

五、Docker安装MySQL (和第四部分2选1)

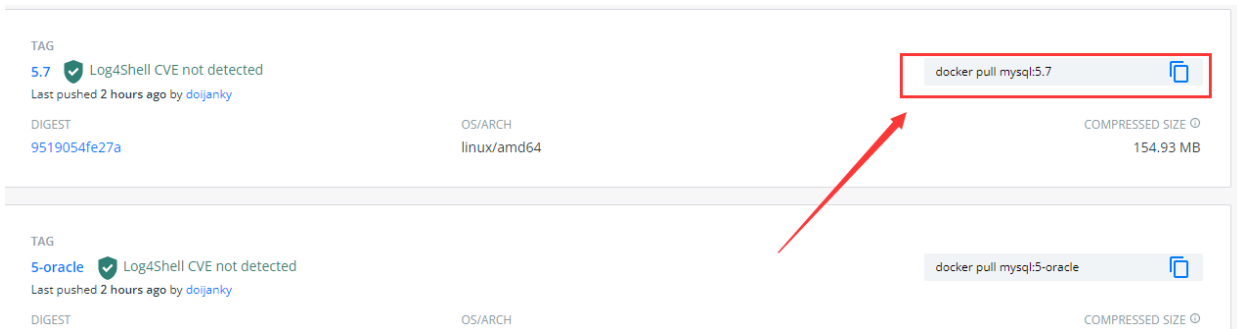
五、Docker安装MySQL (和第四部分2选1)

(备注：第三部分有docker具体的安装使用)

1) 查看可用的 MySQL 版本

```
1 docker search mysql #
2 docker search mysql:5.7 #带着版本号查询
```

也可以通过浏览器访问 MySQL 镜像库地址: https://hub.docker.com/_/mysql?tab=tags



2) 拉取 MySQL 镜像

```
1 docker pull mysql:5.7
```

3) 查看本地镜像

```
1 docker images
```

4) 创建本地数据卷

```
1 mkdir -p /docker/mysql/conf
2 mkdir -p /docker/mysql/data
```

目录设置权限。3种不同类型：文件所有者、群组用户、其他用户的 权限制都设为7。

```
1 chmod 777 /docker/mysql/conf
2 chmod 777 /docker/mysql/data
```

5) 运行容器

```
1 docker run --restart=always -d -p 3307:3306 --privileged=true -v
  /docker/mysql/conf/my.cnf:/etc/my.cnf -v /docker/mysql/data:/var/lib/mysql
```

```
-e MYSQL_ROOT_PASSWORD=root --name mysql mysql:5.7 --character-set-server=utf8mb4 --collation-server=utf8mb4_general_ci
```

参数说明:

run 运行一个容器

--restart=always 开机启动，失败也会一直重启

-restart具体参数值详细信息:

no - 容器退出时，不重启容器;

on-failure - 只有在非0状态退出时才从新启动容器;

always - 无论退出状态是如何，都重启容器;

-d 后台运行

-p 3307:3306 端口映射，将服务器的3307端口映射到镜像的3306端口

--privileged=true 设置mysql用户，否则外部不能使用root用户登录

-v /docker/mysql/conf/my.cnf:/etc/my.cnf 服务器的/docker/mysql/conf/my.cnf配置映射到docker的my.cnf

-v /docker/mysql/data:/var/lib/mysql 映射数据库的数据目录，避免docker删除重新运行mysql容器，导致数据丢失

-e MYSQL_ROOT_PASSWORD=root 设置root账号的密码

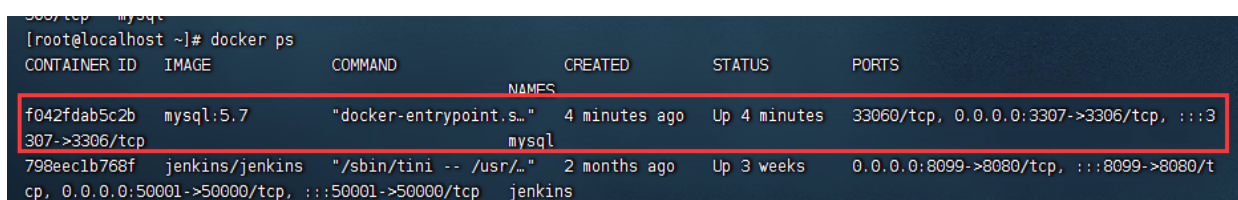
--name mysql 设置容器的名称为mysql

mysql:5.7 从docker镜像mysql:5.7启动一个容器

--character-set-server=utf8mb4 --collation-server=utf8mb4_general_ci 设置数据库默认编码

6) 查看运行容器

```
1 docker ps #查看所有运行容器
```



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
f042fdab5c2b	mysql:5.7	"docker-entrypoint.s..."	4 minutes ago	Up 4 minutes	33060/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3306/tcp
798eec1b768f	jenkins/jenkins	"/sbin/tini -- /usr/_..."	2 months ago	Up 3 weeks	0.0.0.0:8099->8080/tcp, :::8099->8080/tcp, 0.0.0.0:50001->50000/tcp, :::50001->50000/tcp

