

# 项目代码配置流程

## 软件安装

### Docker

<https://www.docker.com/products/docker-desktop/>

#### Docker Desktop

Docker Desktop is collaborative containerization software for developers. Get started and download Docker Desktop today on Mac, Windows, or Linux.

为了尽量减少环境配置的困扰，后续将在docker环境里进行开发

下面是Windows安装可能遇到的问题，如果遇到了可以查看文档

<https://cloud.tencent.com/developer/article/2208269> 和

<https://blog.csdn.net/Ninelif19/article/details/134841395>

不了解docker的同学可以阅读下面的文章，提前学习如何简单使用

<https://zhuanlan.zhihu.com/p/30713987>

#### 【全干货】5分钟带你看懂 Docker！

欢迎大家前往 腾讯云社区，获取更多腾讯海量技术实践干货哦~ 作者 | 唐文广：腾讯工程师，负责无线研发部地图测试。导语：Docker,近两年才流行起来的超轻量级虚拟...

### VScode

建议安装VScode作为代码编辑器

<https://code.visualstudio.com/download>

#### Download Visual Studio Code - Mac, Linux, Windows

Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows. Download Visual Studio Code to experience a redefined code editor, optimized for building and debuggin

下载后可以在本地配置C++环境，但本课程可以在Docker里操作，所以如果配不好也没有关系（因为不同电脑可能会遇到不同的复杂的配置问题，长远看还是建议配好）下面分别是Windows和Mac的

<https://zhuanlan.zhihu.com/p/87864677>

#### VSCode配置C/C++环境

VSCode简介VSCode是一款微软出的轻量级编辑器，它本身只是一款文本编辑器而已，所有的功能都是以插件扩展的形式所存在，想用什么功能就安装对应的扩展即可，...

<https://blog.csdn.net/xiaonuo911teamo/article/details/111704610>

## mac上VScode如何配置C++使用(方法出自于官网,很傻瓜,也很明白)\_mac如何找到c++ 标准库的安装路径或头文件目录-CSDN博客

文章浏览阅读2.1k次,点赞5次,收藏15次。下面的方法基本是照抄官网的,只是翻译了一下,再加上一丢丢自己的备注。本来...

## Git

如果可以访问GitHub看这个



<https://zhuanlan.zhihu.com/p/369486197>

## 还不会使用 GitHub? GitHub 教程来了! 万字图文详解

在编程届有个共识,想要成为一个合格的程序员必须要掌握 GitHub 的用法!接下来,我们用两万字加一百张图片从头到尾的给你介绍 GitHub 的具体使用,通过这个...

翻不了墙用Gitee也可以

[https://blog.csdn.net/qq\\_38403590/article/details/120907444](https://blog.csdn.net/qq_38403590/article/details/120907444)

## gitee教程(超全,超详细,超长)-CSDN博客

文章浏览阅读5.9w次,点赞273次,收藏1.5k次。俺今天刚写完 专利技术交底书,于是有了一点点空闲时间。虽然常用github,但心里面总感觉缺点什么,于是有了这篇文档,希望这篇文档能在我忘记怎么用gitee的时候瞬间教会我。并且也能让我在写文档...

## 自选书目推荐

1. **Effective C++**: 实用的编程技巧和最佳实践,可以立即应用于你自己的项目和代码中,提高代码的质量和性能(用语也算是相对幽默的)
2. **CSAPP (深入理解计算机系统)**: 加深实际的计算机理论知识的理解,这本书最有价值的是他的课程lab。很有助于新手入门学会配环境和通过代码理解OSI七层架构,网上也有大量教程和讲解,适合自驱力强的同学自学。

