

Docker 应用部署-MongoDB

可以通过官网连接: <https://www.mongodb.com/try/download/community> 获取你需要的稳定版本。

Give it a try with a free, highly-available 512 MB cluster.

Version

6.0.2 (current) 这里是版本号

▼

Platform

RedHat / CentOS 8.0

▼

Package

server

▼

Download ⬇

More Options ...

命令参考连接:

- <https://docs.mongodb.com/manual/>
- <https://docs.mongodb.com/mongodb-shell/>
- <https://jockchou.gitbooks.io/getting-started-with-mongodb/content/index.html>

1 准备目录

在home目录下面创建 mongodb 目录

```
1 mkdir /home/mongodb
2 cd /home/mongodb
```

2 开放端口

注意: 如果是云服务不需要使用下面命令开放端口, 需要在云控制台去做入方向规则设置。

```
1 #查看是否已经开放27017端口
2 firewall-cmd --list-port
3 #没有开放使用下面命令开放
4 firewall-cmd --add-port 27017-27019/tcp --permanent
5 #重新加载防火墙
6 firewall-cmd --reload
```

3 服务编排

3.1 编写初始化脚本

创建一个初始化脚本，命名为 `mongo-init.sh`。

执行命令行 `vi mongo-init.sh`，并写入以下内容。

```
1  #6.x以前的版本
2  #mongo -- "$MONGO_INITDB_DATABASE" <<EOF
3  #6.x版本
4  mongosh -- "$MONGO_INITDB_DATABASE" <<EOF
5  db = db.getSiblingDB('admin')
6  db.auth('$MONGO_INITDB_ROOT_USERNAME', '$MONGO_INITDB_ROOT_PASSWORD')
7  db = db.getSiblingDB('$MONGO_INITDB_DATABASE')
8  db.createUser({
9      user: "$MONGO_USERNAME",
10     pwd: "$MONGO_PASSWORD",
11     roles: [
12         { role: 'readWrite', db: '$MONGO_INITDB_DATABASE' }
13     ]
14 })
15 EOF
```

注意：6.x后去掉了mongo指令改为mongosh指令，所以你在安装的时候一定要使用正确的指令。

当然你可以自己使用资源目录中文件，上传到 `mongodb` 目录，但是要注意文件编码格式为 `unix`，不然无法执行。

用户权限角色说明，可以参考下表。

规则	说明
root	只在admin数据库中可用。超级账号，超级权限
read	允许用户读取指定数据库
readWrite	允许用户读写指定数据库
dbAdmin	允许用户在指定数据库中执行管理函数，如索引创建、删除，查看统计或访问system.profile
userAdmin	允许用户向system.users集合写入，可以找指定数据库里创建、删除和管理用户
clusterAdmin	只在admin数据库中可用，赋予用户所有分片和复制集相关函数的管理权限
readAnyDatabase	只在admin数据库中可用，赋予用户所有数据库的读权限
readWriteAnyDatabase	只在admin数据库中可用，赋予用户所有数据库的读写权限
userAdminAnyDatabase	只在admin数据库中可用，赋予用户所有数据库的userAdmin权限
dbAdminAnyDatabase	只在admin数据库中可用，赋予用户所有数据库的dbAdmin权限

更多角色相关信息参考：

3.2 docker-compose

执行命令行 `vi docker-compose.yml` 创建服务编排文件，然后在文件中写入下列内容后保存。

当然你也可以使用资源目录中提供 `yml` 文件直接上传到 `mongodb` 目录即可，然后跳过本步骤。

```
1  version: '3'
2  services:
3    mongodb:
4      image: mongo:6.0.2
5      container_name: mongodb
6      environment:
7        - MONGO_INITDB_ROOT_USERNAME=root
8        - MONGO_INITDB_ROOT_PASSWORD=123456
9        - MONGO_INITDB_DATABASE=firstDb
10       - MONGO_USERNAME=awei
11       - MONGO_PASSWORD=123456
12      command: [ --auth ]
13      volumes:
14        - ./db:/data/db
15        - ./mongo-init.sh:/docker-entrypoint-initdb.d/mongo-init.sh
16      ports:
17        - 27017-27019:27017-27019
18      restart: always
```

