我们部署containers需要containers之间可以交流，例如连接container可以与另一个充当data-store的container通信。

**通过Link**

Link相当于在两个container之间拉一条线进行通信. 连接以后我们可以看到container中的环境变量env, 以及container中/etc/hosts已经根据我们给出连接设置好。并且可以在本container中ping通data-store container.

流程：

docker run -d --name redis-server redis # 从image redis中创建一个名为redis-server的container

docker run --link redis-server:redis alpine # 从名为aplpine的image中创建container，并且创建时候通过link进行连接。连接的对象为名为redis-server的容器，并且在本容器中该对象别名为redis.连接以后，在本容器的env中会看见关于连接的目的容器的环境变量，以别名\_开头。

**通过network**

通过docker构建一个network,如果container设置连接这个network,那么可以与这个network上的所有container进行通信。可以将container attach或者detach. 当container加入network后，并不会像Link设置那样，设置环境变量以及/etc/hosts.而是Docker实现内置一个DNS服务器，修改container的/etc/resolve.conf.

Network的设置比Link设置更为灵活，我们可以为一个container连接多个network, 也可以一个network中加入多个container.

流程：

docker network create backend-network # 构建一个名字为backend-network的network

docker run --net=backend-network alpine # 从名为alpine的image中启动container加入backend-network网络

docker network create frontend-network2 # 构建一个名字为frontend-network的network

docker network connect --alias db frontend-network2 redis # 将名为redis的container，加入frontend-network2并且设置该container的网络别名为db