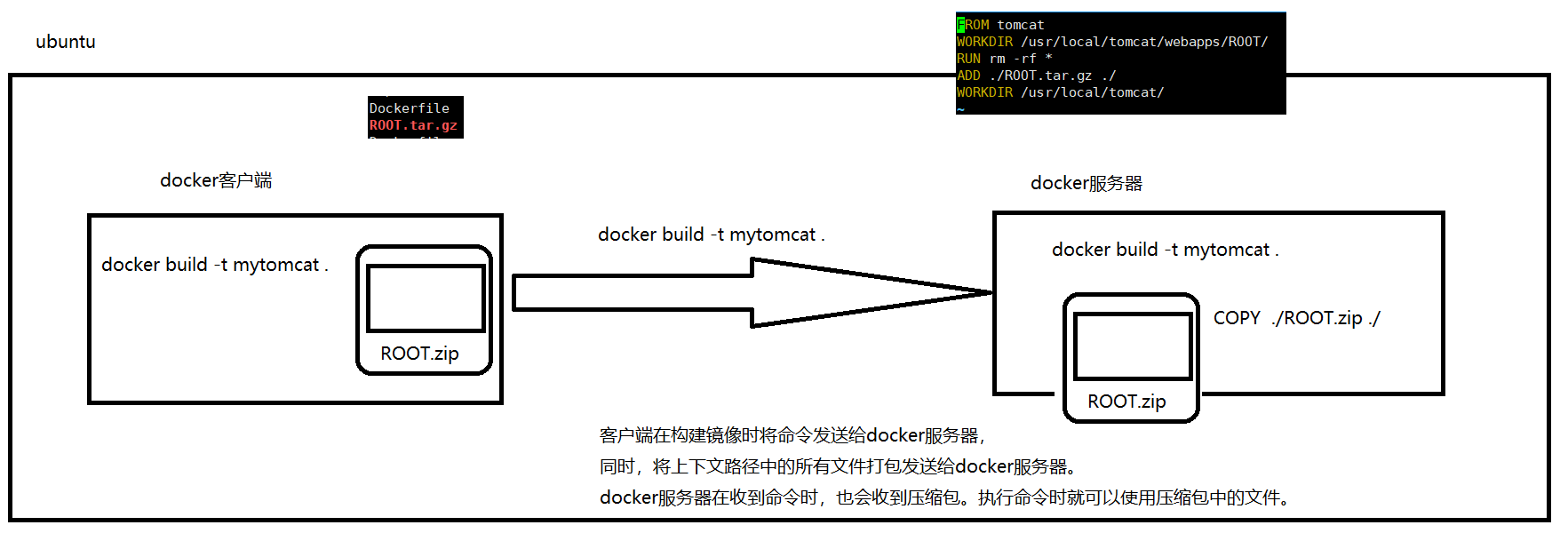
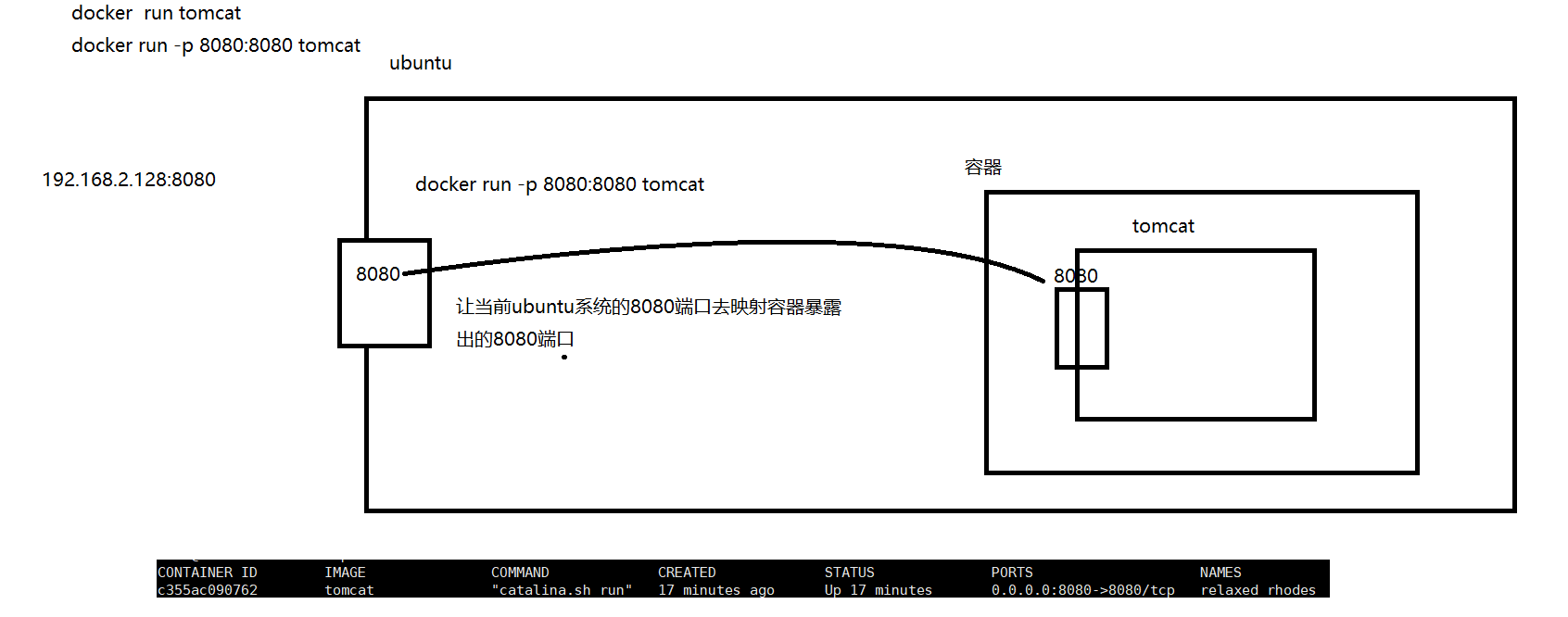
# 一、镜像的上下文



