



Docker虚拟化容器应用与实战



第4章 Dockerfile

本章目标

- Dockerfile介绍
- Dockerfile构建过程
- 实战Dockerfile
- 实战tomcat镜像
- 构建nginx镜像



Dockerfile介绍

Dockerfile 是用来构建docker镜像的文件，命令参数脚本。

构建步骤：

- 1、编写一个dockerfile文件
- 2、 docker build 构建成为一个镜像
- 3、 docker run运行镜像
- 4、 docker push发布镜像（DockerHub 、阿里云仓库）

但是很多官方镜像都是基础包，很多功能没有，我们通常会自己搭建自己的镜像！



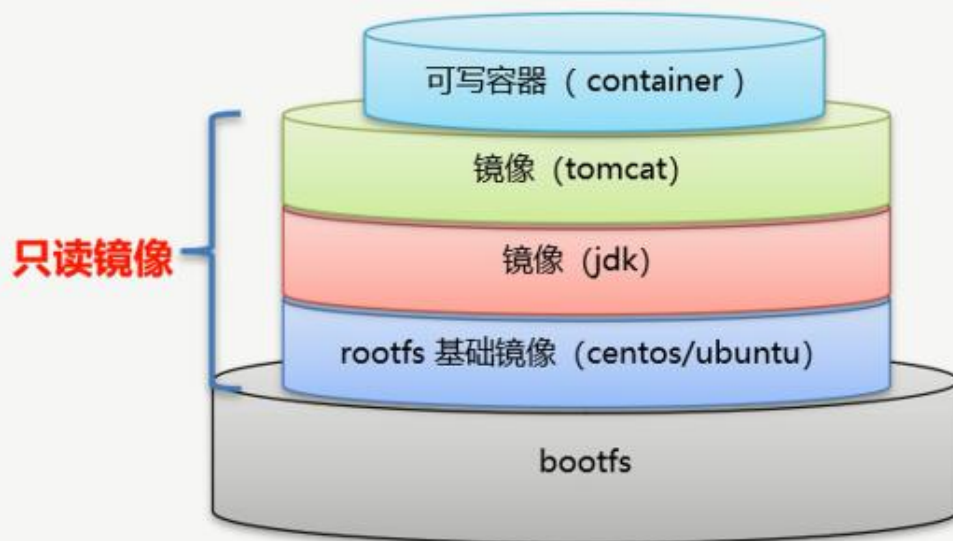
Dockerfile构建过程

基础知识:

- 1、每个保留关键字(指令) 都是必须是大写字母
- 2、执行从上到下顺序
- 3、#表示注释
- 4、每一个指令都会创建提交一个新的镜像层，并提交！



Dockerfile构建过程



Dockerfile构建过程

- Dockerfile是面向开发的，我们以后要发布项目，做镜像，就需要编写dockerfile文件，这个文件十分简单！
- Docker镜像逐渐成企业交付的标准，必须要掌握！
- DockerFile：构建文件，定义了一切的步骤，源代码
- DockerImages：通过DockerFile构建生成的镜像，最终发布和运行产品。
- Docker容器：容器就是镜像运行起来提供服务



赛博梦工厂

Cyber Works

Dockerfile常用命令

- FROM # 基础镜像，一切从这里开始构建
- MAINTAINER # 镜像是谁写的， 姓名+邮箱
- RUN # 镜像构建的时候需要运行的命令
- ADD # 编译镜像时添加文件到镜像中
- WORKDIR # 设置工作目录
- EXPOSE # 设置暴露的端口
- ENV # 设置容器环境变量
- CMD # 指定这个容器启动的时候要运行的命令
- ENTRYPOINT # 指定这个容器启动的时候要运行的命令，可以追加命令
- ONBUILD # 当构建一个被继承 DockerFile 这个时候就会运行ONBUILD的指令，触发指令
- COPY # 类似ADD，将我们文件拷贝到镜像中 ENV # 构建的时候设置环境变量！



