

Docker 命令

Docker安装Mysql

1.拉取镜像

2.运行mysql

```
1 docker run -d -p 3306:3306 --name [Name] [ImageName]
2 //示例
3 docker run -d -p 3306:3306 --name mysql01 mysql/mysql-server
```

3.进入容器

```
1 docker exec -it [ContainerName] bash
2 //示例
3 docker exec -it mysql01 bash
```

4.进入mysql的命令行

命令行进入mysql的root账户：

```
1 mysql -uroot -p
```

这时候操作mysql会报错，让修改密码 请看 常见错误解决模块

创建新用户

```
1 mysql> CREATE USER 'test'@'localhost' IDENTIFIED BY 'pwd123456';
2 mysql> GRANT ALL PRIVILEGES ON *.* TO 'test'@'localhost' WITH GRANT
  OPTION;
3 mysql> CREATE USER 'test'@'%' IDENTIFIED BY 'pwd123456';
4 mysql> GRANT ALL PRIVILEGES ON *.* TO 'test'@'%' WITH GRANT OPTION;
```

修改MYSQL 默认字符集

添加到my.cnf

```
1 [mysqld]
2 character-set-server=utf8
3 [client]
4 default-character-set=utf8
5 [mysql]
6 default-character-set=utf8
```

查看字符集命令

```
1 show variables like '%char%';
```

docker容器参数启动Mysql (用容器参数添加新用户和修改root账户密码, 修改字符集)

直接复制这个就可以了

```
1 docker run -d -p 3306:3306 -e MYSQL_USER="test" -e
  MYSQL_PASSWORD="password" -e MYSQL_ROOT_PASSWORD="password" --name [Name]
  [ImageName] --character-set-server=utf8 --collation-server=utf8_general_ci
```

这个和上面一样只不过用换行整理过 (不要复制这个, 用不了)

```
1 docker run -d -p 3306:3306
2 -e MYSQL_USER="test"
3 -e MYSQL_PASSWORD="password"
4 -e MYSQL_ROOT_PASSWORD="password"
5 --name [Name] [ImageName]
6 --character-set-server=utf8
7 --collation-server=utf8_general_ci
```

挂载资料卷

mysql挂载资料卷

#注意:需要先创建/docker/mysql/config/my.cnf文件和/docker/mysql/data文件夹。

my.cnf如下

```
1 [mysqld]
2 user=mysql
3 character-set-server=utf8
4 [client]
5 default-character-set=utf8
6 [mysql]
7 default-character-set=utf8
```

用外部volume

```
1 docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD="root" --name mysql01 -
  v=/docker/mysql/config:/mnt/etc/my.cnf -
  v=/docker/mysql/data:/mnt/var/lib/mysql mysql/mysql-server
```

mongo挂载资料卷

```
1 docker run -d -p 27017:27017 --name mongo01 -v
  /docker/mongo/config:/data/configdb -v /docker/mongo/db:/data/db mongo
```

常见错误解决

修改密码语句

mysql5.7版本语法:

```
1 SET PASSWORD FOR 'root'@'localhost' = PASSWORD('password');
```

mysql5.7之后版本语法：

```
1 ALTER user 'root'@'localhost' IDENTIFIED BY 'password'
```

navicat连接mysql报错1251解决方案

MySQL8.0，发现Navicat连接不上，总是报错1251；

原因是MySQL8.0版本的加密方式和MySQL5.0的不一样，连接会报错。

1. 先通过命令行进入mysql的root账户：

```
1 mysql -uroot -p
```

2. 更改加密方式

```
1 ALTER USER 'root'@'localhost' IDENTIFIED BY 'password' PASSWORD EXPIRE  
NEVER;
```

3. 重新修改密码（注意修改你要远程连接的用户）

```
1 ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'password';
```

4. 刷新

```
1 FLUSH PRIVILEGES;
```

Docker run with 环境变量和镜像配置

- ```
1 备注： : When this is true (which is its default state, unless
MYSQL_ROOT_PASSWORD is set or MYSQL_ALLOW_EMPTY_PASSWORD is set to true),
a random password for the server's root user is generated when the Docker
container is started
2 当这个MYSQL_RANDOM_ROOT_PASSWORD 为true 时 为root用户设置密码是没有作用的还是会
生成随机的密码，设置为false 或者不加这个环境变量
```

用外部mount

```
1 docker run -d -p 3306:3306
2 -e MYSQL_USER="username"
3 -e MYSQL_PASSWORD="password"
4 -e MYSQL_ROOT_PASSWORD="password"
5 --mount type=bind,source=C:\mysql\conf\my.cnf,target=/etc/my.cnf
6 --mount type=bind,source=C:\mysql\data,target=/var/lib/mysql
7 --name mysql01
8 mysql/mysql-server
9
```

