

Docker 中包括三个基本的概念：

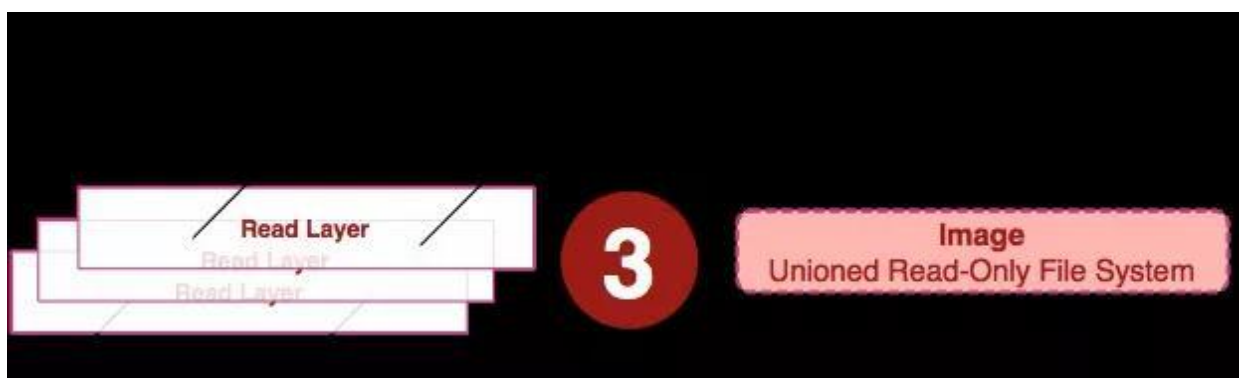
1. Image(镜像)
2. Container(容器)
3. Repository(仓库)

镜像是 Docker 运行容器的前提，仓库是存放镜像的场所，可见镜像更是 Docker 的核心。

Image(镜像)

那么镜像到底是什么呢?Docker 镜像可以看作是一个特殊的文件系统，除了提供容器运行时所需的程序、库、资源、配置等文件外，还包含了一些为运行时准备的一些配置参数(如匿名卷、环境变量、用户等)。

镜像不包含任何动态数据，其内容在构建之后也不会被改变。镜像(Image)就是一堆只读层(read-only layer)的统一视角，也许这个定义有些难以理解，下面的这张图能够帮助读者理解镜像的定义：



从左边我们看到了多个只读层，它们重叠在一起。除了最下面一层，其他层都会有一个指针指向下一层。这些层是 Docker 内部的实现细节，并且能够在主机的文件系统中访问到。

统一文件系统(Union File System)技术能够将不同的层整合成一个文件系统，为这些层提供了一个统一的视角。

这样就隐藏了多层的存在，在用户的角度来看，只存在一个文件系统。我们可以在图片的右边看到这个视角的形式。

Container(容器)

容器(Container)的定义和镜像(Image)几乎一模一样，也是一堆层的统一视角，唯一区别在于容器的最上面那一层是可读可写的。



由于容器的定义并没有提及是否要运行容器，所以实际上，容器 = 镜像 + 读写层。

Repository(仓库)

Docker 仓库是集中存放镜像文件的场所。镜像构建完成后，可以很容易的在当前宿主上运行。

但是，如果需要在其他服务器上使用这个镜像，我们就需要一个集中的存储、分发镜像的服务，Docker Registry(仓库注册服务器)就是这样的服务。

有时候会把仓库(Repository)和仓库注册服务器(Registry)混为一谈，并不严格区分。

Docker 仓库的概念跟 Git 类似，注册服务器可以理解为 GitHub 这样的托管服务。

实际上，一个 Docker Registry 中可以包含多个仓库(Repository)，每个仓库可以包含多个标签(Tag)，每个标签对应着一个镜像。

所以说，镜像仓库是 Docker 用来集中存放镜像文件的地方，类似于我们之前常用的代码仓库。

通常，一个仓库会包含同一个软件不同版本的镜像，而标签就常用于对应该软件的各个版本。

我们可以通过 <仓库名>:<标签> 的格式来指定具体是这个软件哪个版本的镜像。如果不给出标签，将以 Latest 作为默认标签。

仓库又可以分为两种形式：

Public(公有仓库)

Private(私有仓库)

Docker Registry 公有仓库是开放给用户使用、允许用户管理镜像的 Registry 服务。

一般这类公开服务允许用户免费上传、下载公开的镜像，并可能提供收费服务供用户管理私有镜像。

除了使用公开服务外，用户还可以在本地搭建私有 Docker Registry。Docker 官方提供了 Docker Registry 镜像，可以直接使用做为私有 Registry 服务。

当用户创建了自己的镜像之后就可以使用 Push 命令将它上传到公有或者私有仓库，这样下次在另外一台机器上使用这个镜像时候，只需要从仓库上 Pull 下来就可以了。