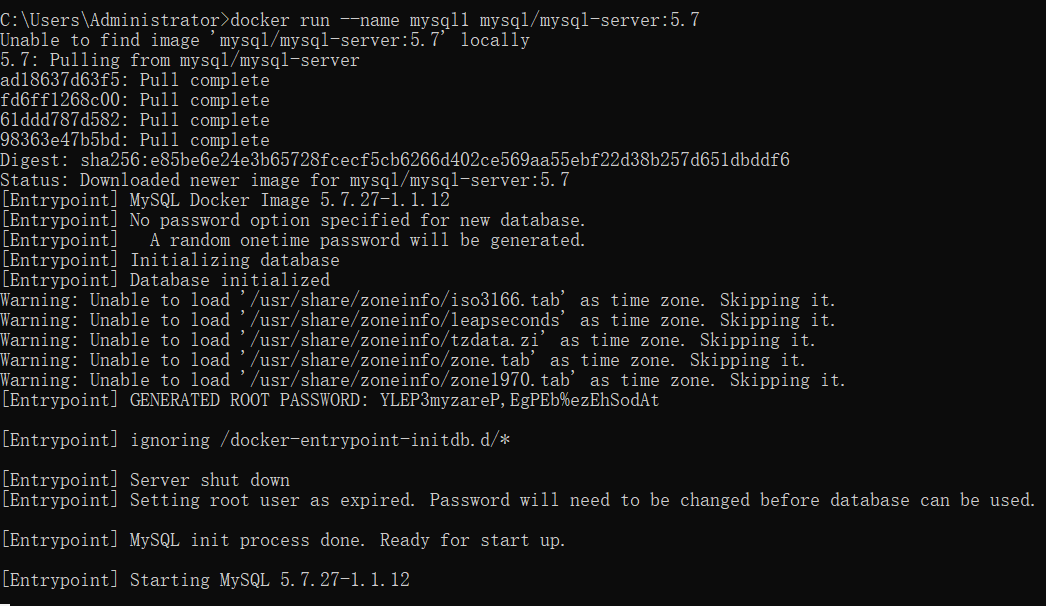
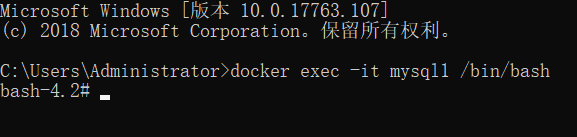
docker安装使用mysql

docker run --name mysql1 mysql/mysql-server:5.7



docker exec -it mysql1 /bin/bash

与docker虚拟的容器进行交互，在docker虚拟的容器内执行命令



解读：

docker exec ：在运行的容器中执行命令

语法

docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

OPTIONS说明：

-d :分离模式: 在后台运行

-i :即使没有附加也保持STDIN 打开

-t :分配一个伪终端

实例

在容器 mynginx 中以交互模式执行容器内 /root/runoob.sh 脚本:

runoob@runoob:~$ docker exec -it mynginx /bin/sh /root/runoob.sh

http://www.runoob.com/

在容器 mynginx 中开启一个交互模式的终端

也可以通过 docker ps -a 命令查看已经在运行的容器，然后使用容器 ID 进入容器。

查看已经在运行的容器 ID：

# docker ps -a

...

9df70f9a0714 openjdk "/usercode/script.sh…"

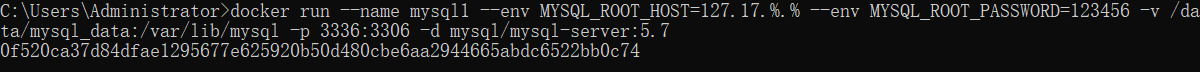
...

第一列的 9df70f9a0714 就是容器 ID。

通过 exec 命令对指定的容器执行 bash:

# docker exec -it 9df70f9a0714 /bin/bash

docker run --name mysql1 --env MYSQL\_ROOT\_HOST=127.17.%.% --env MYSQL\_ROOT\_PASSWORD=123456 -v /data/mysql\_data:/var/lib/mysql -p 3336:3306 -d mysql/mysql-server:5.7



