

# DOCKERFILE

Dockerfile的基本用法

Dockerfile优化实战

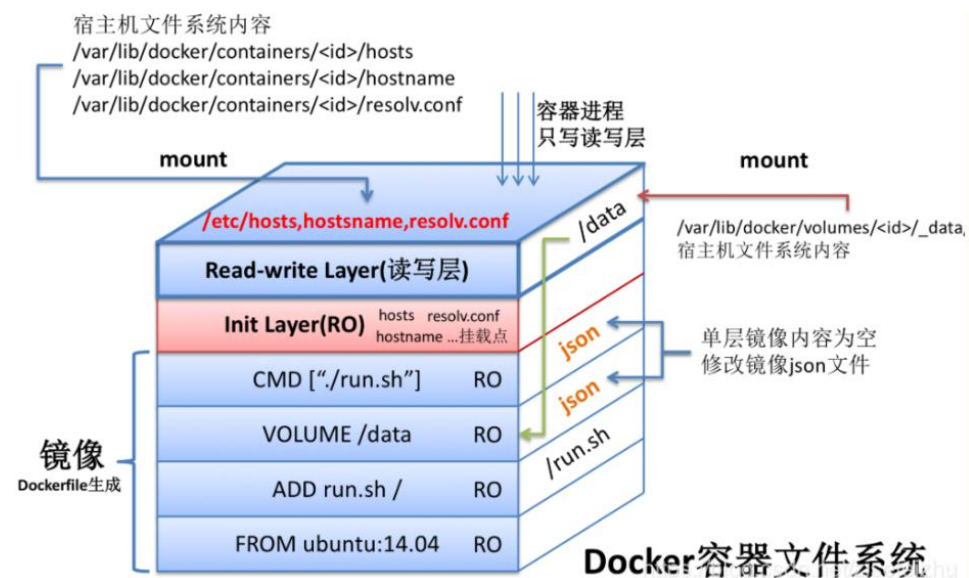
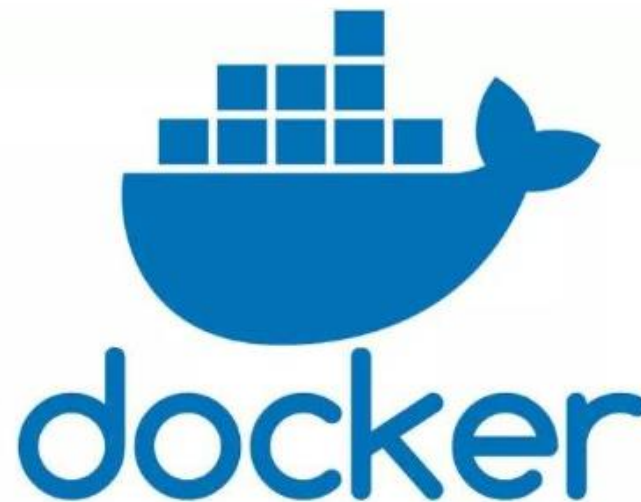
主讲人：林坤

