

Please visit website: <http://cxyroad.com>

解决Docker容器内无法访问外网问题

=====

解决Docker容器内无法访问外网问题

=====

> tips: 由于每个人环境不同，问题的解决方法也自然不同，下面是小编所在环境的解决方式记录下来分享给各位，如果觉得文章不错，欢迎点赞评论

...

```
{ \_ / }  
( ^ ) ~  
| |
```

...

法一：重建网络`docker0`

...

```
sudo service docker stop  
sudo pkill docker  
sudo iptables -t nat -F  
sudo ifconfig docker0 down  
sudo brctl delbr docker0  
sudo service docker start
```

...

法二：开启宿主机的ipv4转发功能

> 针对解决`WARNING: IPv4 forwarding is disabled. Networking will not work.`

...

修改配置

```
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
```

```
# 重启network  
systemctl restart network
```

```
# 查看 (0->标识未开启 1->标识开启)  
sysctl net.ipv4.ip_forward  
# net.ipv4.ip_forward = 1
```

```
# 重启docker  
systemctl restart docker
```

```
...
```

```
#### 法三：使用`--net=host`宿主机网络方式启动容器
```

```
...
```

```
# 示例  
docker run --net=host --name ubuntu_bash -i -t ubuntu:latest  
/bin/bash
```

```
docker run --net=host --rm alpine ping -c 5 baidu.com
```

```
...
```

```
#### 法四：关闭SELinux
```

```
...
```

```
# 查看SELinux状态  
getenforce
```

```
# 临时关闭SELinux  
setenforce 0
```

```
# 永久关闭SELinux  
vim /etc/selinux/config  
# 将 `SELINUX=enforcing` 改成 `SELINUX=disabled`
```

```
# 重启linux  
reboot
```

```
...
```

```
#### 方法五
```

> 参考 [github.com/coolsnowwol...](http://cxyroad.com/"https://github.com/coolsnowwolf/lede/issues/1760")

...

修改配置

```
echo "net.bridge.bridge-nf-call-ip6tables = 1" >> /etc/sysctl.conf
echo "net.bridge.bridge-nf-call-iptables = 1" >> /etc/sysctl.conf
echo "net.bridge.bridge-nf-call-arptables = 1" >> /etc/sysctl.conf
```

```
# Luci > 网络 > 防火墙 > 转发: 接受
# Luci > 状态 > 防火墙 > 重启防火墙
```

```
service docker restart
```

...

方法六

...

```
rm -rf /var/lib/docker/network/*
```

```
systemctl restart docker
```

...

方法七: 修改DNS客户端解析文件`resolv.conf`

...

```
# 写入Linux的DNS客户端解析文件resolv.conf里
#           114.114.114.114 => 国内移动、电信和联通通用的DNS
#           8.8.8.8 => google提供, 更适合国外以及访问国外网站的用户使
用
# echo nameserver 8.8.8.8 > /etc/resolv.conf
echo nameserver 114.114.114.114 > /etc/resolv.conf
```

查看配置

```
cat /etc/resolv.conf
```

测试

```
docker run --rm alpine ping -c 5 baidu.com
```

```
...
```

```
----
```

> 上面是解决宿主机DNS问题，下面为容器内部DNS与宿主机DNS不一致问题，一般情况下应该不存在此情况。

修改容器内部的DNS解析配置

```
...
```

1、进入容器

```
docker exec -it 容器ID/容器名 /bin/bash
```

2、先查看DNS配置

```
cat /etc/resolv.conf
```

配置文件如下

```
nameserver 127.0.0.11
```

```
options timeout:2 attempts:3 rotate single-request-reopen ndots:0
```

3、修改配置 => 注释`options timeout:2 attempts:3 rotate single-request-reopen ndots:0`即可

```
nameserver 127.0.0.11
```

```
# options timeout:2 attempts:3 rotate single-request-reopen ndots:0
```

如果没有vi命令可尝试使用echo

```
echo "nameserver 127.0.0.11" > /etc/resolv.conf
```

4、测试

```
ping baidu.com
```

```
...
```

> 注：网上有人说程序重启后，需手动修改，根本的解决办法如下

> 这一步小编未测试！！

```
...
```

```
# 注释其中的内容 `options timeout:2 attempts:3 rotate single-request-  
reopen ndots:0`  
# 原因为: `/etc/resolvconf/resolv.conf.d/tail`文件会默认覆盖  
`/etc/resolv.conf` 中的最后一行配置。  
vim /etc/resolvconf/resolv.conf.d/tail
```

```
# 使修改生效  
sudo resolvconf -u
```

```
...
```

方法八：和网络工程师沟通下是否做了一定限制

```
...
```

```
# 先运行一个访问外网的程序  
docker run --rm alpine ping -c 50 baidu.com
```

```
# 安装tcpdump  
yum install tcpdump  
# 利用tcpdump进行抓包分析  
tcpdump -i docker0 icmp  
# 发现有request包, 表明本机到baidu的包, baidu是接收到的, 可能是百度没  
响应 (可能性不大) 或者被公司防火墙阻断了
```

```
...
```

方法九：宿主机防火墙开启伪装IP功能

> tips: 感觉无用, 小编直接关防火墙也没解决网络问题

```
...
```

```
# 查看防火墙状态  
firewall-cmd --state  
# 查看防火墙是否开启ip地址转发 (ip地址伪装)  
firewall-cmd --query-masquerade  
# 开启ip地址转发  
firewall-cmd --add-masquerade --permanent  
# 将网络接口 docker0 加入 trusted zone, 解决 DNS 问题  
firewall-cmd --permanent --zone=trusted --add-interface=docker0  
# 更新防火墙规则  
firewall-cmd --reload
```

...

方法十：重装docker

> 此方式乃是最后无奈之举了...

临时测试容器内能够ping

...

```
docker run --rm alpine ping -c 5 baidu.com
```

...

小编个人问题记录

基于`电信网`环境，在局域网window上安装centos7.6系统，宿主机能正常访问外网，但容器内不行，且容器内dns解析有问题；
如果直接访问ip的话，`tcpdump`抓包`docker0`发现只有请求，无任何响应。

临时使用`--net=host`方式解决外网访问问题，还待持续研究！

原文链接: <https://juejin.cn/post/7361720687956901899>