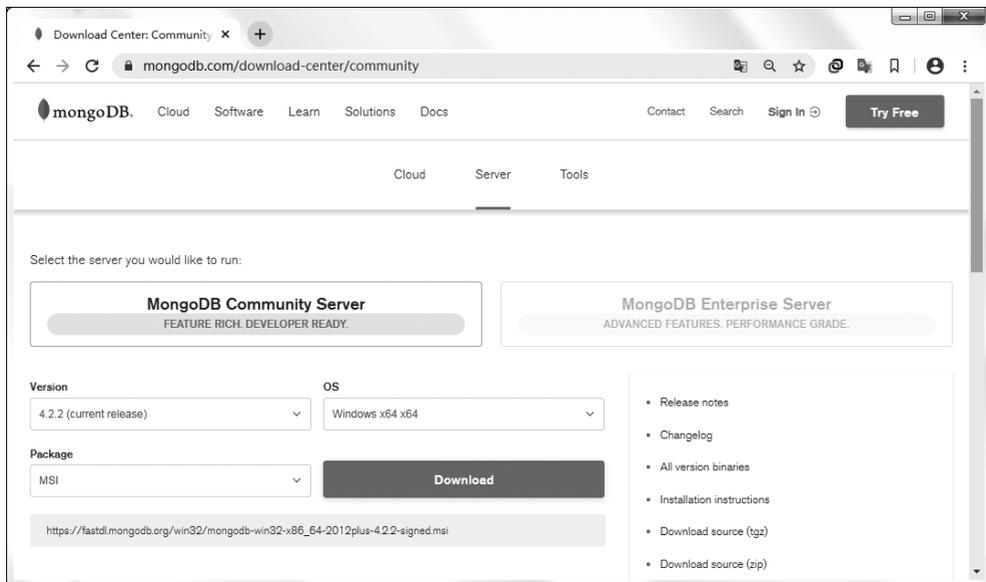


图 3-14 MongoDB 官网下载页面

在图 3-14 中,单击 Version 处的下拉框,选择需要安装的版本 4.2.2;单击 OS 处的下拉框选择要适配的系统或平台,由于本书的 Linux 系统是 CentOS 7 64 位,因此选择“RHEL 7.0 Linux 64-bit x64”选项;单击 Package 处的下拉框,选择安装包的打包方式,这里选择 TGZ 方式。单击 Download 按钮,下载所选择的 MongoDB 安装包。下载的 MongoDB 安装包如图 3-15 所示。

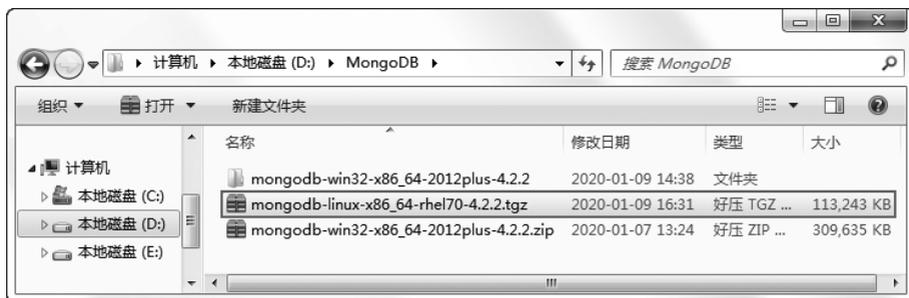


图 3-15 下载好的 MongoDB 安装包

## 2. 解压 MongoDB 安装包

下载 MongoDB 安装包后,使用 SecureCRT 工具将 MongoDB 安装包上传至 Linux 平

台的/opt/software目录下(需提前进入/opt/software目录下)。首先执行sudo rz上传文件命令(可以通过sudo yum install lrzsz -y指令安装lrzsz工具,实现rz上传和sz下载命令),弹出Select Files to Send using Zmodem对话框,然后选择要上传的MongoDB安装包,单击Add按钮,将其添加至Files to send文件框中,最后单击OK按钮,将MongoDB安装包上传至/opt/software目录下,如图3-16所示。

