

docker宿主通信问题

解决办法：固定容器ip并暴露容器端口

其本质是在主机的指定端口和容器的指定端口形成一个映射关系，当访问宿主机指定端口时，请求就会被转发到容器的指定端口上。实际上并没有完全解决宿主机通信问题。但主机之间的通信本身就是以端口来进行的。理论上只要暴露足够多的端口，就可以实现日常所需的所有通信需求。

1. 首先在docker里创建一个新的网桥。

```
docker network create --subnet=172.30.0.0/16 docker01
```

subnet 为指定的网段，可以自行指定，但要注意，默认网段是172.17.0.0/16，与新建网段会无法通信。docker01则是网桥名

2. 在使用镜像生成容器时，添加 --network --ip -p 选项 语法格式为：

```
docker run -itd --network=docker01 --ip 172.30.0.6 -p <主机端口>:<映射端口> ....
```

如果想要暴露多个端口，就加多个 -p：

```
docker run -itd -p <主机端口1>:<映射端口1> -p <主机端口2>:<映射端口2> ....
```

推荐至少要暴露这3个端口：21、22、3306。分别对应ftp、ssh、mysql服务。如果打算在容器内装多个mysql实例，则看情况增加。

例：

```
docker run --privileged=true --network=docker01 --name mycentos7 --ip 172.30.0.7 -itd -p 3376:3306 -p 7021:21 -p 7022:22 -v /sys/fs/cgroup:/sys/fs/cgroup centos:v1 /usr/sbin/init
```

注：这里的端口映射要注意，window下貌似只有65535个端口，所以映射端口要在这个范围内。不然会报如下错误

```
PS E:\Docker> docker run --privileged=true --network=docker01 --name mycentos7 --ip 172.30.0.7 -itd -p 73306:3306 -p 7021:21 -p 7022:22 -v /sys/fs/cgroup:/sys/fs/cgroup centos:v1 /usr/sbin/init
C:\Program Files\Docker\Docker\resources\bin\docker.exe: invalid publish opts format (should be name=value but got '73306:3306').
See 'C:\Program Files\Docker\Docker\resources\bin\docker.exe run --help'.
PS E:\Docker>
```