<http://csapp.cs.cmu.edu/3e/labs.html>

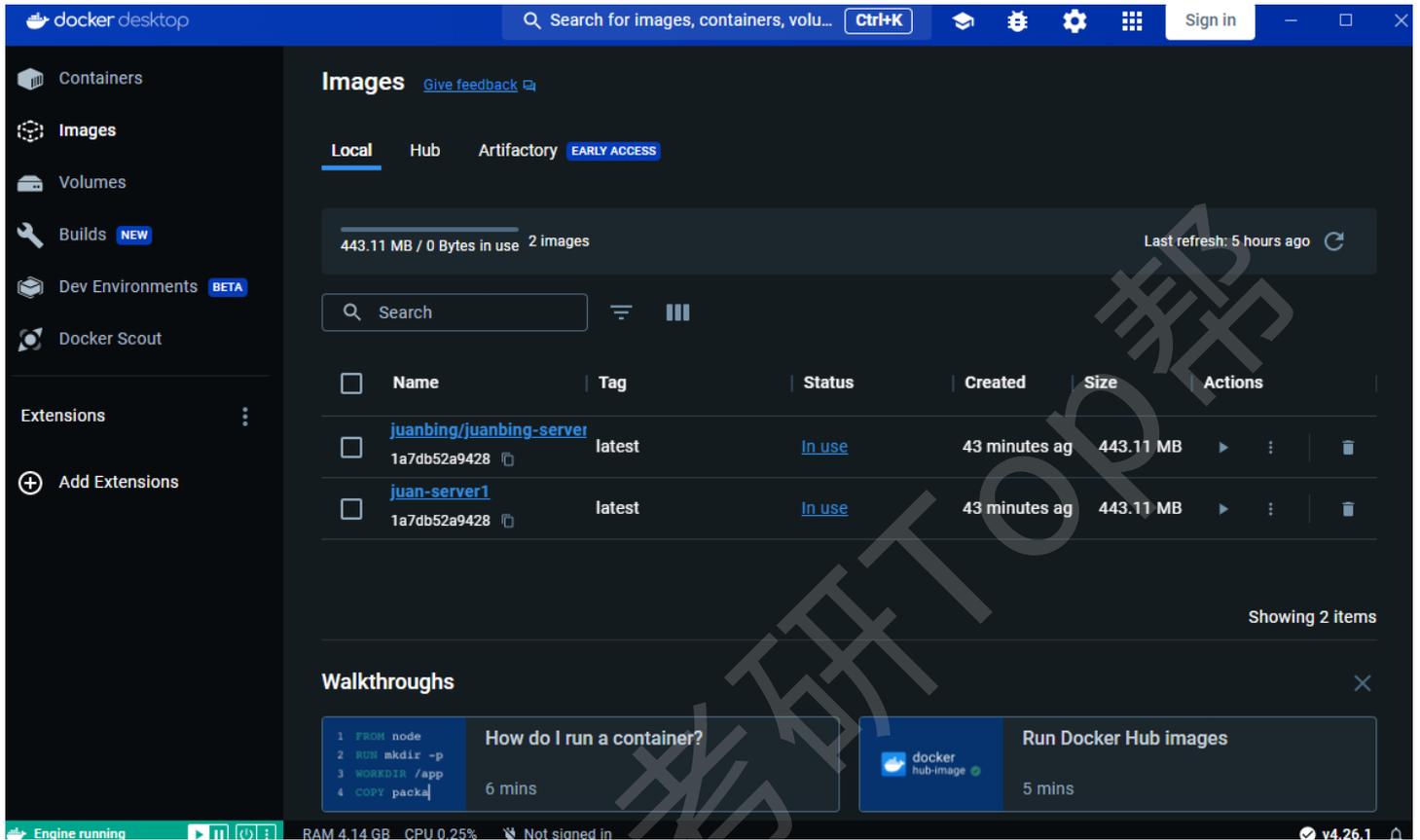
## CS:APP3e, Bryant and O'Hallaron

Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e) Randal E. Bryant and David R. O'Hallaron, Carnegie Mellon University Request a CS:APP instructor's account Change your CS:APP password Home

3. **C++ Primer (《C++ Primer》)**: 这是一本经典的C++学习教材,由Stanley B. Lippman、Josée Lajoie和Barbara E. Moo合著。它逐步引导你深入了解C++的基础知识和高级概念(适合当字典)
4. **Effective Modern C++ (《Effective Modern C++》)**: Scott Meyers的这本书是《Effective C++》系列的续集,重点介绍了C++11、C++14和C++17的新特性和最佳实践。
5. **STL 源码剖析 (《STL 源码剖析》)**: 这本书由侯捷编写,深入剖析了C++标准模板库(STL)的内部工作原理,对于理解STL非常有帮助。