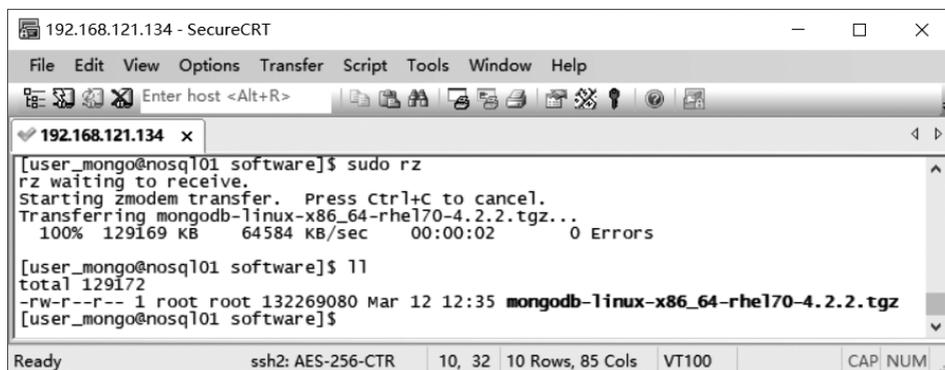


图 3-16 上传到 Linux 平台的 MongoDB 安装包

在图 3-16 中,首先将 MongoDB 安装包的用户和用户组权限修改为 user\_mongo;然后将/opt/servers/目录下mongodb\_demo目录的用户和用户组权限修改为 user\_mongo;最后解压 MongoDB 安装包至/opt/servers/mongodb\_demo目录,具体命令如下:

```
#修改 MongoDB 安装包的用户和用户组权限
$sudo chown user_mongo:user_mongo mongodb-linux-x86_64-rhel70-4.2.2.tgz
#修改 mongodb_demo 文件夹的用户和用户组权限
$sudo chown -R user_mongo:user_mongo /opt/servers/mongodb_demo/
#解压安装包
$tar -zxvf mongodb-linux-x86_64-rhel70-4.2.2.tgz -C /opt/servers/mongodb_demo/
```

执行上述命令,将 MongoDB 安装包进行解压,具体如图 3-17 所示。

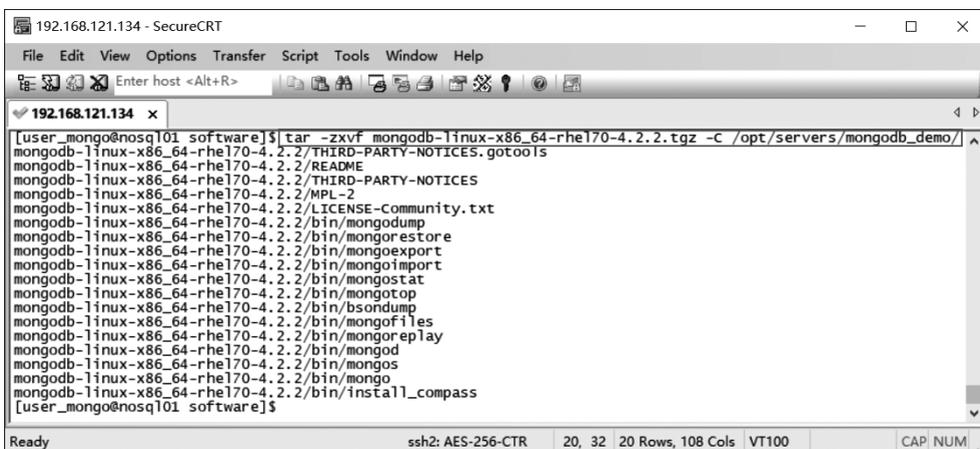


图 3-17 解压 MongoDB 安装包

在图 3-17 中,解压完 MongoDB 安装包后,进入到/opt/servers/mongodb\_demo 目录,如果觉得解压后的文件名过长,可以对文件进行重命名,具体命令如下:

```
#重命名为 mongodb
$mv mongodb-linux-x86_64-rhel70-4.2.2 mongodb
```

执行上述命令后,查看修改名称后的 MongoDB 安装目录,具体如图 3-18 所示。

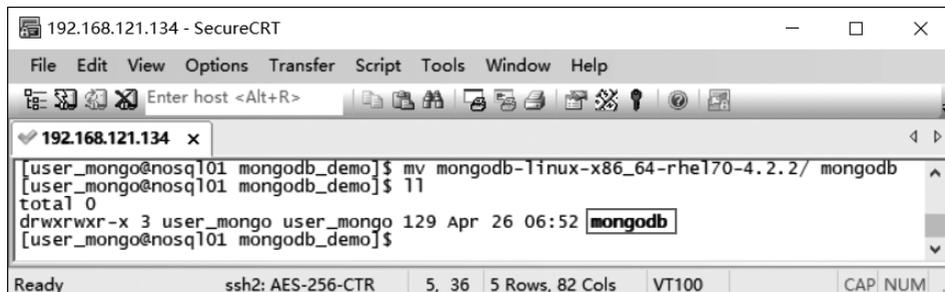


图 3-18 解压并重命名后的 MongoDB 文件夹

### 3. 配置 MongoDB

(1) 通常情况下,MongoDB 的数据文件存储在 data 目录的 db 目录下,日志文件存储在 logs 目录下,但是这两个目录在解压缩方式安装时不会自动创建。因此需要在 mongodb 目录下手动创建 data 目录和 logs 目录,并在 data 目录中创建 db 目录,在 logs 目录下创建一个 mongologs.log 日志文件,具体命令如下:

```
#创建数据文件存放目录
mkdir -p standalone/data/db/
#创建日志文件存放目录
mkdir standalone/logs/
#创建日志文件
touch standalone/logs/mongologs.log
```