3. 如果生成容器时没有暴露端口，想要暴露端口，有两种方法：

- a. 把容器打包提交成镜像，再用新的镜像加上 -p 指令重新生成容器。能保留配置，但耗时较长。
- b. 修改配置文件。

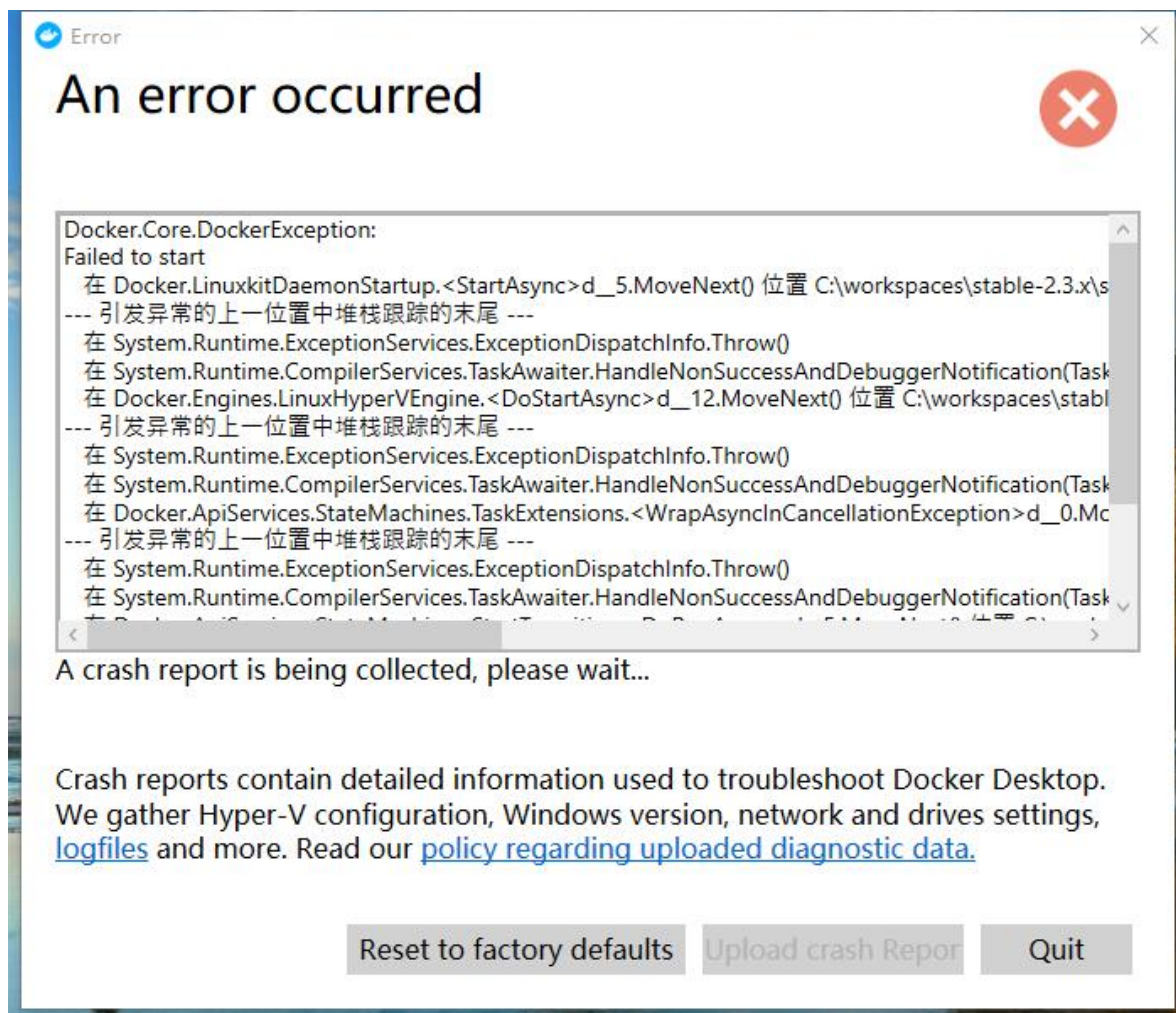
① linux下：/var/lib/docker/containers/[hash_of_the_container]/hostconfig.json 以及 config.v2.json

我宿主机不是linux的，所以没试。

② windowx下：daemon.json

追加 "hosts": ["tcp://0.0.0.0:3306"] 配置

注：windows下的配置是微软的官方文档里给的。按理说应该不会有问题。但我自己尝试的时候发现改了配置docker就会无法启动。可能是我自己的解读有误。



附上文档[链接](https://docs.microsoft.com/zh-cn/virtualization/windowscontainers/manage-docker/configure-docker-daemon)，可以自行阅读尝试

