4 docker镜像讲解

## 4.1 docker 镜像是什么

镜像事一种轻量级、可执行的独立软件包，用来打包软件运行环境和基本运行镜像开发的软件，它包含运行软件所需的所有内容，包括代码、运行时所需的库、环境变量和配置文件。

所有的应用程序，直接打包docker镜像，就可以直接跑起来。

如何得到镜像：

* 从远程仓库下载
* 朋友拷贝
* 自己制作一个dockerFile

## 4.2 docker 镜像加载原理

### 联合文件系统

UnionFS( 联合文件系统)是一种种分层、轻量级并且高性能的文件系统，它支持对文件系统的修改作为一次提交来一层层的叠加,同时可以将不同目录挂载到同一个虚拟文件系统下lunite severaldirectoriesinto a single virtuafilesystem)。Union 文件系统是 Docker 镜像的基础。镜像可以通过分层来进行继承，基于基础镜像(没有父镜像 )，可以制作各种具体的应用镜像。

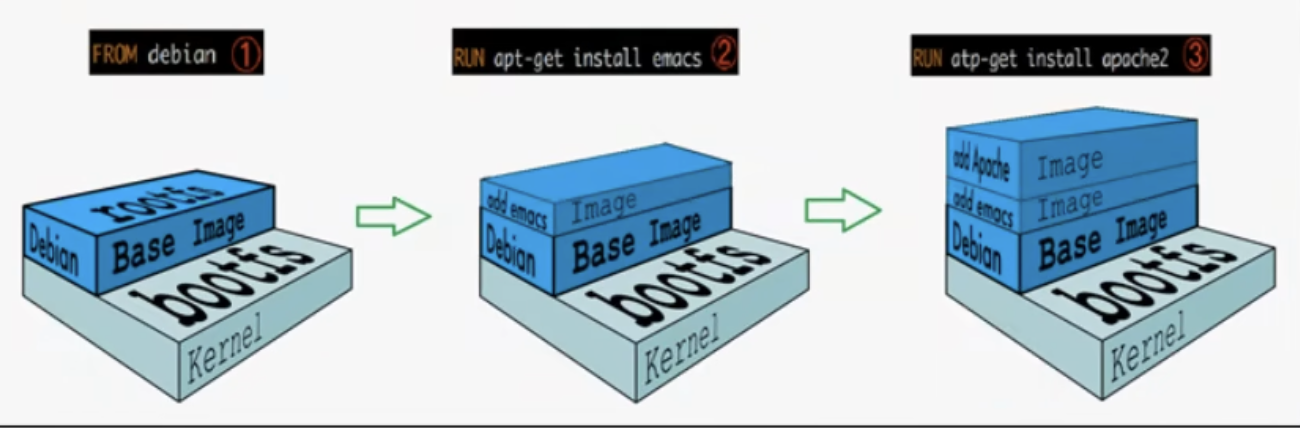
特性:一次同时加载多个文件系统，但从外面看起来，只能看到一个文件系统，联合加载会把各层文件系统叠加起来，这样最终的文件系统会包含所有底层的文件和目录

### Docker镜像加载原理

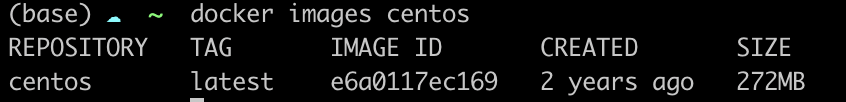
docker的镜像实际上由一层一层的文件系统组成，这种层级的文件系统UnionFS。

bootfs(boot file system)庄要包含bootloader和kernel, bootloader主要是引导加载kernel, Linux刚启动时会加载bootfs文件系统，在Docker镜像的最底层是bootfs。这一层与我们典型的Linux/Unix系统是一样的，包含boot加载器和内核。当boot加载完成之后整个内核就都在内存中了，此时内存的使用权已由bootfs转交给内核，此时系统也会卸载bootfs。

rootfs (root file system)，在bootfs之上，包含的就是典型 Linux 系统中的/dey,/proc,/bin,/ec等标准目录和文件。rootfs就是各种不同的操作系统发行版，比如Ubuntu，Centos等等。



平时安装的Centos都是几个G，为什么docker里的只有200多M?