--env MYSQL\_ROOT\_PASSWORD=123456 指定密码为123456

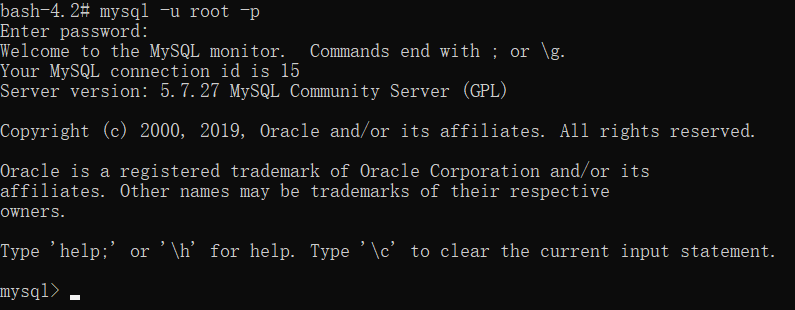
进入容器，进入mysql

docker exec -it mysql1 /bin/bash

mysql -u root -p

输入密码123456





docker run ：创建一个新的容器并运行一个命令

语法

docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

OPTIONS说明：

-a stdin: 指定标准输入输出内容类型，可选 STDIN/STDOUT/STDERR 三项；

-d: 后台运行容器，并返回容器ID；

-i: 以交互模式运行容器，通常与 -t 同时使用；

-P: 随机端口映射，容器内部端口随机映射到主机的高端口

-p: 指定端口映射，格式为：主机(宿主)端口:容器端口

-t: 为容器重新分配一个伪输入终端，通常与 -i 同时使用；

--name="nginx-lb": 为容器指定一个名称；

--dns 8.8.8.8: 指定容器使用的DNS服务器，默认和宿主一致；

--dns-search example.com: 指定容器DNS搜索域名，默认和宿主一致；

-h "mars": 指定容器的hostname；

-e username="ritchie": 设置环境变量；

--env-file=[]: 从指定文件读入环境变量；

--cpuset="0-2" or --cpuset="0,1,2": 绑定容器到指定CPU运行；

-m :设置容器使用内存最大值；

--net="bridge": 指定容器的网络连接类型，支持 bridge/host/none/container: 四种类型；

--link=[]: 添加链接到另一个容器；

--expose=[]: 开放一个端口或一组端口；

--volume , -v: 绑定一个卷

实例

使用docker镜像nginx:latest以后台模式启动一个容器,并将容器命名为mynginx。

docker run --name mynginx -d nginx:latest

使用镜像nginx:latest以后台模式启动一个容器,并将容器的80端口映射到主机随机端口。

docker run -P -d nginx:latest

使用镜像 nginx:latest，以后台模式启动一个容器,将容器的 80 端口映射到主机的 80 端口,主机的目录 /data 映射到容器的 /data。

docker run -p 80:80 -v /data:/data -d nginx:latest

绑定容器的 8080 端口，并将其映射到本地主机 127.0.0.1 的 80 端口上。

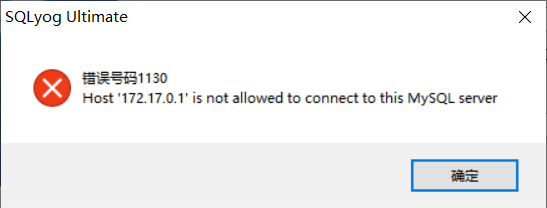
$ docker run -p 127.0.0.1:80:8080/tcp ubuntu bash

使用镜像nginx:latest以交互模式启动一个容器,在容器内执行/bin/bash命令。

runoob@runoob:~$ docker run -it nginx:latest /bin/bash

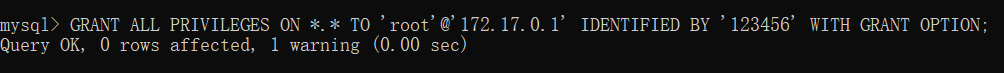
root@b8573233d675:/#

mysql连接不上

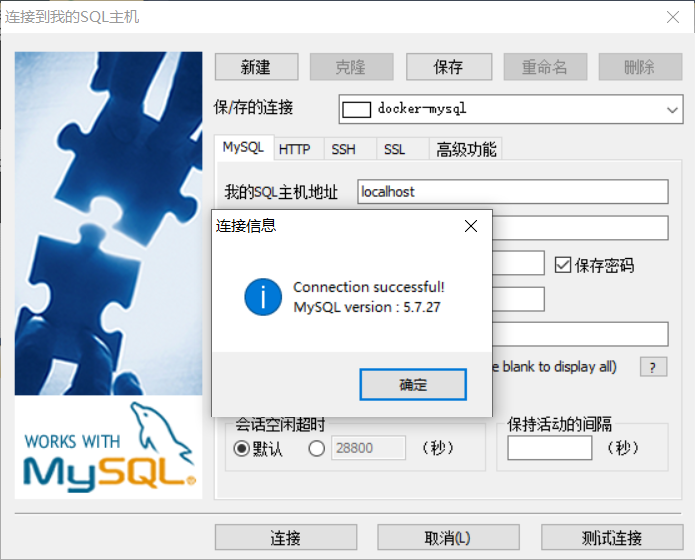


进入docker容器的mysql，然后执行命令：

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'172.17.0.1' IDENTIFIED BY '123456' WITH GRANT OPTION;