参数说明：

- environment：是将在mongodb容器上使用的变量
 - MONGO_INITDB_ROOT_USERNAME：填写root用户名
 - MONGO_INITDB_ROOT_PASSWORD：填写root密码
 - MONGO_INITDB_DATABASE：初始化要创建的数据库名称
 - MONGO_USERNAME=awei：初始化普通用户的用户名
 - MONGO_PASSWORD=123456：初始化普通用户的密码
- volumes：定义存放容器的文件/文件夹
 - ./mongo-init.sh:/docker-entrypoint-initdb.d/mongo-init.sh
 - 指定初始化脚本。
 - ./mongo-volume:/data/db
 - 设置本地文件夹db作为MongoDB的持久化数据存储文件夹
- ports：用于定义要公开的端口，在本例中，使用默认的MongoDB端口27017直到27019

创建完成后目录结构如下

```
[root@localhost mongodb]# tree
.
├── docker-compose.yml
└── mongo-init.sh
```

4 启动服务

执行命令 `docker-compose up`。

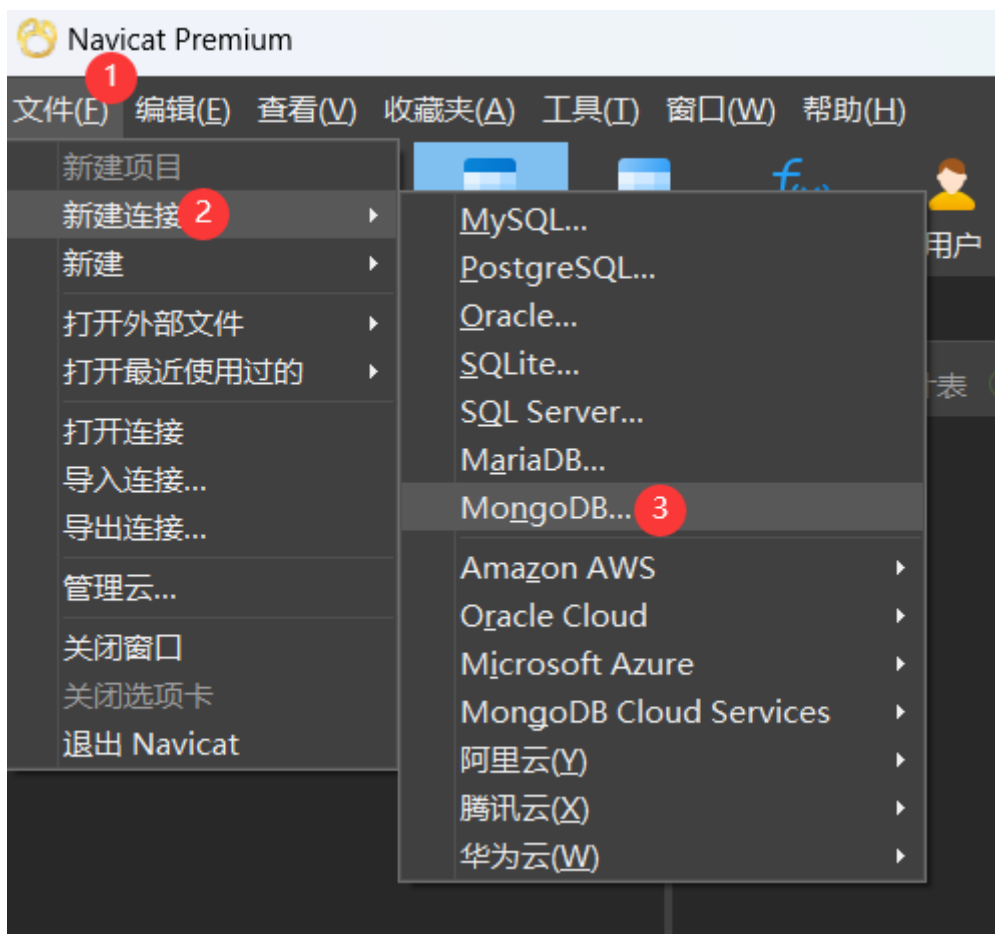
首次启动会拉取镜像，效果如下图所示。

```
[root@localhost mongodb]# docker-compose up
[+] Running 0/10
 : mongodb Pulling
  :: eaead16dc43b Pulling fs layer
  :: 8a00eb9f68a0 Download complete
  :: f683956749c5 Pulling fs layer
  :: b33b2f05ea20 Waiting
  :: 3a342bea915a Waiting
  :: fa956ab1c2f0 Waiting
  :: 138a8542a624 Waiting
  :: acab179a7f07 Waiting
```

