链滴

Docker 系列之 MySQL 容器化部署

作者：hyboll

原文链接：https://ld246.com/article/1594368425834

来源网站：链滴

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)
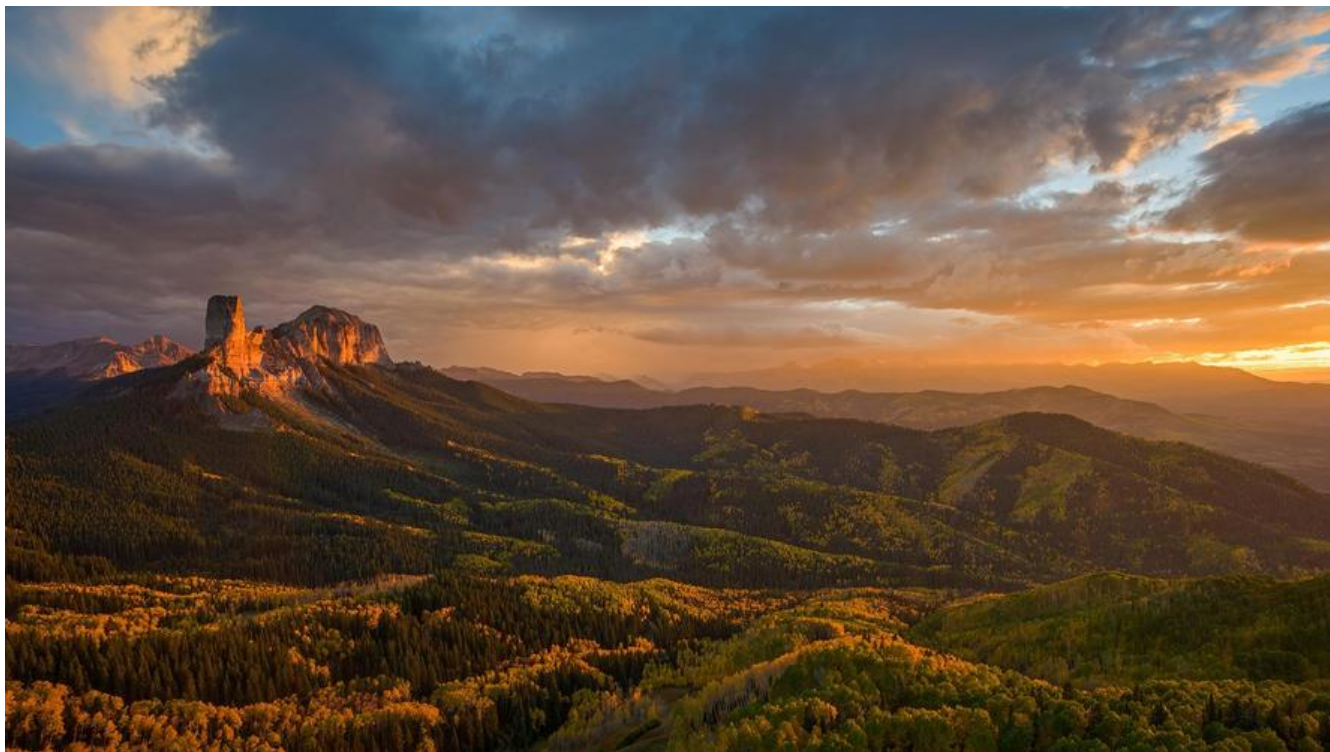
目录

- 镜像获取
- 创建容器
- 远程连接

镜像获取

使用 docker 命令下载最新版的MySQL镜像，docker 的安装和使用可以查看《Docker 系列之 Docke 简单介绍与安装》和《Docker 系列之 Docker 常用命令》