执行上述命令后,在/mongodb/standalone 目录下出现了 data 目录和 logs 目录。进入 data 目录下,可以看到数据文件存放目录 db,如图 3-19 所示;进入 logs 目录下,可以看到日志文件 mongologs.log,具体如图 3-20 所示。

(2) 由于 MongoDB 的相关服务均存放在解压后/mongodb/bin 目录下,若是想要启动 MongoDB 服务,必须在 bin 目录下启动,因此为了避免启动 MongoDB 服务之前进入到 bin 目录下,我们需要配置用户环境变量,即执行 vi ~/.bash\_profile 命令打开并编辑.bash\_profile 文件,添加如下内容:

```
#配置用户环境变量
export PATH=/opt/servers/mongodb_demo/mongodb/bin:$PATH
```

添加上述内容后,执行:wq 命令,保存并关闭 bash\_profile 文件,然后执行 source ~/.bash\_

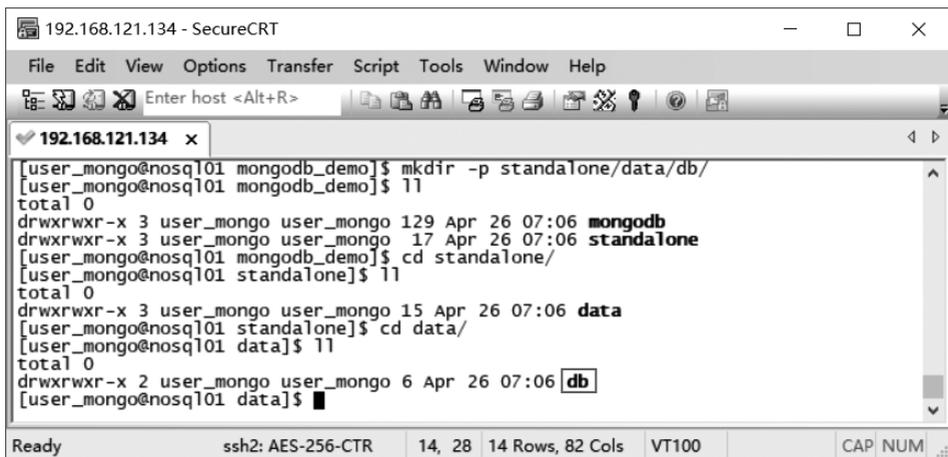


图 3-19 db 目录

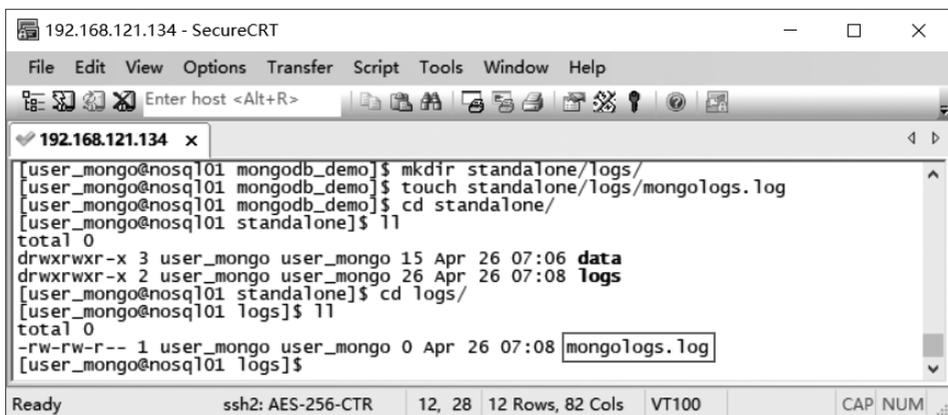


图 3-20 mongologs.log 日志文件

profile 命令,使得修改后的.bash\_profile 文件生效。需要注意的是每次切换到 user\_mongo 用户后,都需要执行 source ~/.bash\_profile 命令初始化用户环境变量。

#### 4. 启动 MongoDB 服务

启动 MongoDB 服务共有两种不同的方式,即使用命令行参数的方式和使用配置文件的方式,这两种启动方式的介绍如下:

