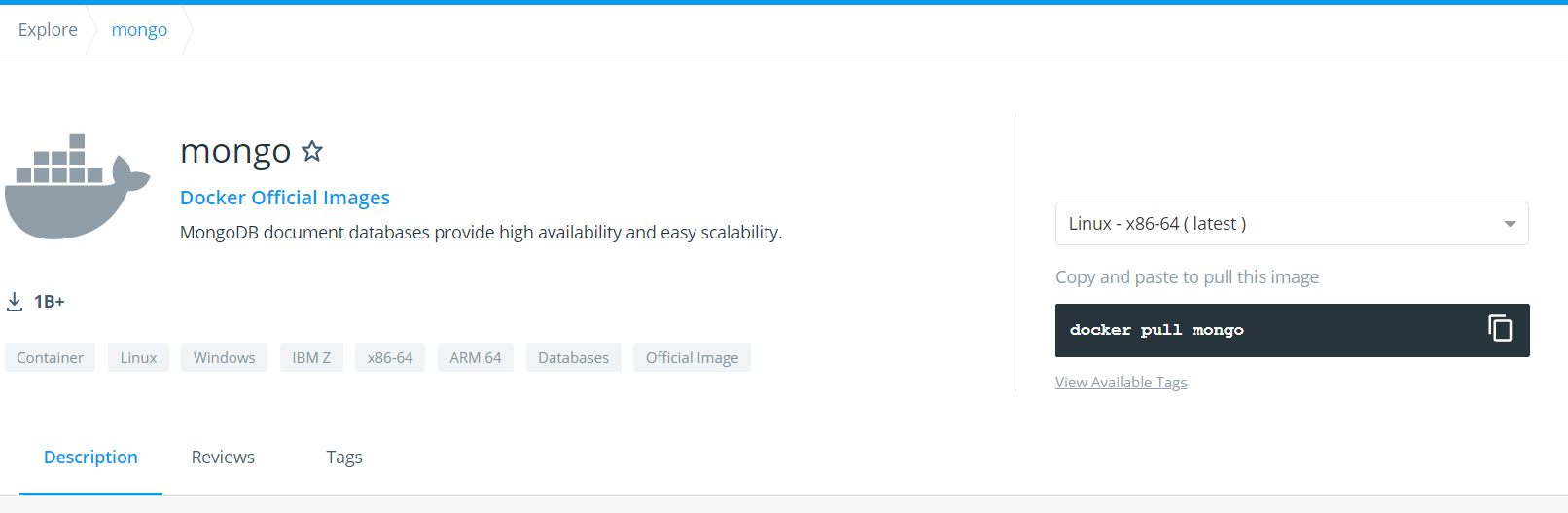
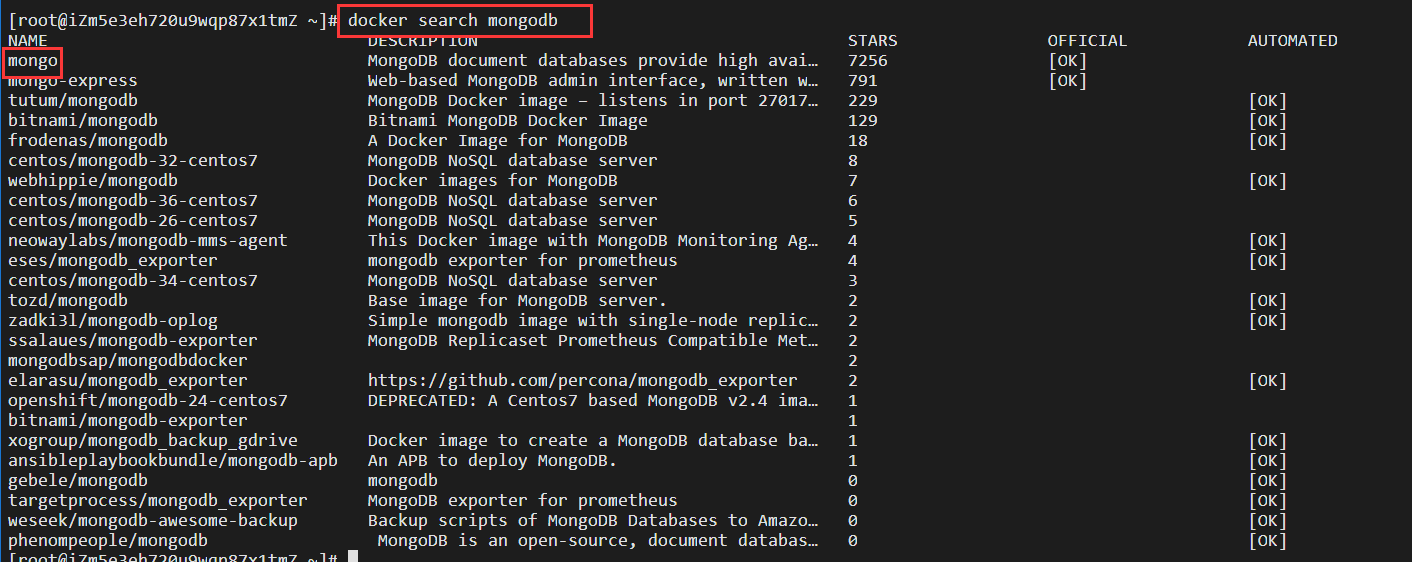
01)、查询 mongodb镜像