# 二、Dockerfile指令

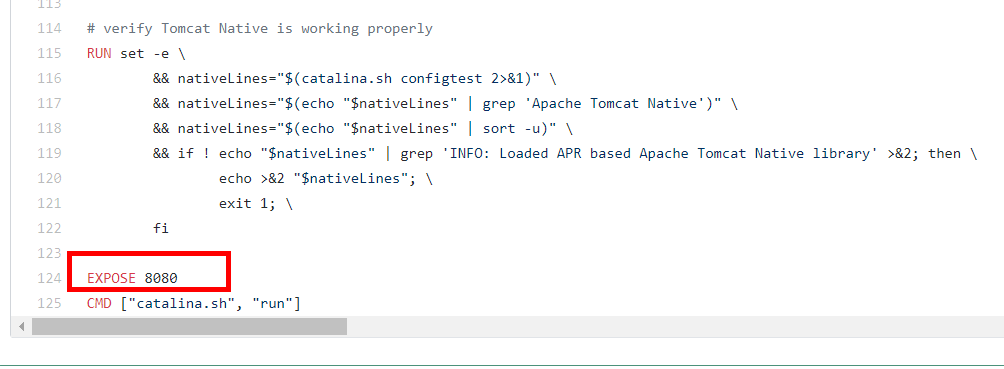
1.EXPOSE

让当前的镜像（容器）暴露出指定的端口，该端口可以被host主机所映射



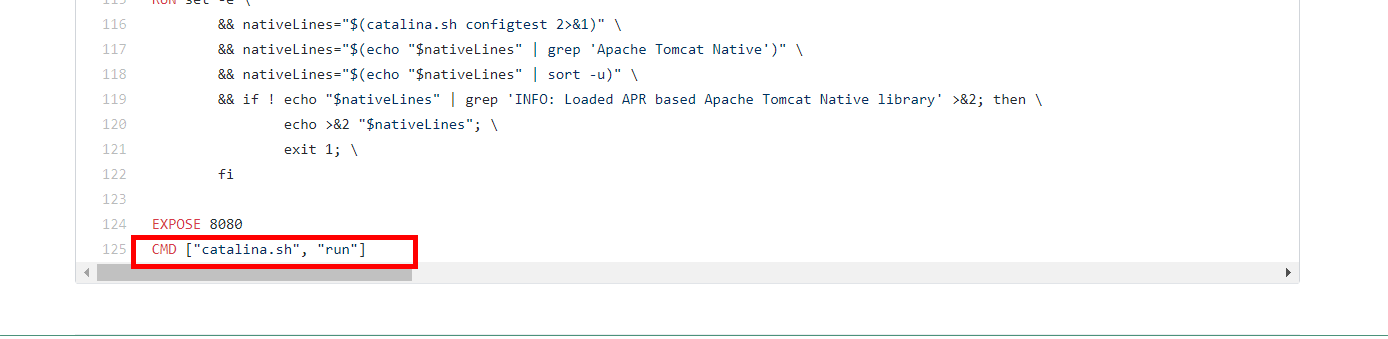
docker run -p 80:8080 tomcat

用host主机的80端口去映射tomcat容器暴露出的8080端口，那么就可以通过主机ip：80访问到该tomcat容器。



2.CMD

让容器直接启动，通过cmd指令



3.ENV 设置环境变量

4.ENTRYPOINT

ENTRYPOINT 和 CMD指令是同一个目的，都是用于设置启动容器时的参数，但是CMD只能执行一次，而ENTRYPOINT可以执行多次！

5.VOLUME

数据卷，为容器指定数据卷。

6.FROM

7.RUN

8.WORKDIR

9.ADD

10.COPY

# 三、管理容器

## 1.创建容器

docker run tomcat

创建一个没有端口映射的容器，外部浏览器访问不到该容器

docker run -p 8080:8080 tomcat

创建一个主机8080端口映射到容器暴露出的8080端口的容器，通过本机浏览器的8080端口访问该容器

