Docker的网络模式以及区别，使用场景？

1. Bridge模式

bridge模式是默认模式，此模式会为每一个容器分配Network Namespace、 设置IP等， 并将一个主机上的Docker容器连接到一个虚拟网桥上。此模式与外界通信使用NAT协议

2. Host模式

基于宿主机的网络，它的网络跟宿主机的网络一模一样，那是因为，在创建这个容器之初、并没有对它的Net网络栈进行隔离，而是直接使用的宿主机的网络，此模式没有网络隔离性， 同时会引起网络资源的竞争与冲突。

使用场景：性能较好，但不便之处是灵活性不高，此模式没有网络隔离性，容器与宿主机出现端口冲突问题。

3. Container模式

这个模式指定新创建的容器和已经存在的一个容器共享一个Network Namespace，而不是和宿主机共享，新创建的容器不会创建自己的网卡，配置自己的IP，而是和一个指定的容器共享IP、端口范围等

使用场景：在于可以将一个应用的多个组件放在不同的容器趾， 这些容器配成container模式的网络， 这样它们可以作为一个整体对外提供服务。 同时， 这种模式也降低了容器间的隔离性。

4. None模式

什么都没有的网络，Docker容器拥有自己的Network Namespace， 但是， 并不为Docker容器进行任何网络配置。 也就是说， 这个Docker容器没有网卡、 IP、 路由等信息。 需要我们自己为Docker容器添加网卡、 配置IP等，给用户很高的自由度来配置网络环境

使用场景：此网络可以运行一些关于安全方面的验证码、效验码等服务

简要描述AUFS文件系统的特点以及优缺点，如何解决性能问题，如何实现数据持久化（如数据库重启后数据不丢失），并在作业中体现

Docker有两种方法实现数据持久化，一种是volumn，一种是bind mount。在这两种方法中，docker推荐使用的是第一种。。。。to be continue