用外部volume

```
1 docker run -d -p 3306:3306 --name mysql01 -
 v=/docker/mysql/config/my.cnf:/etc/my.cnf -
 v=/docker/mysql/config/data:/var/lib/mysql mysql/mysql-server
```

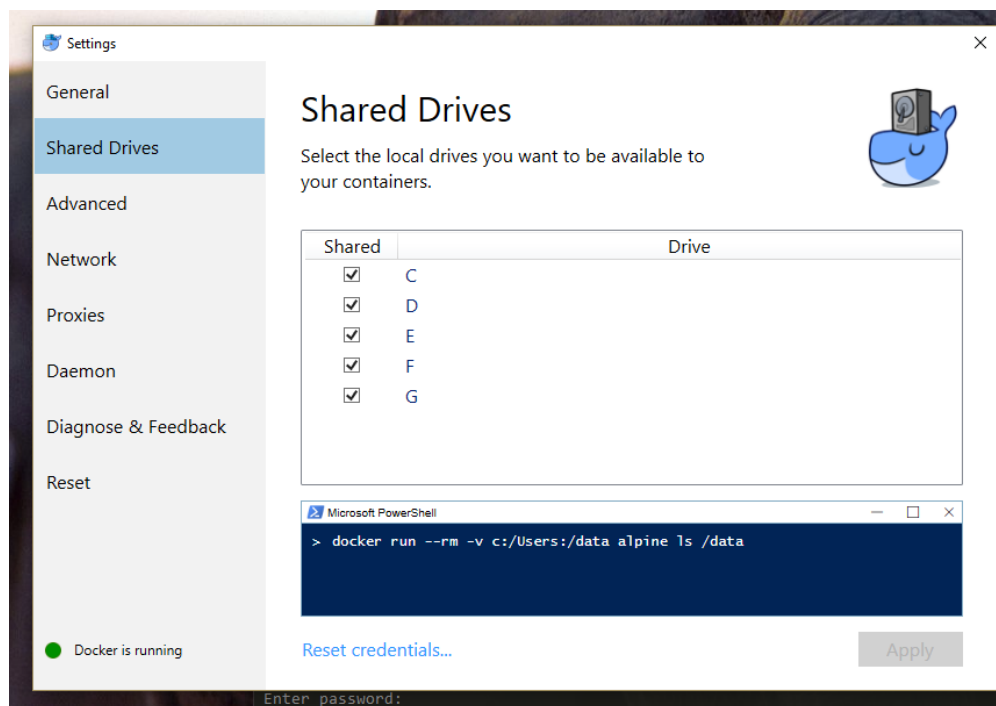
在window环境下,挂载mysql 配置文件

详细步骤:

1.提前在指定的目录下创建一个my.cnf文件,目录名最好为英文且不带特殊符号和空格,文件内容如下, 注意: 粘贴时要把每一行末尾的空格去除, 否则运行时会上报说utf8编码错误

```
1 [mysqld]
2 user=mysql
3 character-set-server=utf8
4 [client]
5 default-character-set=utf8
6 [mysql]
7 default-character-set=utf8
8
```

2. 将文件所在的磁盘设为共享磁盘,这样docker才有权限对文件进行读写,方法:启动docker后,点
3. 击桌面右下角docker图标,右键选择settings,在SharedDrives 中勾选文件所在的磁盘,完成后docker需要重启



4. 执行以下命令,磁盘路径(d:/mysql/config/my.cnf)需要根据自己配置文件所在的位置修改

```
1 docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=password123 -e
 MYSQL_ROOT_HOST=% -v e:/mysql/config/my.cnf:/etc/my.cnf --name mysql01
2 mysql/mysql-server
```

4.使用 docker ps 查看是否启动,状态为 healthy 时 才算成功,然后通过以下命令进入容器

```

1 docker exec -it mysql01 bash
2 # 确认配置文件是否正确挂载
3 # 确认root登录密码
4 ``
5
6 password123
7 select user,host from user;

```

8 5,最后可以用navicat 等外部工具测试是否能在外部正常连接docker

| CONTAINER ID | IMAGE              | COMMAND               | CREATED        | STATUS                  | PORTS                             | NAMES   |
|--------------|--------------------|-----------------------|----------------|-------------------------|-----------------------------------|---------|
| f7e018475e3e | mysql/mysql-server | "entrypoint.sh mysql" | 50 seconds ago | Up 49 seconds (healthy) | 0.0.0.0:3306->3306/tcp, 33060/tcp | mysql01 |

```

bash-4.2# cat /etc/my.cnf
[mysqld]
user=mysql
character-set-server=utf8
[client]
default-character-set=utf8
[mysql]
default-character-set=utf8
mysql> select user,host from user;
+-----+-----+
| user | host |
+-----+-----+
root	%
healthchecker	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
+-----+-----+
5 rows in set (0.00 sec)

