

Docker 应用部署-MySQL

一、安装MySQL

1 搜索mysql镜像

```
1 docker search mysql
```

2 拉取mysql镜像

```
1 docker pull mysql:8.0.20
```

3 创建容器

通过下面的命令，创建容器并设置端口映射、目录映射

```
1 # 在用户名目录下创建mysql目录用于存储mysql数据信息
2 mkdir /home/mysql
3 cd /home/mysql
4 # 创建docker容器
5 docker run -id \
6 -p 3306:3306 \
7 --name mysql8 \
8 --restart always \
9 -v $PWD/conf:/etc/mysql/conf.d \
10 -v $PWD/log:/var/log/mysql \
11 -v $PWD/data:/var/lib/mysql \
12 -e MYSQL_ROOT_PASSWORD=123456 \
13 mysql:8.0.20 \
14 --lower_case_table_names=1
```

参数说明：

- **-p 3306:3306**：将容器的3306端口映射到宿主机的3306端口。
- **-v \$PWD/conf:/etc/mysql/conf.d**：将主机当前目录下的conf/目录挂载到容器的 /etc/mysql/conf.d目录。
- **-v \$PWD/log:/var/log/mysql**：将主机当前目录下的log目录挂载到容器的 /var/log/mysql日志目录
- **-v \$PWD/data:/var/lib/mysql**：将主机当前目录下的data目录挂载到容器的/var/lib/mysql数据目录
- **-e MYSQL_ROOT_PASSWORD=123456**：初始化root用户的密码。
- **--lower_case_table_names**：用于控制表名和数据库名在比较时是否应该忽略大小写。它的值可以是0、1或2，分别表示：
 - 0：表名存储为给定的大小和比较是区分大小写的。
 - 1：表名存储在磁盘是小写的，但是比较的时候是不区分大小写。

- 2: 表名存储为给定的大小写但是比较的时候是小写的。

注意：在云服务器上部署的时候需要把密码复杂度设置高一点，避免被无差别扫描攻击，然后在做端口映射的时候宿主机端口就不要使用3306了

```
1 # 查看当前启动的容器
2 docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2da4390a9919	mysql:8.0.20	"docker-entrypoint.s..."	6 seconds ago	Up 6 seconds	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	mysql8

看到STATUS为up状态表示创建容完成。

4 配置防火墙

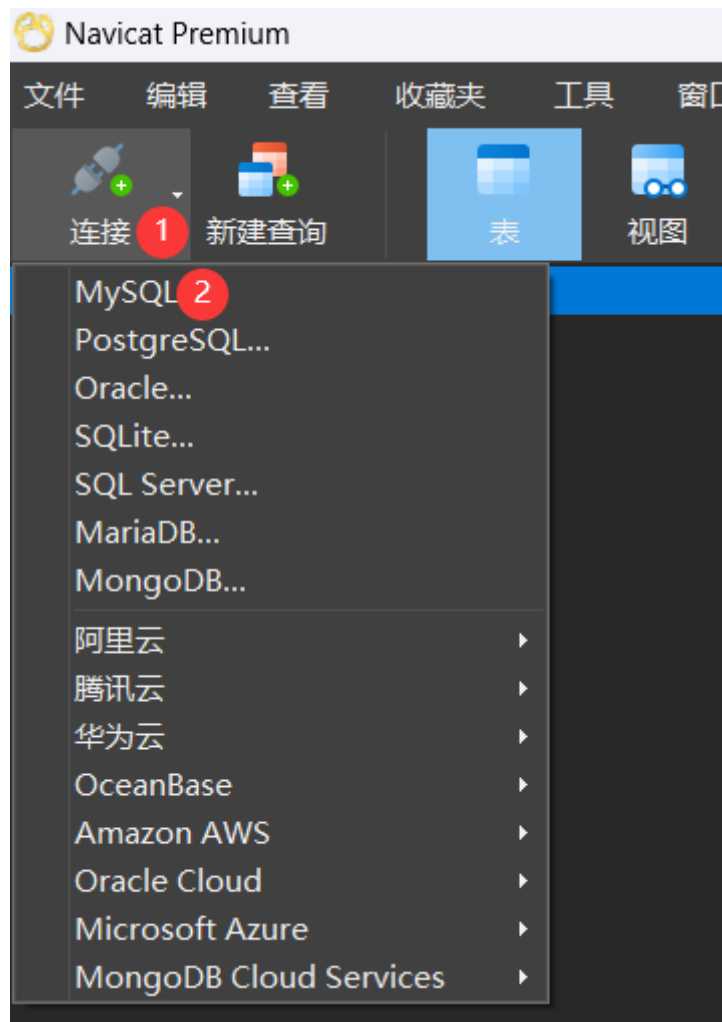
想要外部访问到你的数据库需要开放刚才映射的宿主机端口，我这里映射的是3306端口，所以开放3306端口。

注意：如果是云服务器，不需要通过下面命令开放端口，需要登录云控制台，配置防火墙入方向规则