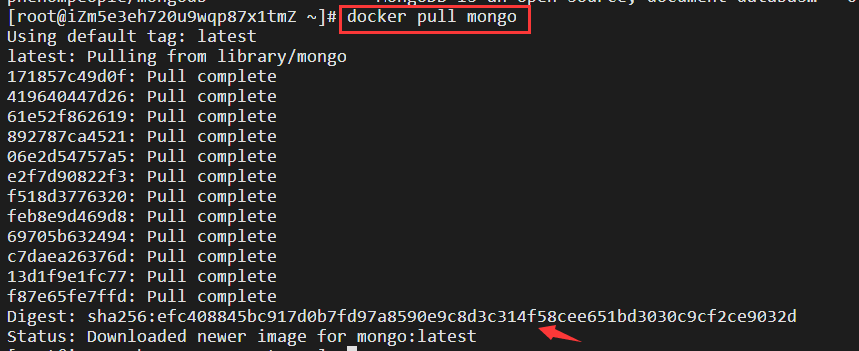


[root@localhost ~]# docker search mongodb



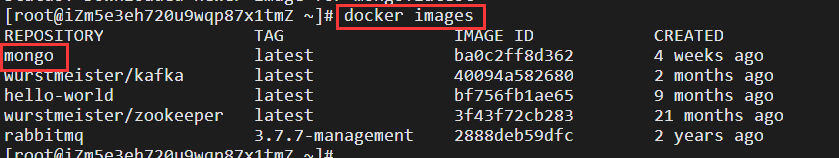
02)、拉取 mongodb镜像 （最新版）

[root@localhost ~]# docker pull mongo



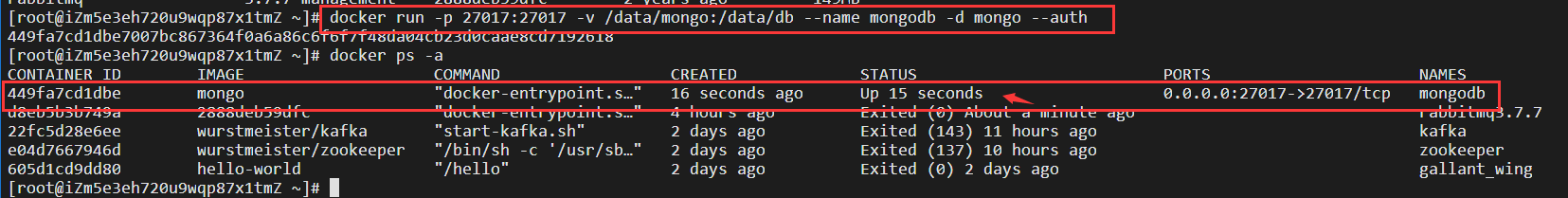
03)、查看下载的 mongodb 镜像

[root@localhost ~]# docker images



04)、运行容器

[root@localhost ~]# docker run -p 27017:27017 -v /data/mongo:/data/db --name mongodb -d mongo –auth



**参数说明**：

-p 27017:27017 ：映射容器服务的 27017 端口到宿主机的 27017 端口。外部可以直接通过 宿主机 ip:27017 访问到 mongo 的服务。