```

(亲测有效)

所先创建相关文件夹与路径

c:/docker/mysql/config/my.cnf

c:/docker/mysql/data

需要使用官方mysql直接而不是使用mysql/mysql-server版本

```

docker run -d -p 3306:3306 -e MYSQL_USER=sqtest -e
MYSQL_PASSWORD=pwd123 -e MYSQL_ROOT_PASSWORD=pwd123 -e
MYSQL_ROOT_HOST=% -v c:/docker/mysql/config/my.cnf:/etc/my.cnf -v
c:/docker/mysql/data:/var/lib/mysql --name mysql01 mysql

```

外部Volume安装mongo on Cent OS

外部Volume安装mongo on Cent OS //添加外部登录

```

1 docker run -d -p 27017:27017 --name mongo01 -v
 /docker/mongo/config:/data/configdb -v /docker/mongo/db:/data/db mongo --
 auth

```

```

1 docker exec -it mongo01 mongo admin

```

```
1 db.createUser({ user: 'jsmith', pwd: 'some-initial-password', roles: [{
 role: "userAdminAnyDatabase", db: "admin" }] });
```

Windows10挂载Mongo镜像：

已知问题：无法使用Windows路径挂载，改用虚拟磁盘方式OK。

【实测通过】

来源：<http://www.moguf.com/post/windockerrunmongo>

```
1 --创建数据卷
2 docker volume create --name mongodata
3 --挂载数据卷运行mongo数据库
4 docker run --name mongodb -v mongodata:/data/db -p 27017:27017 -d
 mongo:latest [--auth]
6 注：测试使用，不需要账号权限不要加 --auth
8 参数：
9 docker run 运行容器
10 --name mongodb 运行容器的名称为mongodb
11 -v mongodata:/data/db 挂接保存数据的位置，冒号前面是本机（mongodata），后面是虚
 拟机中的映射目录（/data/db）
12 -p 27017:27017 映射端口，前面是本机端口，后面是docker内的端口
13 --auth 授权访问
14 --登录镜像
15 docker exec -it mongodb mongo admin
16 --创建账号
17 db.createUser({
18 user: 'root',
19 pwd: 'admin',
20 roles: [{ role: "userAdminAnyDatabase", db: "admin" }]
21 });
22 --账号授权
23 db.auth("root","admin");
```

## Deploy docker registry

本地运行registry仓库

```
1 docker run -d -p 5000:5000 --restart=always --name registry registry:2
```

本地registry – 可视化UI

```
1 docker run -d --name registry-web --link registry:registry -e
 ENV_DOCKER_REGISTRY_HOST=registry -e ENV_DOCKER_REGISTRY_PORT=5000 -p
 8080:80 konradkleine/docker-registry-frontend:v2
```

带https证书（请注意我们host上放证书的文件叫cert, not certs）

```

1 docker run -d \
2 --restart=always \
3 --name registry \
4 -v `pwd`/cert:/certs \
5 -e REGISTRY_HTTP_ADDR=0.0.0.0:443 \
6 -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/1_hub.jessetalk.cn_bundle.crt \
7 -e REGISTRY_HTTP_TLS_KEY=/certs/2_hub.jessetalk.cn.key \
8 -p 443:443 \
9 registry:2

```

### 创建密码文件

```

1 mkdir auth
2 docker run \
3 --entrypoint htpasswd \
4 registry:2 -Bbn testuser testpassword > auth/htpasswd

```

### Base Authentication 用户名和密码

```

1 docker run -d \
2 --restart=always \
3 --name registry \
4 -v `pwd`/auth:/auth \
5 -e "REGISTRY_AUTH=htpasswd" \
6 -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
7 -e "REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd" \
8 -v `pwd`/cert:/certs \
9 -e REGISTRY_HTTP_ADDR=0.0.0.0:443 \
10 -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/1_hub.jessetalk.cn_bundle.crt \
11 -e REGISTRY_HTTP_TLS_KEY=/certs/2_hub.jessetalk.cn.key \
12 -p 443:443 \
13 registry:2
14

```

gitlab : 基础docker 安装的地址

<https://docs.gitlab.com/omnibus/docker/>

提升gitlab-ci runner permission

```

1 sudo usermod -aG docker gitlab-runner

```

Kitematic 图形化管理Docker容器与镜像:

```

1 下载地址: https://kitematic.com/
2 解压缩复制到: C:\Program Files\Docker
3 启动: docker鲸鱼图标-->右键-->Kitematic
4 使用Docker账号登录
5 1 直接查看容器运行信息

```