(1) 使用命令行参数的方式启动 MongoDB 服务。

```
$mongod --dbpath=/opt/servers/mongodb_demo/standalone/data/db/
--logpath=/opt/servers/mongodb_demo/standalone/logs/mongologs.log
--logappend -fork
```

上述命令中,mongod 是 MongoDB 服务;--dbpath 参数是指定数据文件存放的位置;--logpath 参数是指定日志文件的存放位置;--logappend 参数指定使用追加的方式写日志;

--fork 参数指定以守护进程的方式(即后台)运行 MongoDB 服务。

执行上述命令,启动 MongoDB 服务,具体效果如图 3-21 所示。

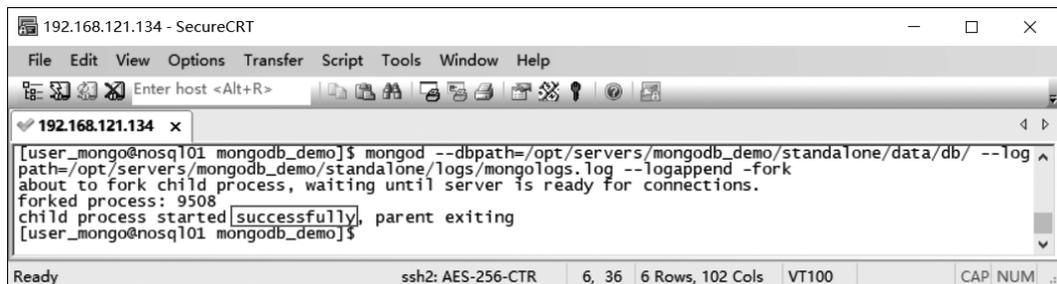


图 3-21 启动 MongoDB 服务

在图 3-21 中,出现了 successfully 单词,则说明我们成功启动了 MongoDB 服务。

(2) 使用配置文件的方式启动 MongoDB 服务。

在使用配置文件的方式启动 MongoDB 服务之前,需要在 mongodb 目录下创建一个 conf 目录,用于存放 MongoDB 数据库的配置文件,并在该目录下新建文件 mongod.conf,用于指定 MongoDB 服务启动所需要的一些参数。

创建 conf 目录,并在该目录下新建 mongod.conf 文件,具体命令如下:

```

#在 mongodb 目录下创建 conf 目录
$mkdir conf/
#在 conf 目录下新建 mongod.conf 文件
$touch mongod.conf

```

执行上述命令后,效果如图 3-22 所示。

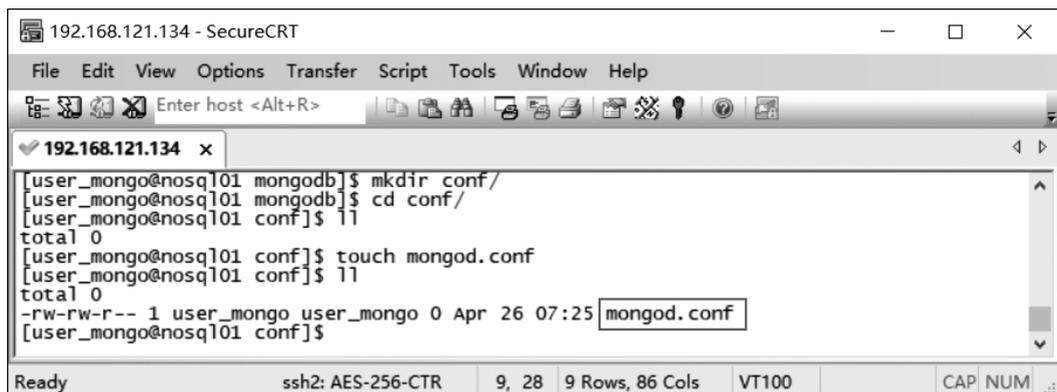


图 3-22 创建 conf 文件夹及 mongod.conf 文件

在图 3-22 中,即 mongodb 目录下执行 vi conf/mongod.conf 命令,打开并编辑 mongod.conf 文件,具体添加的内容如下:

```
systemLog:
  #MongoDB 发送所有日志输出的目标指定为文件
  #The Path of the log file to which mongos should send all diagnostic
  # logging information
  destination: file
  #mongod 发送所有诊断日志记录信息的日志文件的路径
  path: "/opt/servers/mongodb_demo/standalone/logs/mongologs.log"
  #当 mongod 重启时, mongod 会将新条目附加到现有日志文件的末尾
  logAppend: true
storage:
  #mongod 数据文件存储的目录
  dbPath: "/opt/servers/mongodb_demo/standalone/data/db/"
  journal:
    #启用或禁用持久性日志, 以确保数据文件保持有效和可恢复
    enabled: true
processManagement:
  #启用在后台运行 mongod 进程的守护进程模式
  fork: true
```