docker pull mysql   #未指定镜像版本默认下载最新版

```
[root@izwz94gcm9bc3smcx5cpf6z ~]# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
8559a31e96f4: Pull complete
d51ce1c2e575: Pull complete
c2344adc4858: Pull complete
fcf3ceff18fc: Pull complete
16da0c38dc5b: Pull complete
b905d1797e97: Pull complete
4b50d1c6b05c: Pull complete
c75914a65ca2: Pull complete
1ae8042bdd09: Pull complete
453ac13c00a3: Pull complete
9e680cd72f08: Pull complete
a6b5dc864b6c: Pull complete
Digest: sha256:8b7b328a7ff6de46ef96bcf83af048cb00a1c86282bfca0cb119c84568b4caf6
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

或者到docker仓库查看MySQL镜像的版本，选择自己需要的版本下载

原文链接：Docker 系列之 MySQL 容器化部署
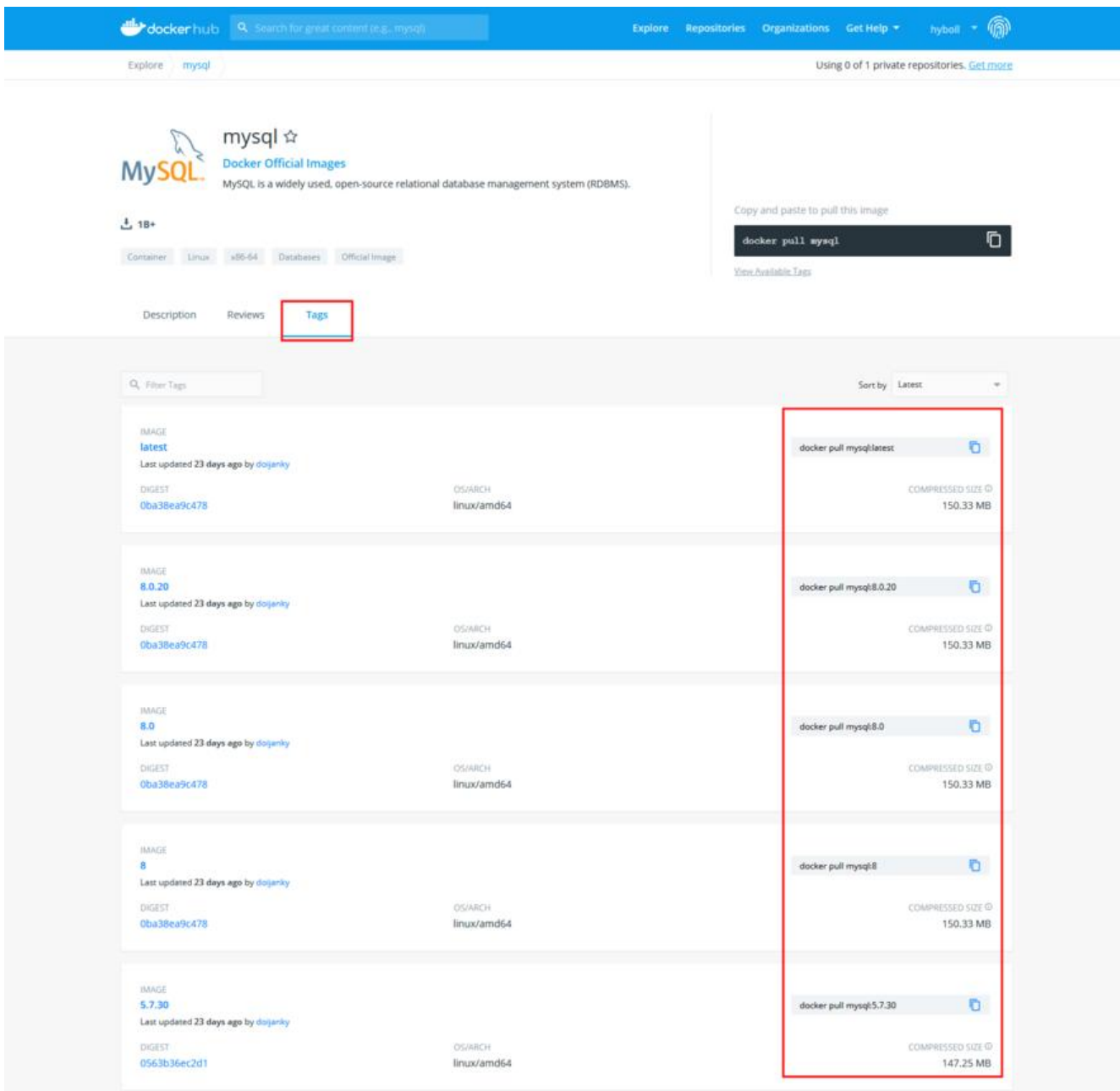
创建容器

使用 docker image ls 查看下载的镜像列表

