

## 2.3.4 镜像、仓库管理

讲师：汪洋





# 目录

1

Docker 镜像特性

2

DockerFile

3

Docker 仓库构建



# 1

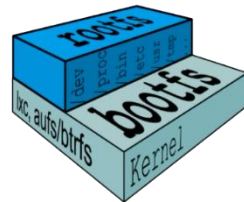
## Docker 镜像管理



容器创建时需要指定镜像，每个镜像都由唯一的标示 Image ID，和容器的 Container ID 一样，默认 128 位，可以使用前 16 为缩略形式，也可以使用镜像名与版本号两部分组合唯一标示，如果省略版本号，默认使用最新版本标签（latest）

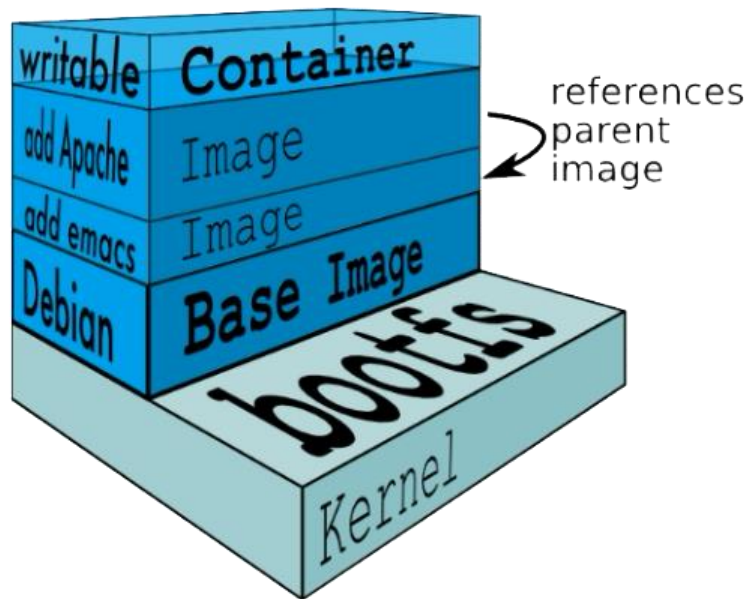
镜像的分层：Docker 的镜像通过联合文件系统（union filesystem）将各层文件系统叠加在一起

- bootfs：用于系统引导的文件系统，包括 bootloader 和 kernel，容器启动完成后会被卸载以节省内存资源
- rootfs：位于 bootfs 之上，表现为 Docker 容器的跟文件系统
  - >> 传统模式中，系统启动时，内核挂载 rootfs 时会首先将其挂载为“只读”模式，完整性自检完成后将其挂载为读写模式
  - >> Docker 中，rootfs 由内核挂载为“只读”模式，而后通过 UFS 技术挂载一个“可写”层





- 已有的分层只能读不能修改
- 上层镜像优先级大于底层镜像





2

DockerFile



1、容器 > 镜像: `docker commit CID xx.xx.xx`

# 工作在前台的守护进程至少一个

## 2、DockerFile

Dockfile 是一种被 Docker 程序解释的脚本，Dockerfile 由一条一条的指令组成，每条指令对应 Linux 下面的一条命令。Docker 程序将这些 Dockerfile 指令翻译真正的 Linux 命令。Dockerfile 有自己书写格式和支持的命令，Docker 程序解决这些命令间的依赖关系，类似于 Makefile。Docker 程序将读取 Dockerfile，根据指令生成定制的 image

生成命令: `docker build -t wangyang/jdk-tomcat .`

[打开 DockerFile 说明](#)



导出: `docker save -o xx.xx.xx xx.xx.xx.tar`

导入: `docker load -i xx.xx.xx.tar`





# 3

## Docker 仓库构建



## 1、官方仓库构建

## 2、Harbor 构建



0 v e r