如果启动过程没有看到明显的错误，那么表示服务器启动成功，示例如下图所示：

```
patibilityVersion : 0.0 ; context : startup {}
mongodb | {"t":{"$date":"2022-11-11T07:40:50.579+00:00"},"s":"I", "c":"STORAGE", "id":5071100,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.582+00:00"},"s":"I", "c":"CONTROL", "id":20536,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.582+00:00"},"s":"I", "c":"FTDC", "id":20625,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.585+00:00"},"s":"I", "c":"REPL", "id":6015317,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.585+00:00"},"s":"I", "c":"STORAGE", "id":22262,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.590+00:00"},"s":"I", "c":"NETWORK", "id":23015,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.590+00:00"},"s":"I", "c":"NETWORK", "id":23015,
mongodb | {"t":{"$date":"2022-11-11T07:40:50.590+00:00"},"s":"I", "c":"NETWORK", "id":23016,
```

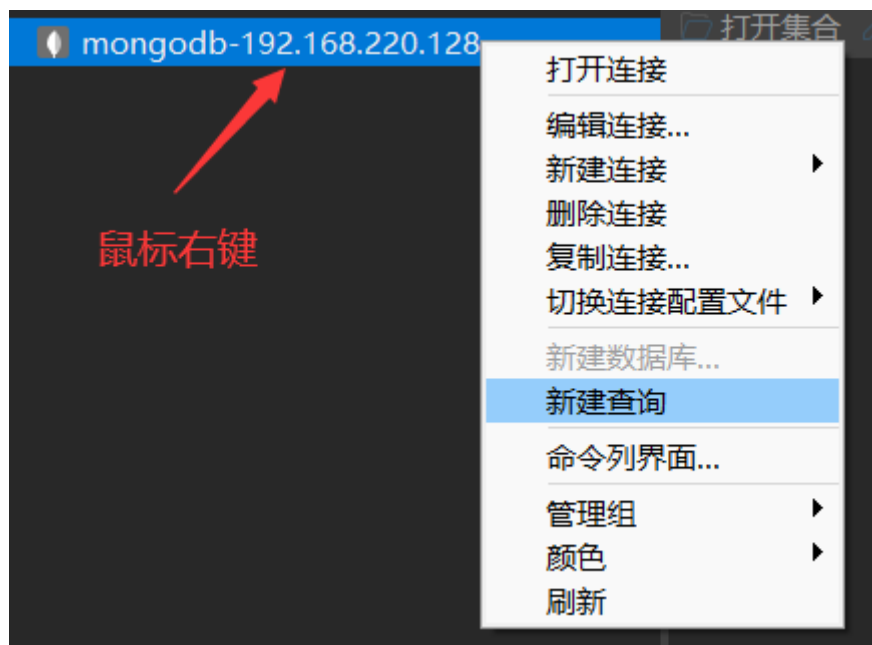
接下来访问一下MongoDB测试，我们使用Navicat来测试。



填写连接信息



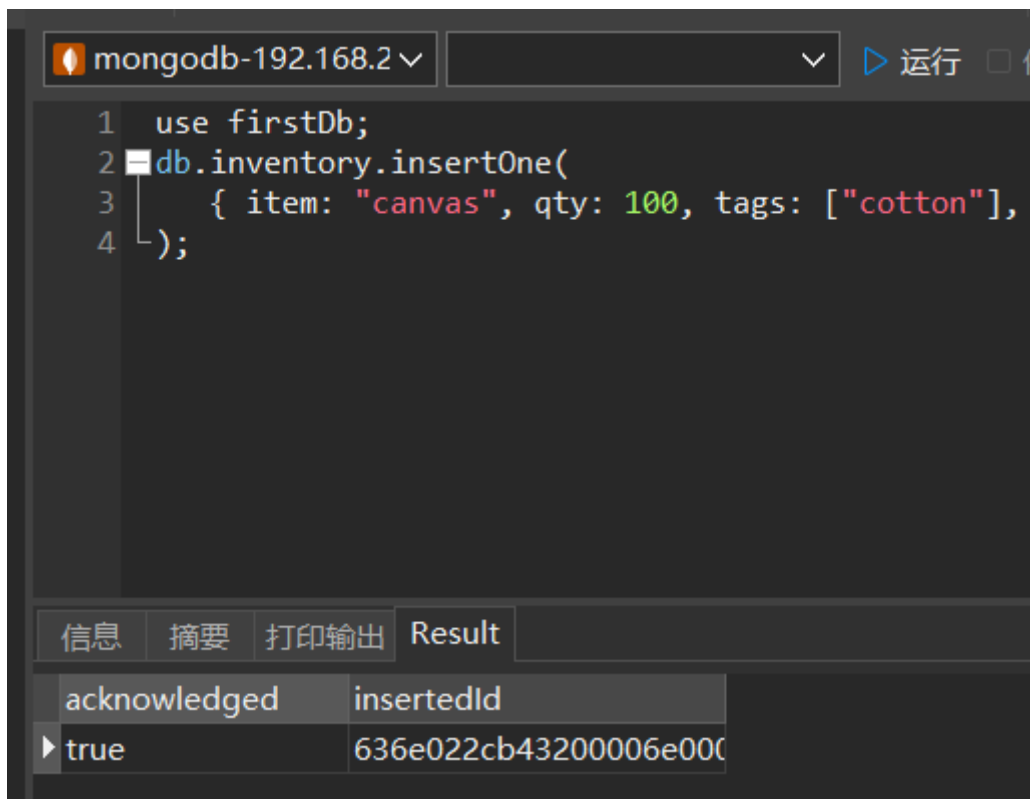