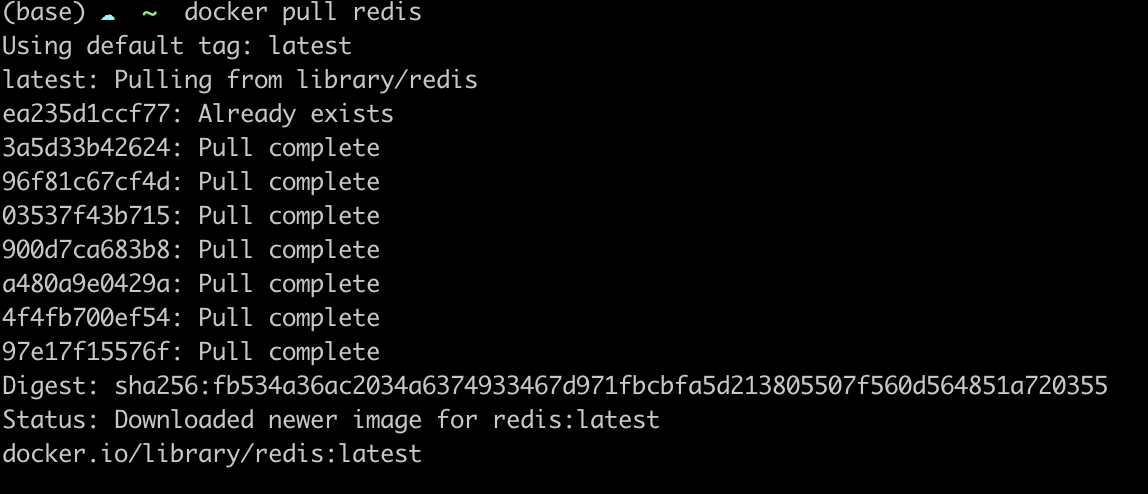
对于一个精简的OS，rootfs 可以很小，只需要包含最基本的命令，工具和程序库就可以了，因为底层直接用Host的kernel，自己只需要提供rootfs就可以了。由此可见对于不同的linux发行版,bootfs基本是一致的,rootfs会有差别,因此不同的发行版可以公用bootfs。

虚拟机是分钟级别，容器是秒级！

## 4.3 分层理解

### 分层的镜像

我们下载一个较大的镜像时，通过观察日志输出看到，镜像是一层一层下载的！



如果某一层在之前下载过，该层就不再下载。

思考: 为什么Docker镜像要采用这种分层的结构呢 ?

我觉得，最大的好处莫过于是资源共享了!

比如有多个镜像都从相同的Base镜像构建而来，那么宿主机只需在磁盘上保留一份base镜像，同时内存中也只需要加载一份base镜像，这样就可以为所有的容器服务了，而且镜像的每一层都可以被共享.

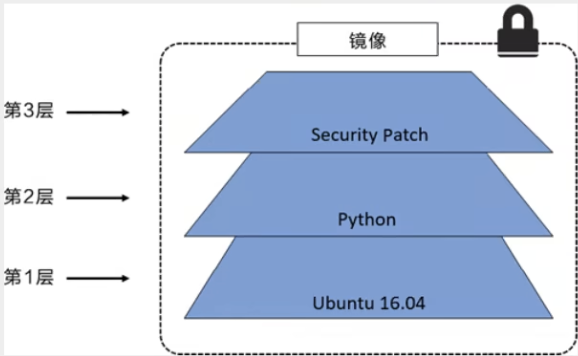
查看镜像分层的方式可以通过 docker image inspect 命令!

**理解：**

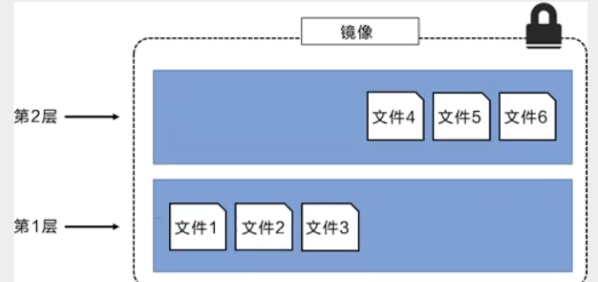
所有的 Docker:

镜像都起始于一个基础镜像层，当进行修改或增加新的内容时，就会在当前镜像层之上，创建新的镜像层。

举一个简单的例子，假如基于 Ubuntu Linux 16.04创建一个新的镜像，这就是新镜像的第一层；如果在该镜像中添加 Python包就会在基础镜像层之上创建第二个镜像层；如果继续添加一个安全补丁，就会创建第三个镜像层。该镜像当前已经包含 3 个镜像层，如下图所示(这只是一个用于演示的很简单的例子)。

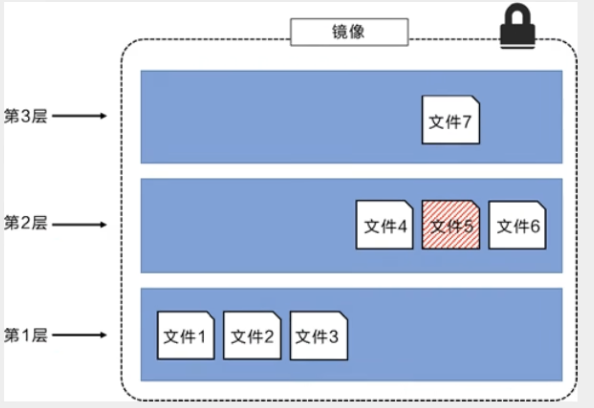


在添加额外的镜像层的同时，镜像始终保持是当前所有像的组合，理解这一点非常重要。下图中举了一个简单的例子，每个镜像层包含 3 个文件，而镜像包含了来自两个镜像层的 6 个文件。



上图中的镜像层跟之前图中的略有区别，主要目的是便于展示文件。

下图中展示了一个稍微复杂的三层镜像，在外部看来整个镜像只有6 个文件，这是因为最上层中的文件7是文件 5的一个更新版本。



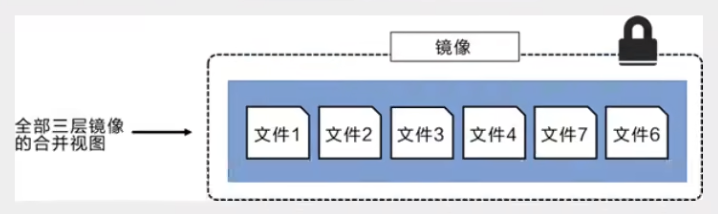
这种情况下，上层镜像层中的文件覆盖了底层镜像层中的文件。这样就使得文件的更新版本作为一个新镜像层添加到镜像当中。

Docker 通过存储引擎(新版本采用快照机制)的方式来实现镜像层堆栈，并保证多镜像层对外展示为统一的文件系统。

Linux 上可用的存街|擎有AUFS、Overlay2、Device Mapper、Btrfs 以及 ZFS，顾名思义，每种存引|擎都基于Linux 中对应的文件系统或者块设备技术，并且每种存储引擎都有其独有的性能特点。

Docker在Windows 上仅支持windowsfilter一种存储引擎，该引擎基于NTFS 文件系统之上实现了分层和 CoW[1].

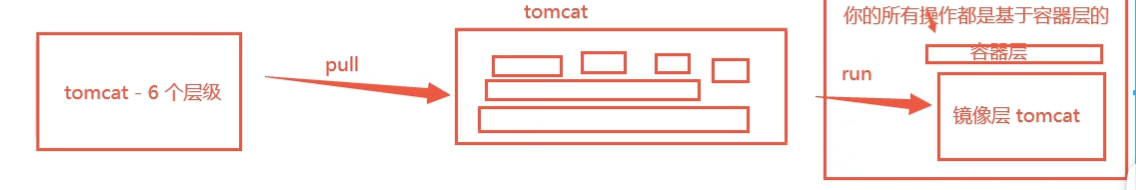
下图展示了与系统显示相同的三层镜像。所有镜像层堆叠并合并，对外提供统一的视图。



### 特点

Docker镜像都是只读的，当容器启动时，一个新的可写层被加载到镜像的顶部!

这一层就是我们通常说的容器层，容器之下的都叫镜像层!



## 4.4 commit 镜像

如何提交一个自己的镜像

实战测试

学习方式：先理解概念，但是一定要实践，最后实践与理论相结合

如果想保存当前容器的状态，就可以通过commit来提交一个新的镜像。

到此，我们已经入门docker！