7) 进入容器

```
1 docker exec -it mysql /bin/bash #进入名称为mysql的容器，也可以用容器id进入
```

8) 登录MySQL，创建用户，宿主机开放端口，登录测试

1】登录mysql, 验证是否安装成功

```
1 mysql -uroot -proot
```

```
root@f042fdab5c2b:/# mysql -uroot -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.38 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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>
```

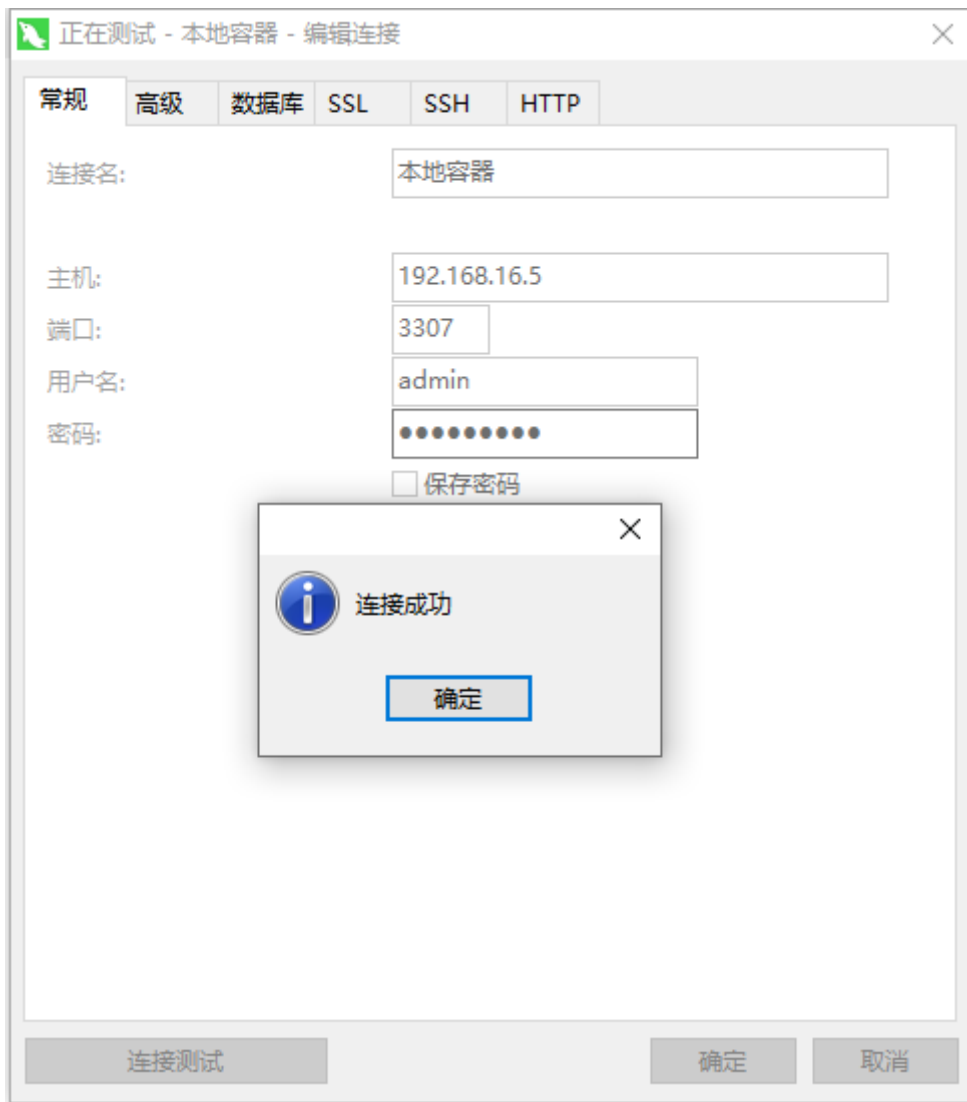
2】创建用户admin, 密码Admin123.。授权所有权限 (all) , 允许正在任意主机登录 (%) , 任意库任意表 (*.*)

```
1 grant all on *.* to 'admin'@'%' identified by 'Admin123.';
```

3】开启宿主机防火墙3307端口的的外部访问, 在启动容器的时候把3306端口映射到宿主机的3307了

```
1 exit #退出mysql
2 exit #退出容器
3 firewall-cmd --zone=public --add-port=3307/tcp --permanent
4 firewall-cmd --reload #配置立即生效
```

4】通过宿主机ip+3307端口+帐号密码, 测试连接成功。



9) 重启容器服务，看看mysql容器是否会自启

备注：以下是在创建容器的时候没有加入 `--restart=always` 命令的测试结果。

感兴趣的话大家可以新run一个容器，去掉 `--restart=always` 命令测试以下，没兴趣可以忽略本步骤。

经测试，外部电脑通过navicat 工具 通过ip、端口、帐号、密码能连接容器数据库

```
1 sudo service docker restart #重启docker容器
```

测试数据库无法远程连接，说明docker容器并没有自启。

执行以下命令

```
1 docker update mysql --restart=always #mysql容器自动启动
2 sudo service docker restart #重启docker容器
```

测试数据库可以远程连接，说明docker容器已经重启！