然后通过Navicat创建一张表



然后执行下面的语句

```
1 use firstDb;
2 db.inventory.insertOne(
3   { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom:
4     "cm" } }
5 );
```

下图是执行结果



通过这些测试已经证明数据库是可以正常使用了。

然后 Ctrl + C 结束前台启动，成功停止效果如下图所示

```
^CGracefully stopping... (press Ctrl+C again to force)
[+] Running 1/1
 # Container mongodb Stopped
canceled
[root@localhost mongodb]#
```

执行后台启动，使用命令 `docker-compose up -d`，启动成功如下图所示

```
Unauthorized / 401msg : not authorized on db to execute
^CGracefully stopping... (press Ctrl+C again to force)
[+] Running 1/1
 # Container mongodb Stopped
canceled
[root@localhost mongodb]# docker-compose up -d
[+] Running 1/1
 # Container mongodb Started
[root@localhost mongodb]#
```

使用 `docker ps` 指令查看进程

```
[root@localhost mongodb]# docker ps | grep mongodb
c576d34ccab1    mongo:6.0.2    "docker-e
ngodb
[root@localhost mongodb]#
```

到此MongoDB的安装过程执行完毕。

5 客户端工具

客户端工具下载地址：<https://www.mongodbmanager.com/download>

其他客户端工具：<https://geekflare.com/nosql-client/>