赛博梦工厂

Cyber Works

创建一个自己的centos
1.编写Dockerfile文件
vim mydockerfile-centos
FROM centos
MAINTAINER ren<12345678@qq.com>
ENV MYPATH /usr/local
WORKDIR \$MYPATH
RUN yum -y install vim
RUN yum -y install net-tools
EXPOSE 80
CMD echo \$MYPATH
CMD echo "-----end-----"
CMD /bin/bash



#通过这个文件构建镜像

命令

- `docker build -f 文件路径 -t 镜像名:[tag] .`
- `docker build -f mydockerfile-centos -t mycentos:0.1 .`



实战测试（Tomcat镜像）

1、准备镜像文件

准备tomcat 和 jdk到当前目录。

```
→ docker-tomcat ls  
apache-tomcat-9.0.35.tar.gz  dockerfile  jdk-8u231-linux-x64.tar.gz  README  
→ docker-tomcat
```



实战测试 (Tomcat镜像)

2、编写Dockerfile

```
FROM centos
MAINTAINER ren<12345678@qq.com>
COPY README /usr/local/README #复制文件
ADD jdk-8u231-linux-x64.tar.gz /usr/local/ #复制解压
ADD apache-tomcat-9.0.35.tar.gz /usr/local/ #复制解压
RUN yum -y install vim #安装vim
ENV MYPATH /usr/local #设置环境变量
WORKDIR $MYPATH #设置工作目录
ENV JAVA_HOME /usr/local/jdk1.8.0_231 #设置环境变量
ENV CATALINA_HOME /usr/local/apache-tomcat-9.0.37 #设置环境变量
ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/lib #设置环境变量分隔符是:
EXPOSE 8080 #设置暴露端口
```



赛博梦工厂

Cyber Works

实战测试 (Tomcat镜像)

因为dockerfile命名使用默认命名 因此不用使用-f 指定文件

- `docker build -t mytomcat:0.1 .`

#run镜像

- `docker run -d -p 8080:8080 --name tomcat01 -v /home/Dockerfile/build/tomcat/test:/usr/local/apache-tomcat-9.0.37/webapps/test - v /home/Dockerfile/build/tomcat/tomcatlogs:/usr/local/apache-tomcat-9.0.37/logs mytomcat:0.1`



实战测试（构建Nginx镜像）

```
[root@localhost ~]# cd /home/
```

```
[root@localhost home]# mkdir dockerfile #创建Nginx目录
```

```
[root@localhost home]#cd dockerfile
```

```
[root@localhost dockerfile]wget http://nginx.org/download/nginx-1.12.2.tar.gz
```

```
[root@localhost dockerfile] vim run.sh
```

```
#!/bin/bash
```

```
/usr/local/nginx/sbin/nginx #开启Nginx服务
```



赛博梦工厂

Cyber Works

实战测试（构建Nginx镜像）

```
vim /home/dockerfile/Dockerfile
```

```
FROM centos:7
```

```
MAINTAINER ren <12345678@qq.com>
```

```
RUN yum install -y proc-devel gcc gcc-c++ zlib zlib-devel make openssl-devel wget
```

```
ADD nginx-1.12.2.tar.gz /usr/local
```

```
WORKDIR /usr/local/nginx-1.12.2/
```

```
RUN ./configure --prefix=/usr/local/nginx && make && make install
```

```
EXPOSE 80
```

```
EXPOSE 443
```

```
RUN echo "daemon off;" >> /usr/local/nginx/conf/nginx.conf
```

```
WORKDIR /root/nginx
```

```
ADD run.sh /run.sh
```

```
RUN chmod 755 /run.sh
```

```
CMD ["/run.sh"]
```



赛博梦工厂

Cyber Works

小结

- Dockerfile介绍
- Dockerfile构建过程
- 实战测试
- 实战tomcat镜像
- 构建nginx镜像





谢谢观赏