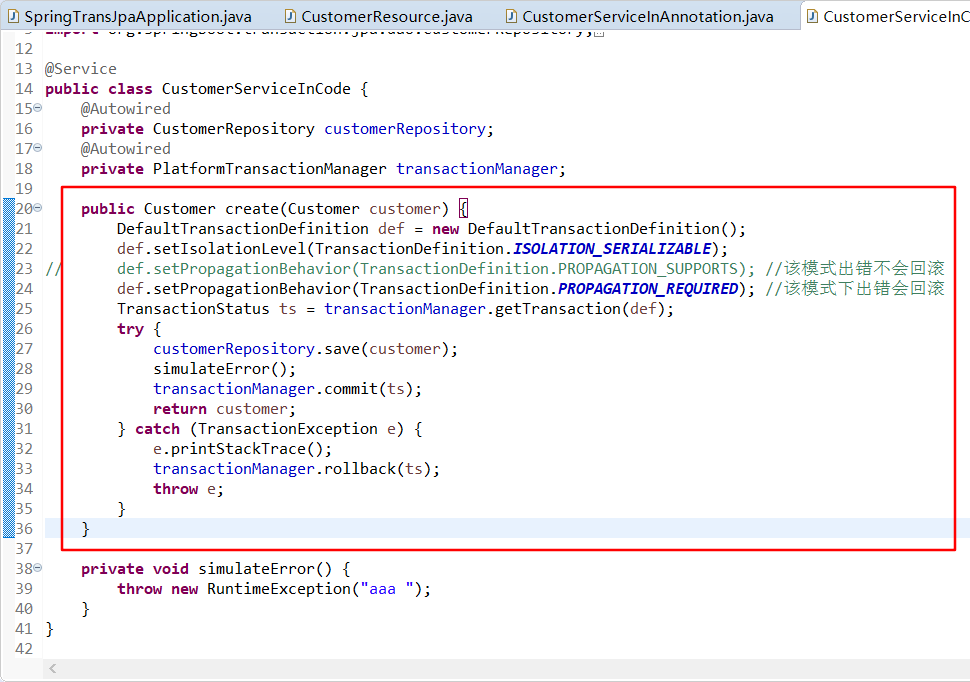
再次验证可以了



事务的实现方式

代码实现

代码实现如下：

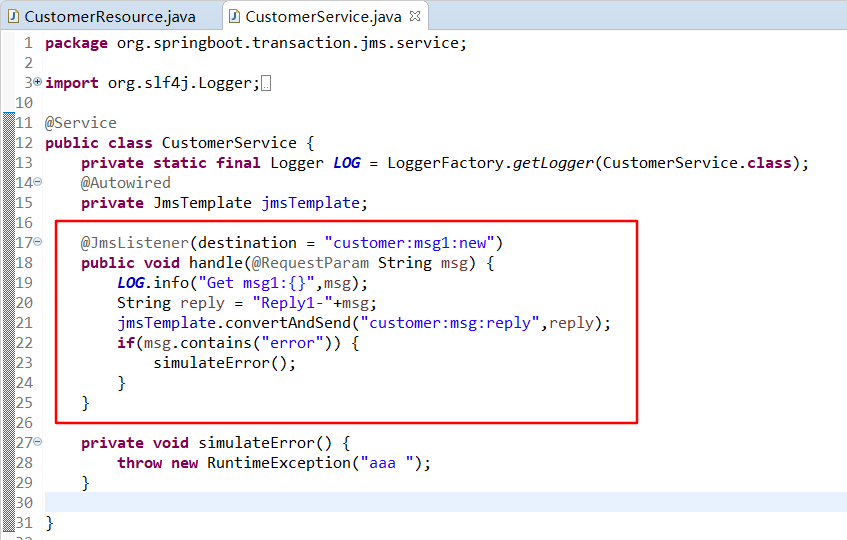


标签实现

标签实现加上@Transactional注解即可

jms的事务

通过@JmsListener的方式加入的话就会有事务，会回滚



直接调用的话则没有事务，不会回滚