编辑上述内容需要注意的是 MongoDB 3.0 及以上版本的配置文件均采用 YAML 格式,其结构类似于大纲的缩排方式,开头使用空格作为缩进,数据结构为 Map 结构,即“Key: Value”,若是“:”之后有 Value,则后面必须增加一个空格;若是 Key 表示层级,则无须增加空格。按照层级结构,一级不需要空格缩进,二级缩进一或两个空格,三级缩进两个或四个空格,以此类推。

上述内容添加后,执行 `mongod -f /opt/servers/mongodb_demo/mongodb/conf/mongod.conf` 命令,启动 MongoDB 服务,具体效果如图 3-23 所示。

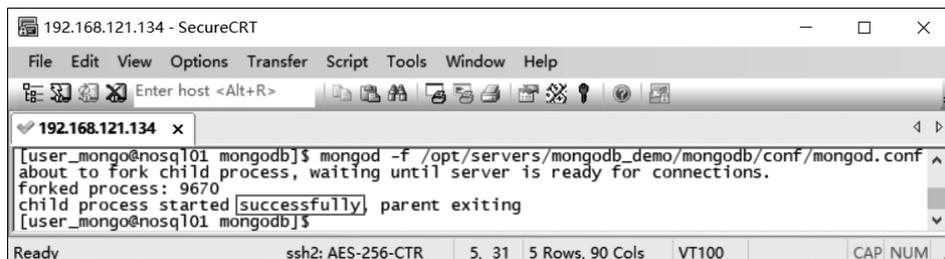


图 3-23 启动 MongoDB 服务

在图 3-23 中,出现了 `successfully` 单词,则说明我们成功启动 MongoDB 服务。若是想要关闭 MongoDB 服务,则先执行 `ps -ef | grep mongod` 命令,查看 MongoDB 的服务进程;然后执行 `kill -2 9670` 命令,结束 MongoDB 的服务进程(MongoDB 的服务进程每次都不同,因此若是想要结束该进程,则必须在命令中提供对应的 MongoDB 进程号);再执行 `ps -ef | grep mongod` 命令,查看 MongoDB 服务的进程是否存在,具体如图 3-24 所示。

从图 3-24 中可以看出,第二次执行 `ps -ef | grep mongod` 命令后,发现 MongoDB 的服务进程已经不存在了,因此说明我们成功关闭了 MongoDB 服务。

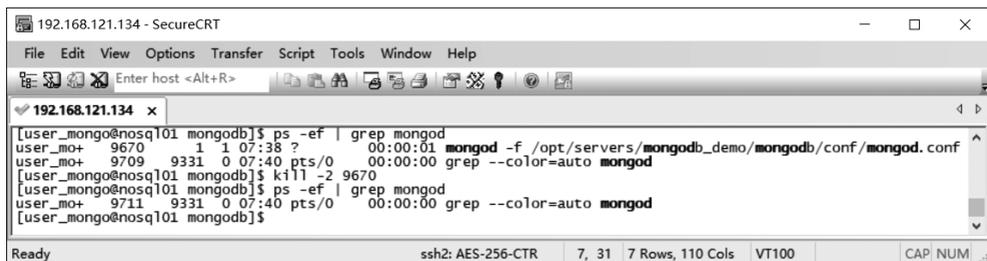


图 3-24 关闭 MongoDB 服务

## 3.2 数据库操作

MongoDB 提供了一个交互式 JavaScript 接口,即 mongo shell。mongo shell 主要用于操作 MongoDB,包括数据库、集合以及文档。本节将详细讲解如何使用 mongo shell 对 MongoDB 数据库进行新建、查看以及删除操作。

### 3.2.1 新建数据库

创建数据库,具体语法如下:

```
use DATABASE_NAME
```

上述语法中,use 是用于创建和切换数据库的命令,若指定的数据库不存在,则创建数据库,否则切换到指定数据库。DATABASE\_NAME 是新建数据库或切换数据库的名称。

下面,我们来创建一个数据库 articledb,用于存放文章的评论数据。首先,启动 MongoDB 服务,然后执行 mongo 命令,进入 mongo shell 中,效果如图 3-25 所示。

