

接下来由我来为大家演示如何构建一个docker镜像，构建镜像最重要是写一个dockerfile。那么什么是dockerfile呢？这是官网中一段关于dockerfile的介绍，第一句是说 Docker可以通过Dockerfile中的指令来自动构建一个镜像。我在这为大家列举了其中常用的指令。

FROM 命令是指定当前镜像基于哪个镜像。比如你的项目基于java，就可以是FROM java:8, 如果基于python，那么就是from python:3.6

WORKDIR 是指定接下来的 `shell` 语句运行在哪个路径下。当我们进入容器时默认会在这个路径下。比如COPY命令，如果是拷贝到镜像中的当前路径，那么就是指拷贝到WORKDIR指定的路径。

COPY 把宿主机某个目录下文件拷贝到镜像中。

RUN命令是构建镜像时运行的命令，比如我们的项目是一个python项目，而某些包需要通过pip install来下载，我们就可以在这里来进行下载。就像 RUN PIP install pytorch等

CMD是启动容器时才会运行的指令。我们希望我们的项目在容器启动时就自动运行，就需要在这里加入对应的启动命令。CMD命令只能有一个，如果有多个的话，那么只有最后一个会生效。

ADD功能和 COPY 相同，都是进行文件拷贝。但是使用ADD还可以进行**网络资源**的拷贝，还自带**解压**。

一般使用 COPY 就够了。

ENTRYPOINT 和 CMD功能类似，它们可以根据后面命令形式不同进行搭配使用，这是官网中关于搭配使用的结果演示。

EXPOSE是暴露容器内端口。可以配合docker run中的参数-P使用，将容器内端口和主机端口进行映射。比如下面这张图，我们访问主机的32776端口其实就是访问这个容器里的80端口。

VOLUME指定挂载的文件。一般是匿名挂载

`VOLUME ["/data1","/data2"]` 将 `"/data1","/data2"` 这两个文件夹挂载到宿主机上

接下来的暂时可以不用介绍，然后进行自建镜像演示。

我准备了一个springboot项目，首先确保它能通过命令运行，这是在9090端口运行

。。。运行成功

这个就是刚刚那个jar包。

接下来编写dockerfile文件

首先它是基于java: 8 注意所有指令都必须是大写

我们可以指定工作路径是 /app

然后我们把dockerfile所在路径下的项目文件拷贝过去

可以通过CMD命令来修改它项目的运行端口为9091

然后我们开放容器内9091端口

最后就是希望容器启动时运行我们的项目

