

Window10 docker 安装配置+Dockerfile文件编写

Window10 docker 安装配置

1. 直接利用dockerDesktop来安装docker，下载地址
`https://desktop.docker.com/win/stable/amd64/Docker%20Desktop%20Installer.exe`
2. 安装dockerDesktop的时候只需修改安装目录，其他虚拟环境配置会自动勾选。
3. 使用 PowerShell 启用 Hyper-V，以管理员身份打开 PowerShell 控制台，运行以下命令：
`Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All`
提示：通过win+Q唤起搜索页搜索powerShell，右键以管理员身份打开
4. 从任务管理器查看性能查看cpu查看虚拟化是否启用
5. 需要重启，不同系统进入BIOS模式按键不同，联想新机型有按Enter进入。找到Intel Virtual Technology选项，enabled。不同系统这个选项所在的tab的可能不一样，联想电脑是在Configuration里面
6. 重启以后再按照步骤4查看虚拟化是否启用，启用即可。
7. 打开dockerDesktop，有可能会提示安装一个wsl_update_x64 的弹窗，按提示直接下载安装就完事了。
8. 能打开dockerDesktop，初次打开会自动打开 Learner Center这个Tab，中的How do I run a container 跟着里面7步骤走即可

Dockerfile文件编写

这里以前端node项目为例，项目是react+umi3自动配置好的package.json,有npm run build等指令

1. 初学者可以按照上述8点，找到那个教程，跟着一步步来。直接第二步 `git clone https://github.com/docker/welcome-to-docker` 拉下来一个有Dockerfile配置文件的项目。
2. 打开如下

```
# Start your image with a node base image
FROM node:18-alpine

# The /app directory should act as the main application directory
WORKDIR /app

# Copy the app package and package-lock.json file
COPY package*.json ./

# Copy local directories to the current local directory of our docker image (/app)
COPY ./src ./src
COPY ./public ./public

# Install node packages, install serve, build the app, and remove dependencies at the end
RUN npm install \
    && npm install -g serve \
    && npm run build \
    && rm -fr node_modules
```

```
EXPOSE 3000
```

```
# Start the app using serve command
```

```
CMD [ "serve", "-s", "build" ]
```

3. 基本就是把你项目的东西复制到`WORKDIR /app` 这个工作目录中，把你原先要手动执行的命令以脚本的形式写给docker

```
RUN npm install \ # 安装依赖
```

```
&& npm install -g serve \ # 安装web服务器
```

```
&& npm run build \ # 构建项目代码，生成dist文件夹
```

```
&& rm -fr node_modules # 项目构建以后node_modules就不需要了，体积非常大，删掉以后镜像体积少差不多2GB。
```

```
EXPOSE 3000 # 暴露端口
```

```
# Start the app using serve command
```

```
# 运行的命令，serve -s build可以不需要cd 到dist文件夹下再执行
```

```
CMD [ "serve", "-s", "build" ]
```

这里面有个小坑

```
# 我们最好先进入dist中，这里不知道cd命令，用WORKDIR代替，但是要拼接前缀
```

```
WORKDIR /app/dist
```

```
# 直接serve的话，单页面项目的路由跳转会有问题
```

```
CMD [ "serve", "--single" ]
```

4. Dockerfile写好了以后，运行build命令

```
docker build -t tcs1619 . -t意思是在 . 当前目录下开始构建一个名为 tcs1619 的镜像（运行容器构建镜像）
```

这里FROM会拉去远程仓库镜像，这个是云端的，仓库在国外，可能会有点慢 也是并行的方式分层拉取，效率还是挺高的，后续是可复用的，所以第一次肯定会慢。

5. 构建成功了就可以run，运行 `docker run -d --name my-tcs -p 8080:3000 tcs1619`

6. 浏览器输入localhost:8080,查看页面是否运行正常大功告成。