```
1 #查看是否已经开放3306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 3306/tcp --permanent
5 #重新加载防火墙
6 firewall-cmd --reload
```

5 远程连接测试

打开Navicat，然后新建一个MySQL连接。



填写连接参数信息，测试并保存连接参数信息。



6 mysql参数调整

```
1 #进入目录修改配置
2 cd /home/mysql/conf
3 #创建配置文件
4 vi my.cnf
```

配置文件中写入以下内容

注意：写入配置的时候将注释内容也去掉，避免中文影响，还每条配置占用一行，不要一条配置写成多行。

```
1 [mysqld]
2 # 修改sql_mode内容，目的在于去掉ONLY_FULL_GROUP_BY
3 sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
4 # 调整mysql最大连接数
5 max_connections=1000
```

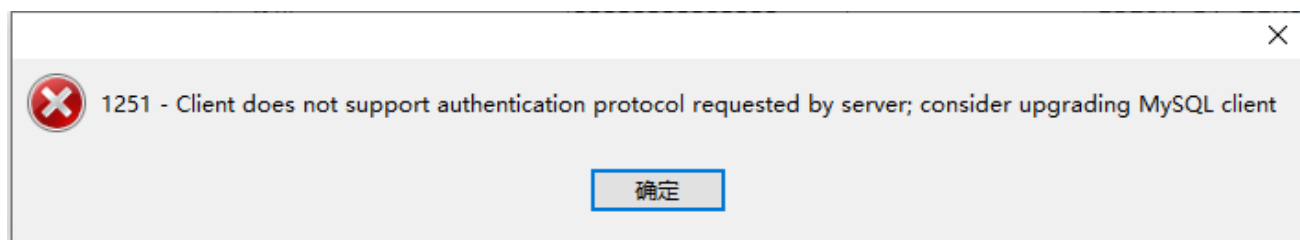
修改完配置后，重启容器，连接mysql，然后执行`show variables like 'sql_mode';`命令来查看对应变量值是否正确，示例如下图所示



二、常见问题的解决方案

1 低版本navicat连接不了mysql8

连接时候出现类似下图的错误



出现这个问题是由于mysql8之前的版本中加密规则是`mysql_native_password`，而在mysql8之后，加密规则是`caching_sha2_password`

处理方式有两种：

- 升级Navicat（推荐）
- 新增一个使用`mysql_native_password`加密方式的用户
 - 进入容器，并连接mysql
 - ```
1 docker exec -it mysql8 /bin/bash
2 mysql -uroot -p
```
  - 然后输入命令

```

1 # 进入MySQL数据库
2 use mysql;
3 # 创建一个新的登录用户
4 create user 'root1'@'%' identified by '123456';
5 # 授予访问权限
6 grant all on *.* to 'root1'@'%';
7 # 修改登录账号密码，使用mysql_native_password加密方式
8 ALTER USER 'root1'@'%' IDENTIFIED WITH mysql_native_password BY '123456';
9 # 刷新权限
10 FLUSH PRIVILEGES;

```

## 2 忘记密码怎么修改

在my.cnf中加入

```

1 #免密码登录
2 skip-grant-tables

```

然后重启MySQL容器，然后进入容器

```

1 # 进入容器
2 docker exec -it mysql8 /bin/bash
3 # 登录数据库
4 mysql -uroot -p
5 # 提示输入密码的时候，回车即可，因为已开启免密登录，然后输入下列指令将root密码之置空
6 UPDATE mysql.user SET authentication_string='' WHERE user='root' and host='%';

```

退出容器后，将配置文件中免密登录修改回来，重启MySQL容器，然后在进入容器，来修改root的密码

```

1 # 进入容器
2 docker exec -it mysql8 /bin/bash
3 # 登录数据库
4 mysql -uroot -p
5 # 提示输入密码的时候，回车即可，因为密码已经置空了，然后输入下列指令修改root密码
6 ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY '123456';
7 # ALTER USER 'root'@'%' IDENTIFIED WITH caching_sha2_password BY '123456';
8 # 刷新权限
9 FLUSH PRIVILEGES;

```

## 3 数据库时间与主机不同步

使用docker安装mysql，时间要比主机晚8个小时，此时需要将本地时间文件复制到MySQL容器中

```

1 docker cp /usr/share/zoneinfo/Asia/Shanghai 容器名称:/etc/localtime

```

```

[root@aubinll ~]# docker cp /usr/share/zoneinfo/Asia/Shanghai mysql8:/etc/localtime

```

然后进入容器再看看时间是否正常

```
external
[root@aubinll ~]# date
Thu Dec 23 11:18:15 CST 2021
[root@aubinll ~]# docker exec -it mysql8 /bin/bash
root@7a0e4adfee2e:/# date
Thu Dec 23 11:18:25 CST 2021
root@7a0e4adfee2e:/#
```

外部linux服务器

mysql容器