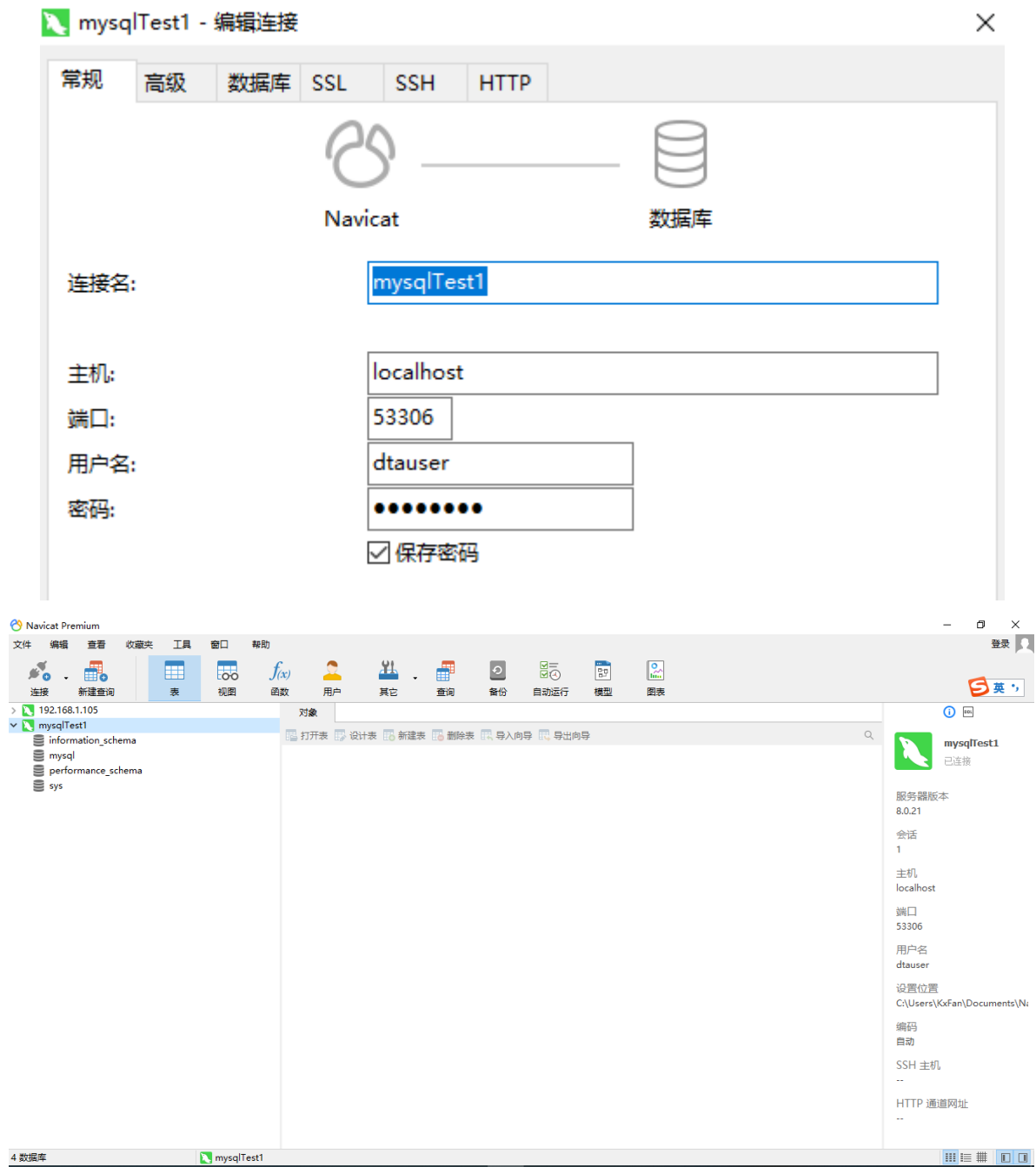
<https://docs.microsoft.com/zh-cn/virtualization/windowscontainers/manage-docker/configure-docker-daemon>