```
[root@izwz94gcm9bc3smcx5cpf6z ~]# docker image ls
REPOSITORY              TAG       IMAGE ID        CREATED        SIZE
mysql                   latest    be0dbf01a0f3    3 weeks ago    541MB
portainer/portainer     latest    1ec116aed60c    4 weeks ago    79.1MB
yangliu/nps             latest    7f406aac6e34    6 weeks ago    21.5MB
b3log/solo              latest    e7389c9516df    8 weeks ago    143MB
snowdreamtech/frps      latest    4c8c5549fbce    2 months ago   18.6MB
```

创建MySQL容器并运行

docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=password -e MYSQL_ATABASE=test_db -e MYSQL_USER=user -e MYSQL_PASSWORD=password -v /usr/docker/msql/conf:/etc/mysql.conf.d -v /usr/docker/mysql/logs:/logs -v /usr/docker/mysql/data:/var/lib

mysql -d mysql:latest

命令说明

```
* --name mysql              #容器的名称
* -p 3306:3306              #将容器的3306端口映射到主机的3306端口。
* -e MYSQL_ROOT_PASSWORD=password        #初始化root用户的密码。
* -e MYSQL_DATABASE=test_db           #可选，创建名为test_db的数据库
* -e MYSQL_USER=user              #可选，创建拥有test_db所有权限的user用户
* -e MYSQL_PASSWORD=password          #可选，设置user用户的密码
* -v /usr/docker/mysql/conf:/etc/mysql/conf.d   #挂载容器中mysql的配置文件夹至主机的conf
件夹
* -v /usr/docker/mysql/logs:/logs       #挂载容器中mysql的日志文件夹至主机的logs文件夹
* -v /usr/docker/mysql/data:/var/lib/mysql    #挂载容器中mysql的数据文件夹至主机的data文件

* -d mysql:latest              #镜像ID或者Repository:Tag，查看上图
```

容器列表
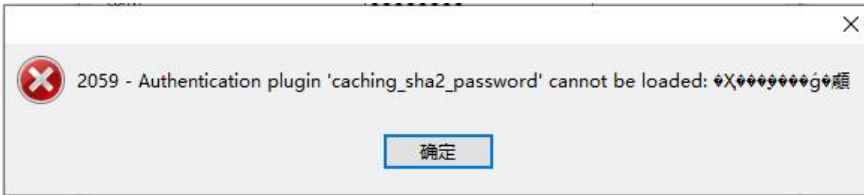
在容器列表中可以看到刚刚运行的mysql容器以及其它正在运行的容器

docker ps

远程连接

使用user用户连接MySQL，MYSQL_USER创建的用户默认拥有远程连接权限

错误

连接MySQL时提示2059错误

原因

MySQL 8之前的版本中加密规则是 mysql_native_password，而在MySQL 8之后，加密规则改为了 aching_sha2_password

解决

更改MySQL的加密规则，首先docker命令进入容器内部

docker exec -it mysql /bin/bash

```
[root@localhost ~]# docker exec -it mysql /bin/bash
root@02f69ac1e0c4:/#
```

接着使用 root用户登录MySQL，输入创建容器时 MYSQL_ROOT_PASSWORD设置的登录密码

mysql -u root -p   #登录

```
root@02f69ac1e0c4:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

登录成功后执行以下操作

use mysql;   #选择数据库

ALTER USER 'user'@'%' IDENTIFIED BY 'password' PASSWORD EXPIRE NEVER;   #更改加密方，其中的'password'为需要设置的密码

ALTER USER 'user'@'%' IDENTIFIED WITH mysql_native_password BY 'password'; #更新用户密

FLUSH PRIVILEGES; #刷新权限

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ALTER USER 'user'@'%' IDENTIFIED BY 'password' PASSWORD EXPIRE NEVER;
Query OK, 0 rows affected (0.13 sec)

mysql> ALTER USER 'user'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

完成以上操作后再次使用Navicat远程连接MySQL，连接成功