

# 通过NFS共享数据

在部分公司已经在使用NFS共享文件了。如果应用部署在docker容器里，如果能访问这些共享的文件呢？

docker不支持NFS开箱即用，而在每个容器中安装NFS客户端以便能挂载远程目录不是一个好方法。建议的方法是使用一个docker容器充当其它容器访问NFS共享数据的代理。

## 问题

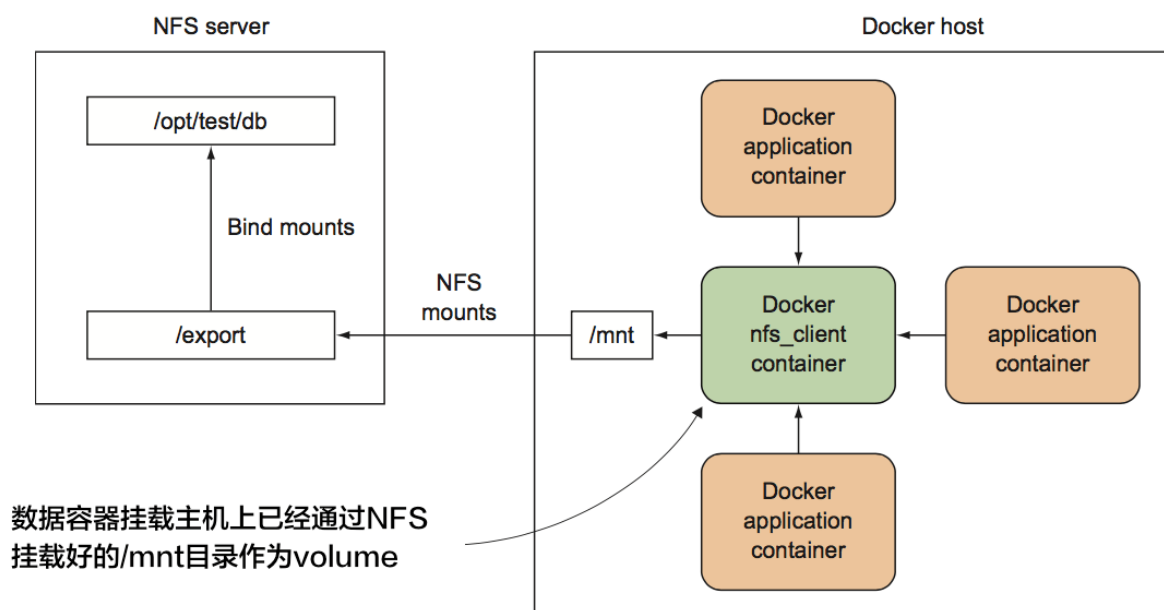
你想无缝访问NFS的远程文件系统

## 解决方法

使用数据容器代理访问

## 讨论

数据容器的相对介绍请参考<https://www.centos.bz/2016/11/docker-data-container/>。下图显示这种方法如何工作。



sharing-data-with-nfs-pic1

NFS服务器暴露内部目录为/export，此目录与/opt/test/db绑定挂载。然后docker主机使用NFS协议挂载到目录/mnt。然后数据容器挂载主机的/mnt来代理其它容器对共享目录的访问。

虽然NFS的介绍超过了本文的范畴，但NFS与本文的主题密切相关，还是有必要说明下。

假设你想共享系统为Ubuntu14.04的/opt/test/db目录给其它主机的容器。  
切换为root权限，安装NFS服务器并创建一个export目录：

```
4. # apt-get install nfs-kernel-server
5. # mkdir /export
6. # chmod 777 /export
```

现在绑定挂载db目录到/export目录：

```
2. # mount --bind /opt/test/db /export
```

你现在应该能在/export目录访问/opt/test/db的内容。

如果你想让重启后绑定还生效，添加/opt/test/db /export none bind 0 0  
到/etc/fstab文件。

现在添加如下行到/etc/exports文件：

```
2. /export 127.0.0.1(ro,fsid=0,insecure,no_subtree_check,async)
```

因为我们只用了一台机器做试验，即nfs服务器与容器都在一台，所以这里填写了  
127.0.0.1，如果是在其它机器，记得把127.0.0.1替换为nfs客户端IP。

考虑到安全，我们在这里只读挂载，不过你也可以把ro替换为rw设置读写挂载，如果  
这样做，记得在async选项后添加no\_root\_squash。

然后通过NFS挂载/export目录到/mnt目录：

```
4. # mount -t nfs 127.0.0.1:/export /mnt
5. # exportfs -a
6. # service nfs-kernel-server restart
```

现在运行一个数据容器：

```
2. # docker run -ti --name nfs_client --privileged -v /mnt:/mnt busybox
/bin/true
```

然后其它容器关联此数据容器的volumes:

```
7. # docker run -ti --volumes-from nfs_client debian /bin/bash
8. root@079d70f79d84:/# ls /mnt
9. myb
10. root@079d70f79d84:/# cd /mnt
11. root@079d70f79d84:/mnt# touch asd
12. touch: cannot touch `asd': Read-only file system
```