-v 为设置容器的挂载目录，这里是将本机的“/data/mongo”目录挂载到容器中的/data/db中，作为 mongodb 的存储目录

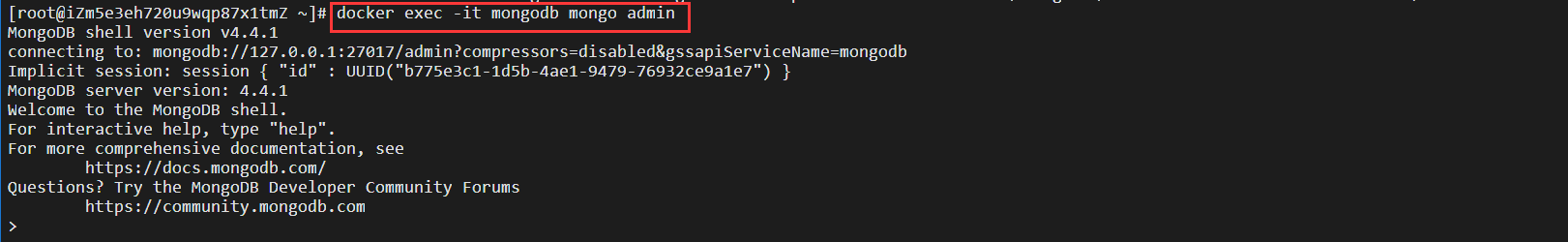
--name 为设置该容器的名称

-d 设置容器以守护进程方式运行

--auth：需要密码才能访问容器服务。

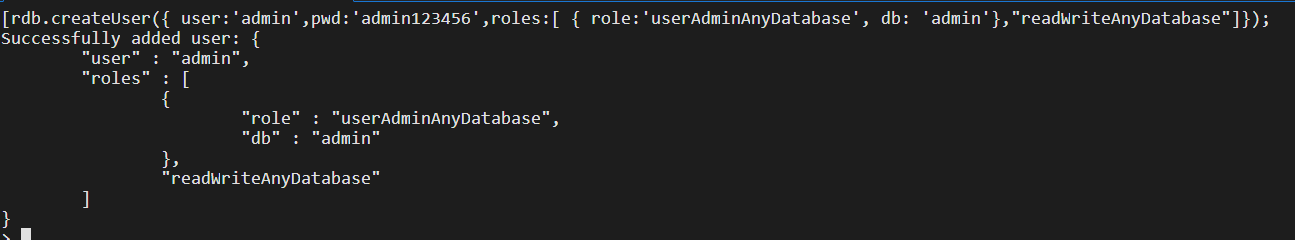
05)、进入容器宿主机

[root@localhost ~]# docker exec -it mongodb mongo admin



06)、创建一个名为 admin，密码为 admin123456 的用户

> db.createUser({ user:'admin',pwd:'admin123456',roles:[ { role:'userAdminAnyDatabase', db: 'admin'},"readWriteAnyDatabase"]});



07)、尝试使用上面创建的用户信息进行连接

> db.auth('admin', 'admin123456')

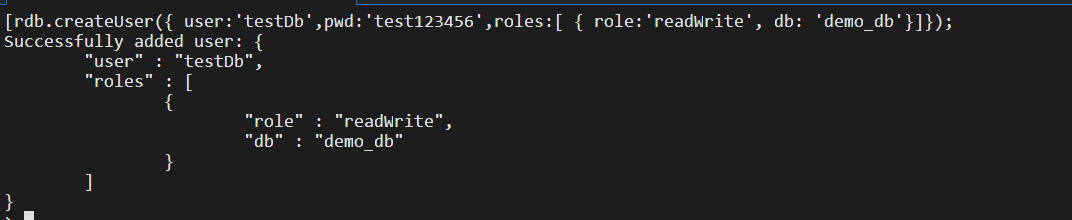


08)、创建一个业务数据库和对应的读写账户

**说明**：再次进入时候需要使用自己数据库：> docker exec -it mongodb mongo demo\_db

> use demo\_db;

> db.createUser({ user:'testDb',pwd:'test123456',roles:[ { role:'readWrite', db: 'demo\_db'}]});



> db.auth('testDb', 'test123456')



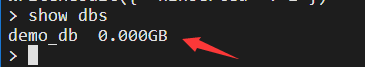
09)、插入一条数据

> db.testTable.insert({"name":"test mongoDb learn"})



10)、查看所有数据库。**说明**：如果查不到是因为没有**插入**数据，可以insert后再重试。

> show dbs

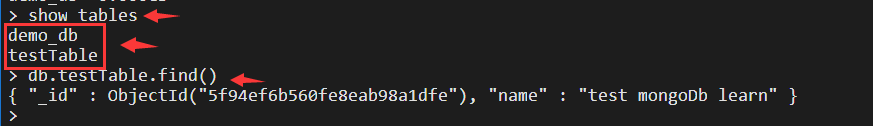


11)、查看表和数据。**说明**: mongodb不需要声明表，可直接写入表数据，即可创建成功!

> show tables

> db.testTable.find()

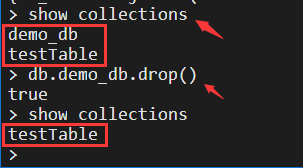
含义：db.表明.find()



12)、删除表 db.collectionName.drop();

> show collections

> db.demo\_db.drop()



13)、使用客户端工具Studio 3T连接mongo

studiot 3T下载地址: <https://studio3t.com/download/>

--------------spring boot 连接 mongodb 操作---------------

# 进入容器，注意此时不是以admin进入,也没有进入mongodb的shell

docker exec -it mongodb bash

#更新源

apt-get update

# 安装 vim

apt-get install vim

# 修改 mongo 配置文件

vim /etc/mongod.conf.orig

1.确保注释掉`# bindIp: 127.0.0.1` 或者改成`bindIp: 0.0.0.0` 即可开启远程连接

2.开启权限认证

security：

authorization: enabled #注意缩进，参照其他的值来改，若是缩进不对可能导致后面服务不能重启。

