## Docker镜像加载原理

#### UFS联合文件系统：

是一种分层、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。Union文件系统是Docker镜像的基础，镜像可以通过分层来继承，基于基础镜像，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加载原理

docker的镜像实际上由一层一层的文件系统组成，UFS。

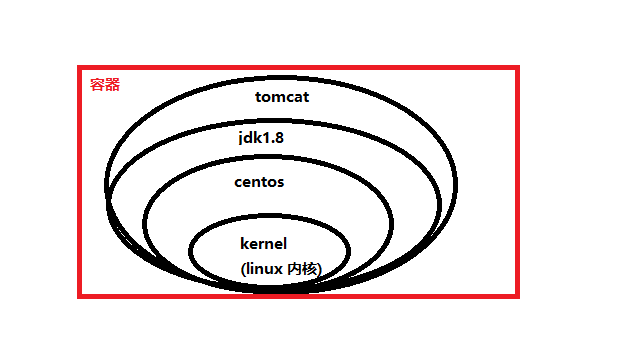
bootfs(boot file system) 主要包含bootloader和kernel，bootloader 主要是引导加载kernel，Linux刚启动时会加载bootfs文件系统，在Docker镜像的最底层是bootfs。这一层与我们典型的Linux/Unix系统是一样的，包含boot加载器和内核。当boot加载完成之后整个内核就存在内存中了，此时内存的使用权已由bootfs转交给内核，此时系统也会卸载bootfs。

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

#### 为什么镜像很小？

对于一个精简的OS，rootfs可以很小，只需要包括最基本的命令、工具和程序库就可以了，因为底层直接用Host（宿主机）的kernel，自己只需要提供rootfs就行了，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。

eg.tomcat镜像为例



#### 分层理解

采用这种分层结构最大的一个好处就是**共享资源**，比如有多个镜像都从相同的base镜像构建而来，那么宿主机只需要在磁盘上保存一份base镜像，同时内存中也只需要加载一份base镜像，就可以为所有容器服务了。而且镜像的每一层都可以被共享。

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。这一层通常被称作 “**容器层**” ，“容器层” 之下的都叫**镜像层**。