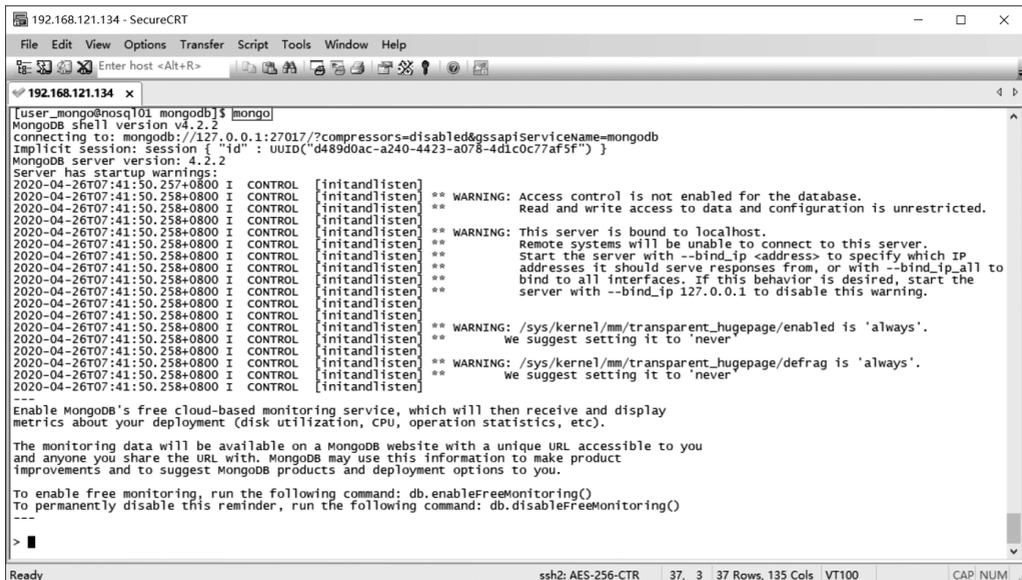


图 3-25 mongo shell 界面

在图 3-25 中,执行 `use articledb` 命令,创建数据库 `articledb`,具体如下:

```
>use articledb
switched to db articledb
```

从上述返回结果可以看出,数据库 `articledb` 已经创建完成。

### 3.2.2 查看数据库

查看数据库,分为查看所有数据库和查看当前数据库两种情况,具体语法如下:

```
#查看所有数据库
show dbs
#查看当前数据库
db
```

上述语法中,`show` 是用于查看所有数据库的名称和存储情况的命令。`dbs` 是 `databases` 的简称;`db` 表示当前数据库对象。

接下来,分别演示执行 `show dbs` 和 `db` 命令,查看所有数据库和当前数据库。首先执行 `show dbs` 命令,查看所有数据库,具体如下:

```
>show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
```

从上述返回结果可以看出,列出的数据库中没有出现数据库 `articledb`,这是由于使用 `use` 命令创建的数据库 `articledb` 存储在内存中,并且数据库中没有任何数据,因此 `show dbs` 命令是查看不到的,但是我们可通过执行 `db` 命令来查看当前数据库 `articledb`。需要注意的是,MongoDB 中默认包含数据库 `admin`、`config`、`local` 及 `test`,但是数据库 `test` 存储在内存中,也无任何数据,因此通过执行 `show dbs` 命令也是查看不到的。

执行 `db` 命令,查看当前的数据库,具体如下:

```
>db
articledb
```

从上述返回结果可以看出,当前数据库为 `articledb`。

### 3.2.3 删除数据库

删除数据库,具体语法如下:

```
db.dropDatabase()
```

上述语法中,`db` 表示当前数据库对象;`dropDatabase()` 是用于删除当前数据库的方法。执行 `db.dropDatabase()` 命令,删除当前数据库 `articledb`,在执行删除数据库命令前,先确保

已经切换到需要删除的数据库下,以免发生误删,具体如下:

```
>db
articledb
>db.dropDatabase()
{ "ok" : 1 }
```

从上述返回结果"ok"可以看出,数据库 articledb 已经删除完毕。需要注意的是在执行删除数据库命令前,先确保已经切换到需要删除的数据库下,以免发生误删。

## 3.3 集合操作

在前面章节中,我们使用 mongo shell 对数据库进行了基本的操作。本节将详细讲解如何使用 mongo shell 对集合进行创建和删除操作。

### 3.3.1 创建集合

创建集合有两种方式,即显式创建集合和隐式创建集合,具体语法如下:

```
# 显式创建集合
db.createCollection(COLLECTION_NAME, [OPTIONS])
# 隐式创建集合
db.COLLECTION_NAME.insert(DOCUMENT)
```

上述语法中,db 表示当前数据库对象(在创建集合前应确保处于对应数据库下);createCollection(COLLECTION\_NAME, [OPTIONS])是用于创建集合的方法,该方法中包含两个参数,参数 COLLECTION\_NAME 表示要创建的集合名称;参数 OPTIONS 表示一个文档,用于指定集合的配置,该参数为可选参数。Insert(DOCUMENT)是用于往集合中插入文档的方法,该方法包含一个参数 DOCUMENT,该参数表示文档。

