

Docker自定义镜像

教学内容

- 第一节 镜像的内部结构
- 第二节 通过Dockerfile构建镜像
- 第三节 制作Java应用镜像示例



为什么要自定义镜像

对于Docker用户来说，最好的情况是不需要自己创建镜像。几乎所有常用的数据库、中间件、应用软件等都有现成的Docker官方镜像或其他人和组织创建的镜像，我们只需要稍作配置就可以直接使用。

某些情况下我们也不得不自己构建镜像，比如：

- 找不到现成的镜像，比如自己开发的应用程序。
- 需要在镜像中加入特定的功能。



镜像的内部结构

如果只是使用镜像，当然不需要了解镜像的内部结构，直接通过docker命令下载和运行就可以了，但如果我们想创建自己的镜像，或者想理解Docker为什么是轻量级的，就非常有必要学习这部分知识了。

- 下载官方hello-world镜像：docker pull hello-world
- 运行：docker run hello-world



Dockerfile

- Dockerfile是镜像的描述文件，定义了如何构建Docker镜像,hello-world的Dockerfile:

```
FROM scratch  
COPY hello /  
CMD ["/hello"]
```

- FROM scratch: 表示镜像从白手起家，从0开始构建。
- COPY hello/将文件 “hello” 复制到镜像的根目录。
- CMD[“/hello”]容器启动时，执行/hello。
- 镜像hello-world中就只有有一个可执行文件 “hello” ， 其功能就是打印出 “Hello from Docker” 等信息。



base镜像

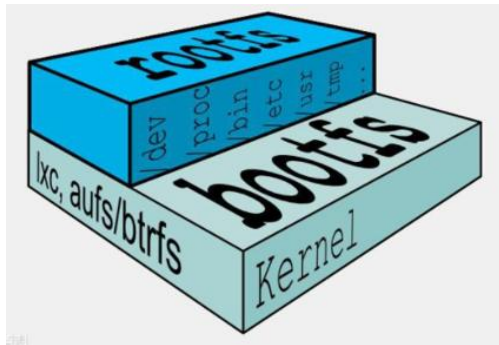
- hello-world虽然是一个完整的镜像，但它并没有什么实际用途。
- 通常来说，我们希望镜像能提供一个基本的操作系统环境，用户可以根据需要安装和配置软件。这样的镜像我们称作base镜像。
- base镜像有两层含义：（1）不依赖其他镜像，从scratch构建；（2）其他镜像可以以之为基础进行扩展。
- 能称作base镜像的通常都是各种Linux发行版的Docker镜像，比如Ubuntu、Debian、CentOS等。
- 下载CentOS镜像： `docker pull centos`



为何系统镜像如此之小

Linux操作系统由内核空间和用户空间组成

- 内核空间是kernel, Linux刚启动时会加载bootfs文件系统, 之后bootfs会被卸载掉。
- 用户空间的文件系统是rootfs, 包含我们熟悉的 /dev、/proc、/bin等目录。
- 对于base镜像来说, 底层直接用Host的kernel, 自己只需要提供rootfs就行了。
- 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。相比其他Linux发行版, CentOS的rootfs已经算臃肿的了, alpine还不到10MB。



CentOS的Dockerfile

base镜像提供的是最小安装的Linux发行版，CentOS镜像的Dockerfile的内容

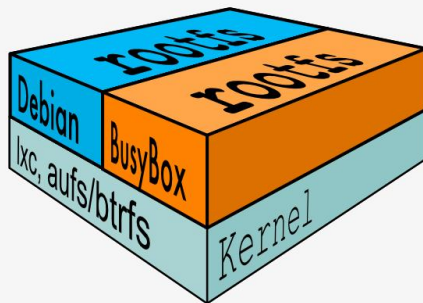
```
FROM scratch
ADD centos-7-x86_64-docker.tar.xz /
CMD ["/bin/bash"]
```

- ADD指令添加到镜像的tar包就是CentOS 7的rootfs。在制作镜像时，这个tar包会自动解压到/目录下，生成/dev、/proc、/bin等目录。
- 可在Docker Hub的镜像描述页面中查看Dockerfile。



Docker支持运行多种Linux OS

- 不同Linux发行版的区别主要就是rootfs。
- 比如Ubuntu 14.04使用upstart管理服务，apt管理软件包；而CentOS 7使用systemd和yum。这些都是用户空间上的区别，Linux kernel差别不大。
- 所以Docker可以同时支持多种Linux镜像，模拟出多种操作系统环境。
- Debian和BusyBox（一种嵌入式Linux）上层提供各自的rootfs，底层共用Docker Host的kernel。



构建镜像

FROM	指定基础镜像	FROM centos:6
ENV	设置环境变量，可在后面指令使用	ENV key value
COPY	拷贝本地文件到镜像的指定目录	COPY ./mysql-5.7.rpm /tmp
RUN	执行Linux的shell命令，一般是安装过程的命令	RUN yum install gcc
EXPOSE	指定容器运行时监听的端口，是给镜像使用者看的	EXPOSE 8080
ENTRYPOINT	镜像中应用的启动命令，容器运行时调用	ENTRYPOINT java -jar xx.jar

构建Java应用镜像

参考操作文档