docker run -P tomcat

创建一个主机默认的32768端口映射到容器暴露出的8080端口的容器，通过本机浏览器的32768端口访问该容器

## 2. 创建容器带名字

docker run -p 8080:8080 --name tomcat1 tomcat

## 3. 创建容器，容器以守护态的方式运行，不会占用主进程

docker run -p 8080:8080 --name tomcat2 -d tomcat

## 4.停止一个容器

docker stop 容器id(或者容器名)

## 5.查看当前正在运行的容器

docker ps

## 6.查看当前所有的容器，包括正在运行的和已停止的

docker ps -a

## 7.删除容器

docker rm 容器id(或者容器名)

注意： 如果容器正在运行。则需要先停止，再删除

## 8.快速的停止所有容器

docker stop $(docker ps)

将搜索出的所有的正在运行的容器，进行停止

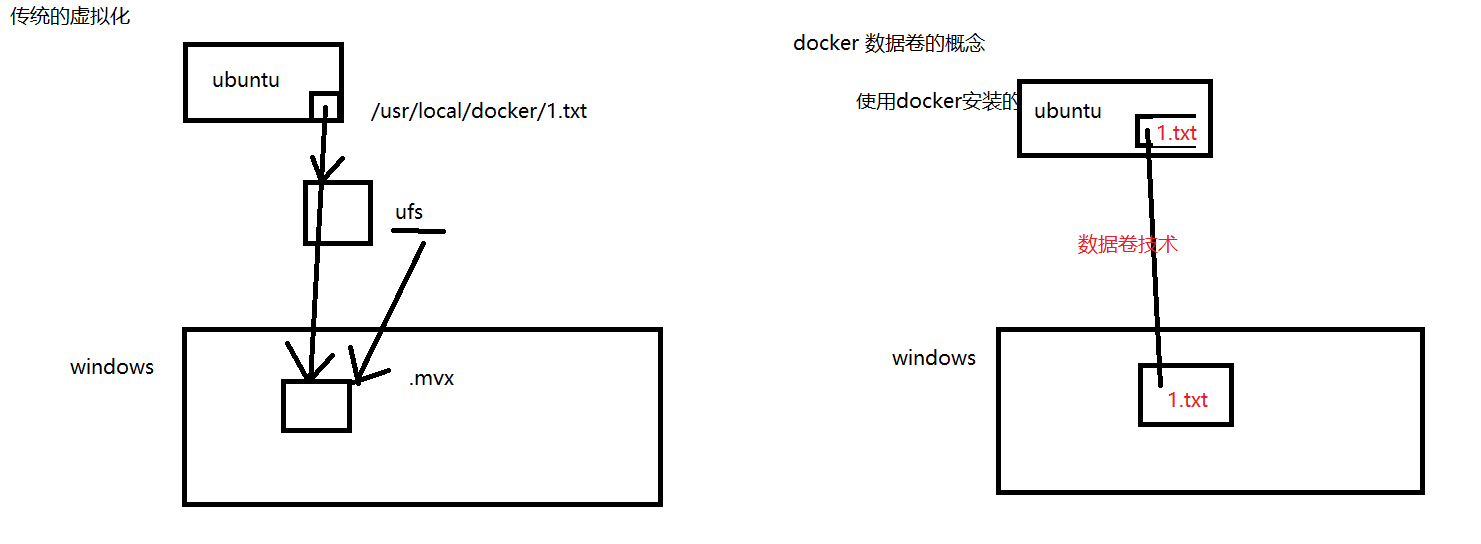
## 9.快速的删除所有容器

docker rm $(docker ps -a)