## 有关内容简介

一句话概括Docker

一句话概括镜像

一句话概括容器



## 引出dockerfile

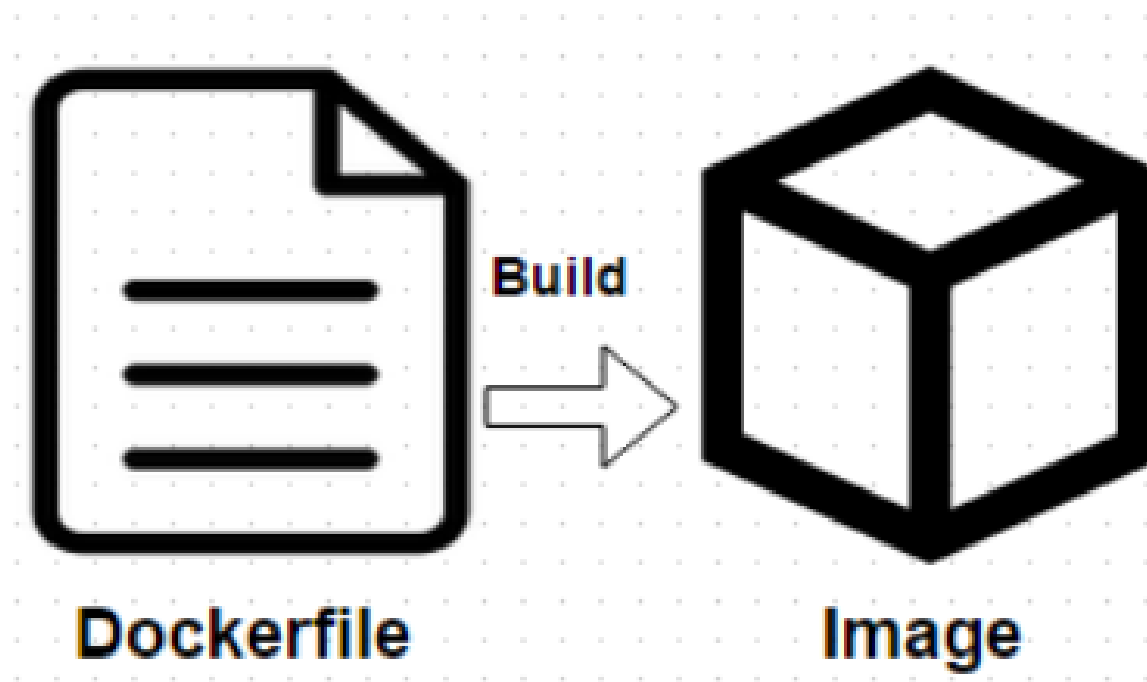
为什么官方镜像不适合使用

修改官方镜像的几种方案的优缺点

1) exec --> vi --> reload

2) 存储卷

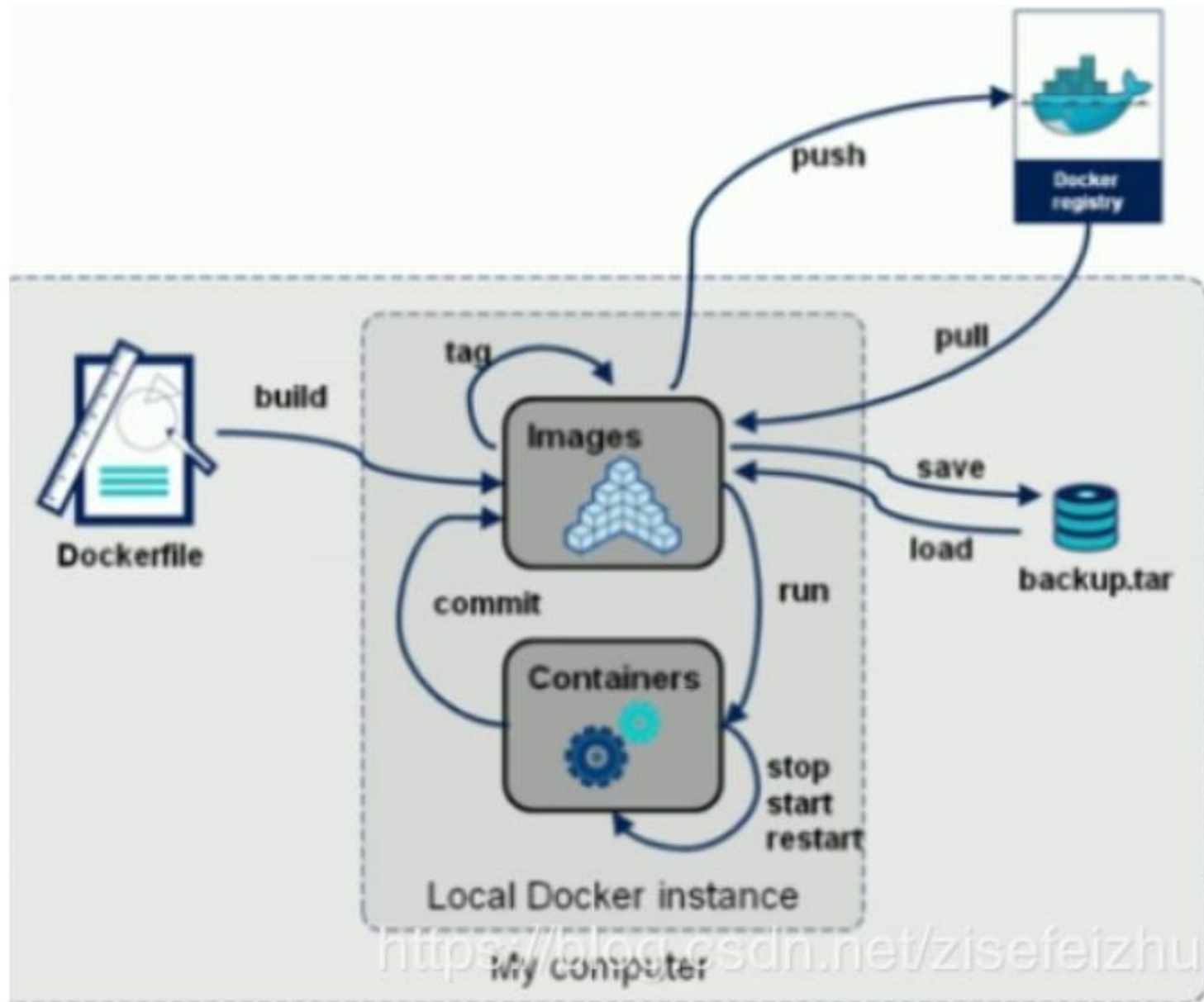
3) commit





# 什么是dockerfile

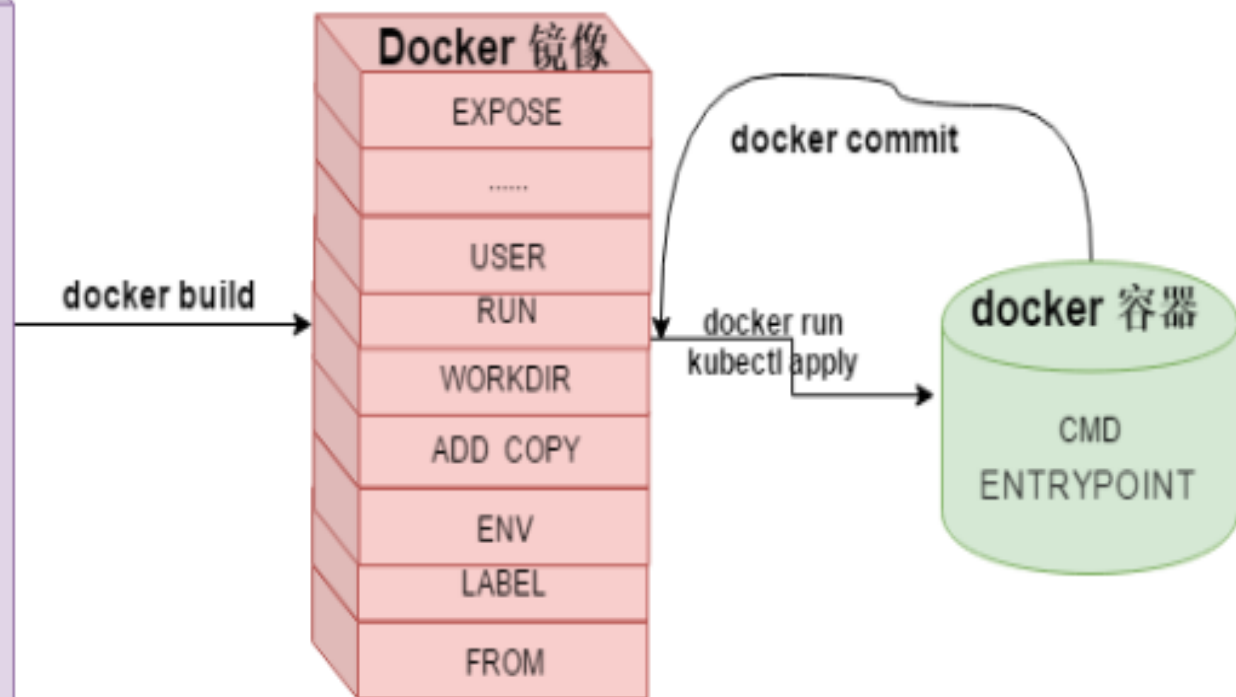
Dockerfile: 相当于是一个文档, 用户可以基于dockerfile生成新的容器。dockerfile仅仅是用来制作镜像的源码文件, 是构建容器过程中的指令, docker能够读取dockerfile的指定进行自动构建容器, 基于dockerfile制作镜像, 每一个指令都会创建一个镜像层, 即镜像都是多层叠加而成, 因此, 层越多, 效率越低, 创建镜像, 层越少越好。因此能在一个指令完成的动作尽量通过一个指令定义。



# dockerfile指令集

## Dockerfile 常见指令

FROM: 设置镜像使用的基础镜像  
LABEL: 设置镜像标签  
ENV: 设置容器的环境变量  
COPY: 编译镜像时复制文件到镜像中  
ADD: 编译时复制文件到镜像中  
WORKDIR: 设置RUN CMD ENTRYPOINT COPY ADD指令的工作目录  
VOLUME: 设置容器的挂在卷  
RUN: 编译镜像时运行的脚本  
CMD: 设置容器的启动命令  
ENTRYPOINT: 设置容器的入口程序  
USER: 设置运行RUN CMD ENTRYPOINT的用户名  
HEALTHCHECK: 检查容器健康状态  
SHELL: 设置容器执行命令  
EXPOSE: 设置镜像暴露的端口  
STOPSIGNAL: 设置容器的退出信号里



# dockerfile优化实战

1. 明智的选择基础镜像
2. 减少镜像的大小，减少依赖，仅安装需要的软件包
3. 减少镜像的图层
4. 一个镜像只做一件事
5. 合理的使用指令
6. 清理镜像构建的中间产物



谢谢!