下面,通过执行 db.createCollection("myCollection")命令,演示如何显式创建集合 myCollection,并执行 show collections 命令,查看是否成功创建集合,具体如下:

```
>db.createCollection("myCollection")
{ "ok" : 1 }
>show collections
myCollection
```

从上述返回结果可以看出,集合 myCollection 创建成功了。关于隐式创建集合,这里先不作介绍,在 3.4 节中将进行详细介绍。

### 3.3.2 删除集合

删除集合的具体语法如下:

```
db.COLLECTION_NAME.drop()
```

上述语法中,db 表示当前数据库对象;COLLECTION\_NAME 表示当前集合对象;drop()是用于删除集合的方法。

下面,通过执行 db.myCollection.drop()命令演示如何删除集合 myCollection,具体如下:

```
>db.myCollection.drop()
true
```

从上述返回结果 true 可以看出,集合 myCollection 已被成功删除。

## 3.4 文档的插入、更新与删除操作

在前面节中,我们使用 mongo shell 对数据库和集合进行了基本的操作。本节,我们将详细讲解如何使用 mongo shell 对文档进行插入、更新以及删除操作。

### 3.4.1 文档插入

文档插入可以分为单文档插入和多文档插入,具体语法如下:

```
#单文档插入
db.COLLECTION_NAME.insert(document)
```

或

```
db.COLLECTION_NAME.save(document)
#多文档插入
db.COLLECTION_NAME.insertMany([document1,document2,...])
```

上述语法中,对象、方法以及参数的详细介绍如下:

- db 表示当前数据库对象。
- COLLECTION\_NAME 表示当前集合对象。
- insert()和 save()是用于插入文档的方法,这两个方法均包含参数 document,该参数表示插入一个文档至集合中。insert()和 save()方法的区别在于,若使用 insert()方法插入文档,且集合中已存在该文档,则会报 E11000 duplicate key error collection 错误,反之则写入;若使用 save()方法插入文档,且集合中已存在该文档,则会更新它,反之则写入。
- insertMany()是用于插入多个文档的方法,该方法包含参数 [document1, document2,...],该参数是有多个文档组成的数组。

下面,演示执行插入单文档命令,先创建数据库 articledb,再隐式创建集合 comment,并在该集合中插入一个文档,具体命令如下:

```
>use articledb
switched to db articledb
```

```
>db.comment.insert(
  {"articleid":"100000","content":"今天天气真好,阳光明媚","userid":"1001",
   "nickname":"Rose","age":"20","phone":"18807141995","createdatetime":new Date(),
   "likenum":NumberInt(10),"state":null}
)
WriteResult({ "nInserted" : 1 })
```

从上述返回结果 WriteResult({ "nInserted" : 1 })可以看出,我们成功将一个文档插入集合 comment 中。需要注意的是,MongoDB 中的数字默认是 Double 类型的,若要存储整型,则需要使用函数“NumberInt(整型数字)”,否则查询就会出现错误;若集合中插入的文档没有指定\_id,则会自动生成主键值 ObjectId,ObjectId 是使用 12 个字节的存储空间,由 24 个十六进制字符组成的字符串(每个字节可以存储两个十六进制字符),若某个键没有值,则可以赋值为 null 或者不写该键。执行 db.comment.find()命令,查看集合 comment 中的内容,具体如下:

```
>db.comment.find()
{ "_id" : ObjectId("5e175913ddb10619b13a001f"), "articleid" : "100000",
  "content" : "今天天气真好,阳光明媚", "userid" : "1001", "nickname" : "Rose",
  "age":"20","phone":"18807141995","createdatetime":ISODate("2020-01-09T16:47:15.897Z"),
  "likenum" : 10, "state" : null }
```

