1. Dcoker数据存储
2. Docker镜像容器特性

Docker镜像是由多个只读层叠加而成的，在启动容器时，Docker只会加载只读镜像层并在镜像栈的顶部添加一个读写层。如果当前运行的容器修改了一个已经存在的镜像文件，那么该文件将会从读写层下面的只读层复制到读写层，该文件原有的只读版本依然存在，只是已经被读写层中该文件的修改副本所隐藏了，即写时复制机制。

（二） 容器数据存在的问题

Docker容器从启动到删除，数据都是保存在读写层中的，如果我们只是stop容器或者restart容器，那么容器中的数据都是不会丢失的，但是如果我们remove容器，那么该容器的改变数据将会丢失。同时，读写层中的数据存在于联合文件系统（ufs）中，不利于宿主机访问，且容器间数据共享不方便。

（三）数据卷

数据卷是容器上的目录，这类目录可以绕过ufs系统，与宿主机上的某个目录绑定，写入容器的数据都会写入到宿主机的绑定目录中，完成数据的持久化。数据卷可以在运行容器时完成创建和绑定操作，当然需要提前声明绑定的对应关系。

（四） 数据卷的分类

1. 绑定卷
2. 概述

人为进行操作，输入绑定参数来关联目录。绑定卷可以使用宿主机中的任何目录。

1. 绑定数据卷

通过启动容器时指定-v，并且将宿主机目录和容器中目录进行绑定即可：

docker run -it –name roc -v HOSTDIR:VOLUMEDIR foc

可以看到，宿主机的目录在前，容器中数据卷在后，且这种绑定方式的优先级大于Docker容器绑定方式。

1. Docker管理卷
2. 概述

同样需要人为设置，但是一旦容器启动后，容器就会接管，会自己创建目录与宿主机中的目录挂载。Docker管理卷的目录是指定的，如/var/lib/docker/volumes/…，如果我们设定容器的数据卷为Docker管理卷，那么它一启动后就会将容器中的某一目录绑定到宿主机的指定目录中。

1. 设置数据卷

我们可以通过Docker容器的DockerFile启动脚本来设置数据卷绑定规则，如下所示：

VOLUME “/data”

表示启动Dockers管理卷，将Docker容器中的data目录绑定到宿主机中的/var/lib/docker/volumes/…中的某一个目录下，该目录下有/\_data目录，就是对应容器中/data目录的内容，即使容器被删除了宿主机中的目录也不会被删除。

1. 删除数据卷

可以手动在宿主机中删除，也可以在删除容器时删除，只要指定 -v 即可删除：

docker rm -f -v test2

1. 联合卷

如果我们想要让两个容器之间通信，可以借助宿主机的某个目录作为中转站实现，即每个容器都和宿主机中的指定目录进行绑定，那么容器之间就能够实现通信了。若是不想使用宿主机中的目录，如何实现呢？

这就需要联合卷了，联合卷是让一个容器直接和另一个容器通信，共享数据。这个卷的前提是每个容器都是使用Dockers管理卷自动管理的，只有每个容器提前指定好了容器中哪个目录需要绑定，那么另一个容器在启动时直接指定 –volumes-from 容器名 就可以与这个容器共享数据了，如下所示：

docker run -it –name roc –volomes-from ContainerName roc

代表着数据卷从ContainerName这个容器中来的，这样两个容器在DockerFile中指定的数据卷就是共享的了。

（五） 存储驱动

1. 概述

Docker存储驱动是Docker的核心组件，是Docker实现分层镜像的基础。

1. 版本

现有的存储驱动有三种：

1. device mapper(DM)

性能和稳定性存在问题，不推荐生产使用。

1. btrfs

社区实现的btrfs driver ，同样是稳定性和性能问题。

1. overlayfs

内核3.18以上版本，overlayfs进入了主线，性能和稳定性优异，最佳选择。

1. overlayfs架构

对应于Docker容器中的架构，镜像层对应最底层，容器读写层对应中间层，容器挂载层对应整合层。最底层和中间层的数据都会出现在整合层中，每个层级的目录挂载命令如下所示：

mount -t overlay -olowerdir=./low,upperdir=./upper,workdir=./work ./merged