## 10.启动一个处于停止状态的容器

docker start 容器id（或者容器名）

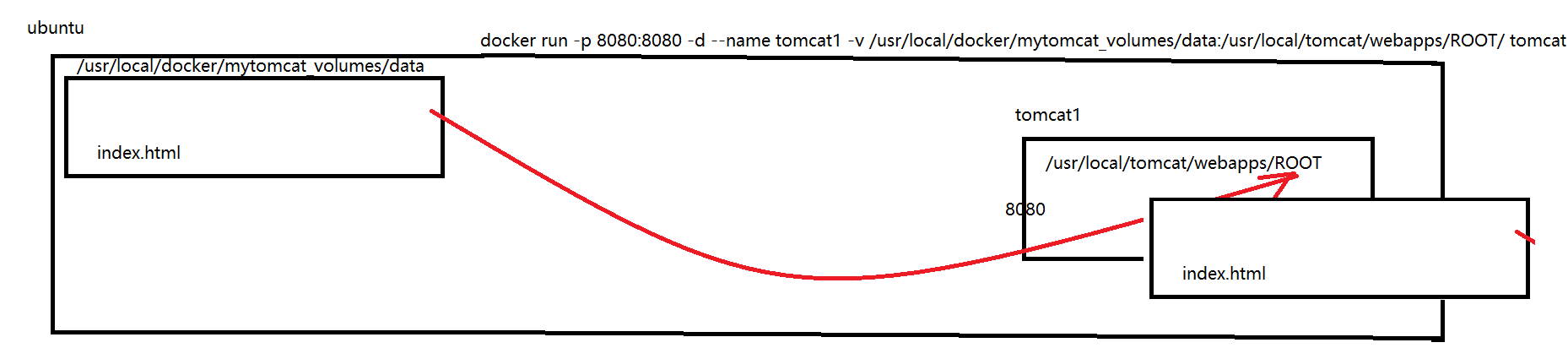
# 四、数据卷



使用数据卷的方式：

创建容器时， -v 宿主目录:容器目录 ...

此时宿主目录内的内容将和容器目录内的内容共享共用



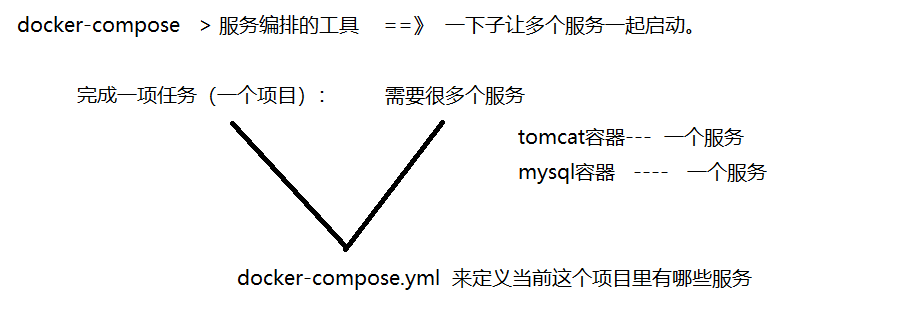
# 五、docker - compose

docker-compose实现服务的编排

一个文件： docker-compose.yml

存放多个服务

两个概念：服务和项目



## 1、如何安装docker-compose

curl -L https://github.com/docker/compose/releases/download/1.25.0-rc2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

## 2.如何编写docker-compose.yml文件

注意事项： 1.通过空格，实现缩进，缩进表示层级关系

2.冒号和值之间要有空格

|  |
| --- |
| version: '3.1'  services:  web: <== 当前服务的名字  image: tomcat <== 当前服务使用的镜像  ports: <== 当前服务的端口映射，是一个数组，数组里面用- 表示一个数据项  - 8080:8080  volumes: <== 数据卷  - ./data:/usr/local/tomcat/webapps/ROOT |

## 3.docker-compose的相关命令

docker-compose up

docker-compose up -d

docker-compose down 关闭并删除服务

docker-compose stop 服务名 关闭服务

docker-compose start 服务名 启动已关闭的服务