本地连接：

这里创建了一个暴露21、22、3306三个端口的容器，分别对应主机的50021、50022、53306端口。启动容器并开启mysql服务和ssh服务。

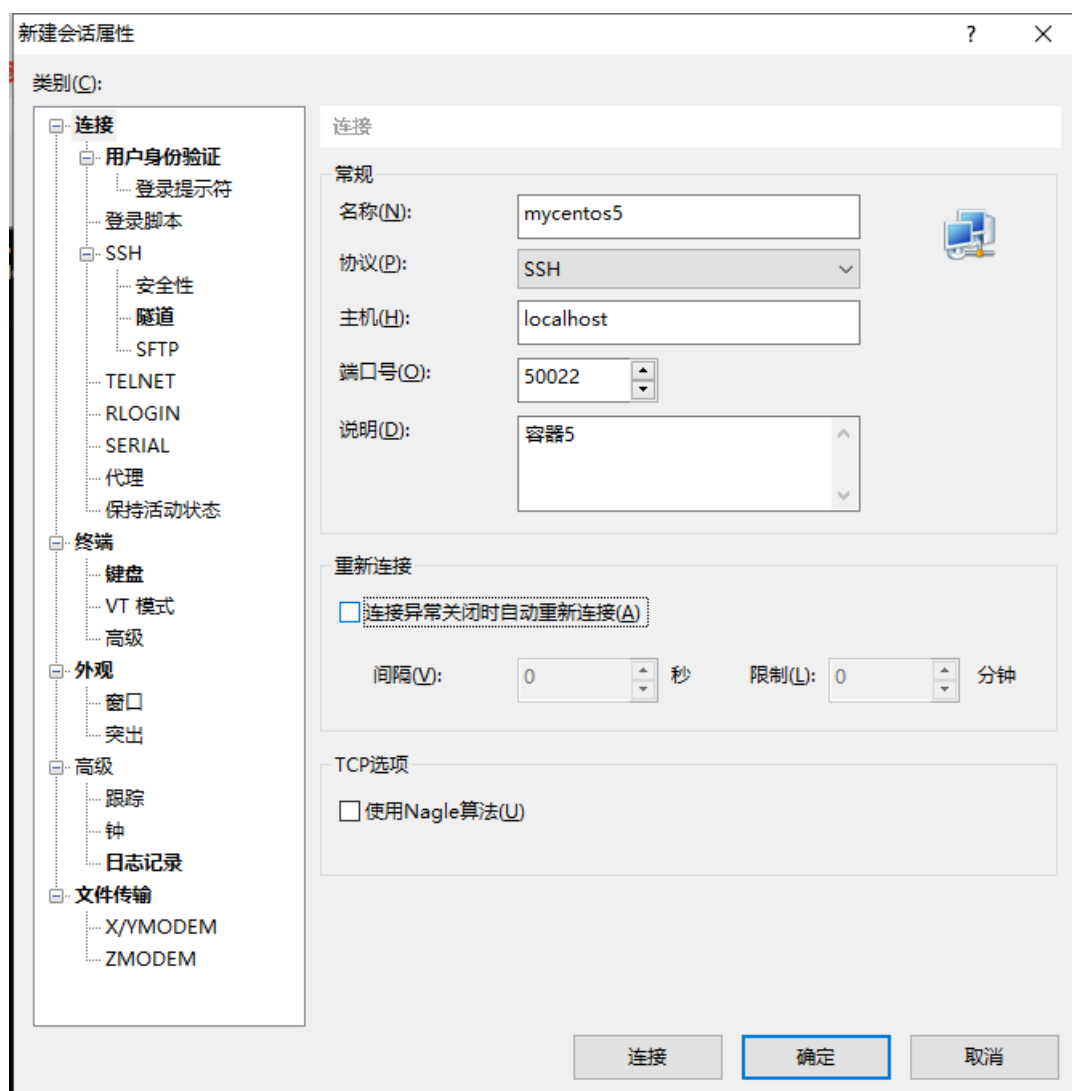
首先在宿主机上启动Navicat，尝试连接mysql服务。

这里主机选Localhost，端口为指定的本地端口，用户名和密码则是容器mysql服务可支持远程登录的用户。

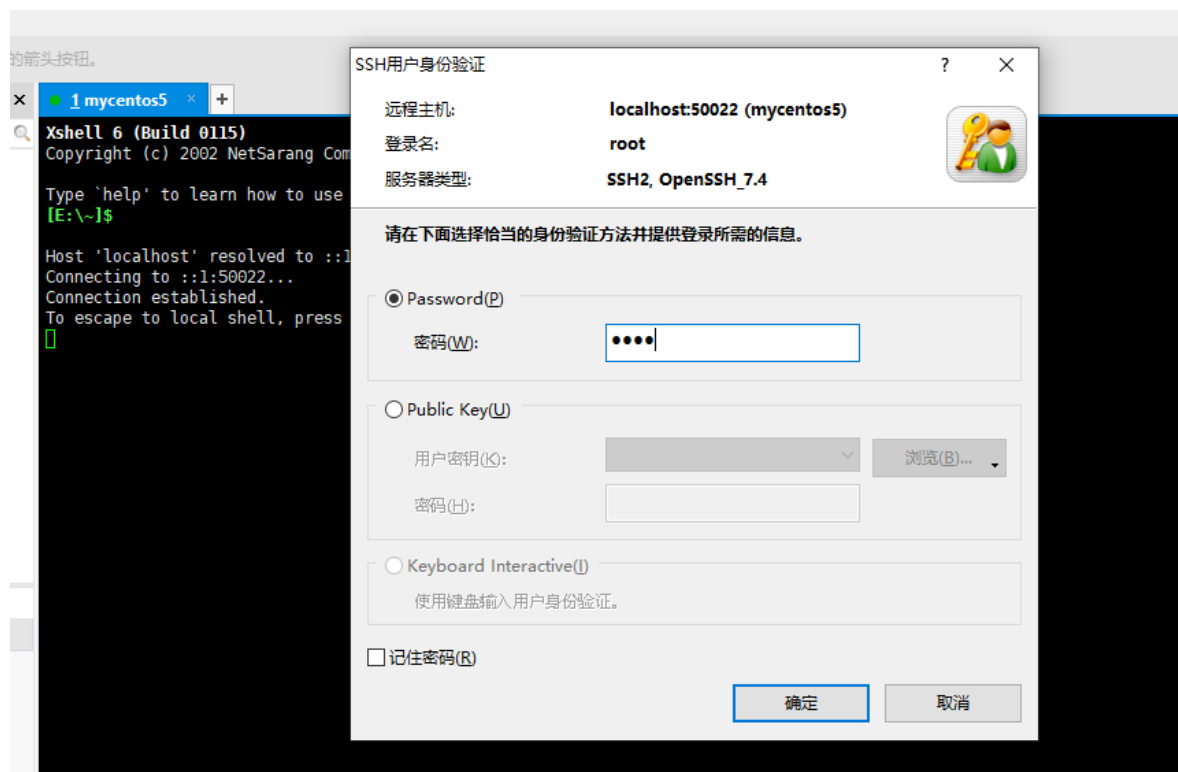


连接成功，接下来尝试用Xshell连接容器

本地主机，端口号50022



在弹出窗口输入用户名和密码 (root,root)。同样连接成功。



存在的问题：

正如开始所说，这种方式并不是实质性的解决通信问题的。只是在指定端口形成了映射关系。

- 在某次重启电脑后尝试开启容器时报了如下错误：（忘记截图了）

Error response from daemon: Ports are not available: listen tcp 0.0.0.0:50021: bind: An attempt was made to access a socket in a way forbidden by its access permissions.

大概是说50021这个端口无效。这种情况在重启docker后可能会出现。

解决措施：以管理员身份开一个cmd窗口

先检查对应端口是否有程序占用

```
netstat -ano | findstr ":50021"
```

一般情况是没有被占用的。这时候检查下tcp协议排除的端口范围

```
netsh interface ipv4 show excludedportrange protocol=tcp
```

```
C:\WINDOWS\system32>netsh interface ipv4 show excludedportrange protocol=tcp
协议 tcp 端口排除范围
开始端口  结束端口
-----
      8391      8391
      9001      9001
     50463     50562
     50563     50662
     50663     50762
     50763     50862
     51128     51227
* - 管理的端口排除。
```

看错误信息中的端口是否在此范围中。（我这是解决后的截图，当时50021是在范围中的）

如果确定端口在排除范围的话。要开启对应端口

首先关闭Hyper-V服务，会提示重启电脑

```
dism.exe /Online /Disable-Feature:Microsoft-Hyper-V
```

重启后开启对应端口（重启后docker启动会失败，不用管）

```
# start 起始端口 num 表示可用端口数 按自己的需求来
netsh int ipv4 set dynamicport tcp start=50000 num=4000

# 排除ipv4动态端口占用 startport 起始端口 numberofports 端口数
netsh int ipv4 add excludedportrange protocol=tcp startport=50000
numberofports=4000
```

然后开启Hyper-V服务，再次重启电脑

```
dism.exe /Online /Enable-Feature:Microsoft-Hyper-V /All
```

这时候就可以正常开启容器了。