# docker DNS相关

## 概述

容器在启动时默认会使用宿主机的域名服务器配置文件(resolve.conf)，会检查里面的域名服务器地址，过滤掉本地地址（127.0.0.1），如果过滤完后没有其他服务器地址，则自动添加Google的域名服务器地址作为容器的域名服务器地址，结果就出现了每次启动都是8.8.8.8的服务器地址。所以**只要在宿主机的配置文件中新增你需要添加的服务器地址，然后重启下容器，容器中的域名服务器都会自动更新**。经过验证，果然如此，重启下容器后，容器中的域名服务器配置都更新正确了。

## 容器的dns解析顺序

容器中的DNS名称解析优先级顺序为：

* 内置DNS服务器127.0.0.11。

* 通过--dns等参数为容器配置的DNS服务器。

* docker守护进程的--dns服务配置（默认为8.8.8.8和8.8.4.4）

* 宿主机上的DNS设置。

Docker如何为每个容器提供主机名和DNS配置，而无需在构建自定义Docker镜像时在内部写入主机名？它的诀窍是利用可以写入新信息的虚拟文件，在容器内覆盖三个关键的/etc 文件。 你可以通过在一个容器中运行mount 来看到这一点：

* hosts文件不能在image里面修改，是因为这个文件是通过动态mount进Docker image的

* DNS，默认从宿主主机上获取DNS，所以，如果修改好宿主的DNS，在其上启动的Docker image的DNS便会是修改好的DNS。

root@278abf1de44c:/app# mount|grep etc

/dev/mapper/centos-root on /etc/resolv.conf type xfs (rw,relatime,attr2,inode64,noquota)

/dev/mapper/centos-root on /etc/hostname type xfs (rw,relatime,attr2,inode64,noquota)

/dev/mapper/centos-root on /etc/hosts type xfs (rw,relatime,attr2,inode64,noquota)

## 修改容器dns

cat /etc/docker/daemon.json   
{  
 "dns" : [  
 "114.114.114.114",  
 "8.8.8.8"  
 ]  
}  
sudo service docker restart