从上述返回结果可以看出,集合 comment 中包含一个文档,该文档即为我们插入的文档。

接着,演示执行插入多文档命令,这里是往集合 comment 中插入 6 个文档,具体如下:

```
>db.comment.insertMany([
  {"_id":"1","articleid":"100001","content":"清晨,我们不该把时间浪费在手机上,健康很重要,
   喝一杯温水,幸福你我他。","userid":"1002","nickname":"相忘于江湖","age":"25",
   "phone":{"homePhone":"82174911","mobilePhone":"13065840128"},
   "createdatetime":new Date("2020-01-02 09:08:15"),"likenum":NumberInt(1000),"state":"1"},
  {"_id":"2","articleid":"100001","content":"我夏天空腹喝凉开水,冬天喝温开水",
   "userid":"1003","nickname":"伊人憔悴","age":"22","phone":"13442031624",
   "createdatetime":new Date("2020-01-02 10:20:40"),"likenum":NumberInt(888),"state":"1"},
  {"_id":"3","articleid":"100001","content":"夏天和冬天,我都喝凉开水","userid":"1004",
   "nickname":"杰克船长","age":"28","phone":"13937163334","createdatetime":
   new Date("2020-01-02 14:56:09"),"likenum":NumberInt(666),"state":null},
  {"_id":"4","articleid":"100001","content":"专家说不能空腹喝冰水,影响健康","userid":"1005",
   "nickname":"罗密欧","age":"18","phone":"15813134403","createdatetime":
   new Date("2020-01-03 11:26:29"),"likenum":NumberInt(2000),"state":"1"},
  {"_id":"5","articleid":"100001","content":"研究表明,刚烧开的水千万不要喝,因为烫嘴",
   "userid":"1005","nickname":"罗密欧","age":"18","phone":"15813134403",
   "createdatetime":new Date("2020-01-03 15:10:37"),"likenum":NumberInt(3000),"state":"1"},
  {"_id":"6","articleid":"100001","content":"喝水是生命体通过口腔摄入水分的方式,
   人体每天通过口腔摄入的液体大约有 2 升","userid":"1006","nickname":"爱德华","age":"30",
   "phone":{"homePhone":"62771541","mobilePhone":"13262984142"},
   "createdatetime":new Date("2020-01-03 15:10:37"),"likenum":NumberInt(3000),"state":"1"}
])
```

```
{
  "acknowledged" : true,
  "insertedIds" : [
    "1",
    "2",
    "3",
    "4",
    "5",
    "6"
  ]
}
```

从上述返回结果“`"acknowledged" : true`”可以看出,我们成功将 6 个文档插入集合 `comment` 中,`insertedIds` 表示插入多个文档的对应 `_id`。执行 `db.comment.find()` 命令,查看集合 `comment` 中的内容,具体如下:

```
>db.comment.find()
{ "_id" : ObjectId("5e162cca238207d6b5e1f585"), "articleid" : "100000",
  "content":"今天天气真好,阳光明媚","userid":"1001","nickname":"Rose","age":"20",
  "phone":"18807141995","createdatetime":ISODate("2020-01-08T19:26:02.544Z"),
  "likenum":10,"state":null}
{ "_id" : "1", "articleid": "100001", "content" : "清晨,我们不该把时间浪费在手机上,
健康很重要,喝一杯温水,幸福你我他。", "userid": "1002", "nickname": "相忘于江湖",
"age": "25", "phone": { "homePhone": "82174911", "mobilePhone": "13065840128" },
"createdatetime": ISODate("2020-01-02T01:08:15Z"), "likenum": 1000, "state": "1" }
{ "_id" : "2", "articleid": "100001", "content": "我夏天空腹喝凉开水,冬天喝温开水",
"userid": "1003", "nickname": "伊人憔悴", "age": "22", "phone": "13442031624",
"createdatetime": ISODate("2020-01-02T02:20:40Z"), "likenum": 888, "state": "1" }
{ "_id" : "3", "articleid": "100001", "content": "夏天和冬天,我都喝凉开水", "userid": "1004",
"nickname": "杰克船长", "age": "28", "phone": "13937163334", "createdatetime":
ISODate("2020-01-02T06:56:09Z"), "likenum": 666, "state": null }
{ "_id" : "4", "articleid": "100001", "content": "专家说不能空腹喝冰水,影响健康", "userid" : "1005",
"nickname": "罗密欧", "age": "18", "phone": "15813134403", "createdatetime":
ISODate("2020-01-03T03:26:29Z"), "likenum": 2000, "state" : "1" }
{ "_id" : "5", "articleid": "100001", "content": "研究表明,刚烧开的水千万不要喝,因为烫嘴",
"userid": "1005", "nickname": "罗密欧", "age": "18", "phone": "15813134403",
"createdatetime": ISODate("2020-01-03T07:10:37Z"), "likenum" : 3000, "state" : "1" }
{ "_id" : "6", "articleid": "100001", "content": "喝水是生命体通过口腔摄入水分的方式,
人体每天通过口腔摄入的液体大约有 2 升", "userid": "1006", "nickname": "爱德华",
"age": "30", "phone": { "homePhone": "62771541", "mobilePhone": "13262984142" },
"createdatetime": ISODate("2020-01-03T07:10:37Z"), "likenum": 3000, "state": "1" }
```

从上述返回结果可以看出,集合 `comment` 中存在 7 个文档,包含第一次插入的单个文档和第二次插入的 6 个文档。

### 3.4.2 文档更新

更新文档的具体语法如下:

```
db.COLLECTION_NAME.update(criteria,objNew,upsert, multi)
```