## 前置步骤

确保你已经按照[预习文档](#)成功安装Docker，并启动Docker Desktop，能正常打开页面即可



成功启动后可以在VScode或者电脑的cmd命令行里输入docker，会显示Docker相关的命令，说明安装成功

```
PS C:\Windows\system32> docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run          Create and run a new container from an image
exec        Execute a command in a running container
ps          List containers
build       Build an image from a Dockerfile
pull        Download an image from a registry
push        Upload an image to a registry
images      List images
login       Log in to a registry
logout      Log out from a registry
search      Search Docker Hub for images
```

下面是两种配置项目的方法

推荐大家用方案1了

## 下载镜像

优势：不容易出问题

劣势：你如果自己想修改代码测试会稍微麻烦一些，（可以看偷懒方案）

下面是步骤

1. 打开VSCode,在terminal中运行命令

Mac电脑 `docker pull juanbing/juanbing-webserver`

Windows电脑 `docker pull juanbing/juanbing-webserver-win`

这句意思就是下载卷饼的卷饼存储在Docker hub 上的server镜像

```
● → C-Server git:(main) x docker pull juanbing/juanbing-webserver
Using default tag: latest
latest: Pulling from juanbing/juanbing-webserver
ea4ac7c2aed5: Pull complete
24aeb0a16855: Pull complete
92f3526ccef0: Pull complete
3e6df0ee7a7b: Pull complete
e1dfd81a9c9d: Pull complete
bc4481095e77: Pull complete
4f4fb700ef54: Pull complete
5e74de9ba24f: Pull complete
7fa88aab8f80: Pull complete
51170edf717b: Pull complete
Digest: sha256:0f5a81924c27c569638f04fdf7b21d9db28b2a835f61a2254d41c104c4512ca3
Status: Downloaded newer image for juanbing/juanbing-webserver:latest
docker.io/juanbing/juanbing-webserver:latest
```

2. 这样你就下载下来了我的镜像，接下来执行命令运行这个镜像（windows电脑下面所有命令里的

`juanbing/juanbing-webserver` 都需要换成 `juanbing/juanbing-webserver-`  
`docker run -it -p 7001:8080 juanbing/juanbing-webserver bash`

我们将本机的端口7001映射到容器的8080端口,运行成功后输入ls可以看到当前目录，每个server开头的文件夹都是不同阶段的完整的服务器，进入文件夹后就可以运行我们的服务器（同一时刻只能有一个端口被占用，7001如果用了可以换成7002等等）

```
docker.io/juanbing/juanbing-webserver:latest
● → C-Server git:(main) x docker run -it -p 7001:8080 juanbing/juanbing-webserver bash
root@85470f9629d1:/usr/src/myapp# ls
Dockerfile  post_data.txt  server-2-http  server-4-database  start.sh
nginx.conf  server-1-hello1rd  server-3-Logger  server-5-epoll  test.sh
```

3. 进入容器并运行我们的第一个服务器，进入server-1的目录，g++编译myserver.cpp，运行可执行文件

`cd server-1-hello1rd`

`g++ mysserver.cpp -o yourserver` (这里yourserver是生成的可执行文件的名称，你可以随意指定)

```
./yourserver
```

```
nginx.com server-1-helloworld server-5-Logger server-5-epoll test.sh  
root@85470f9629d1:/usr/src/myapp# cd server-1-helloworld/  
root@85470f9629d1:/usr/src/myapp/server-1-helloworld# g++ myserver.cpp -o yourserver  
root@85470f9629d1:/usr/src/myapp/server-1-helloworld# ./yourserver
```

4. 运行成功后，再打开一个新的终端

执行命令

Windows系统: `curl.exe http://localhost:7001/`

MAC系统: `curl http://localhost:7001/`

或者直接在浏览器输入 `http://localhost:7001/`

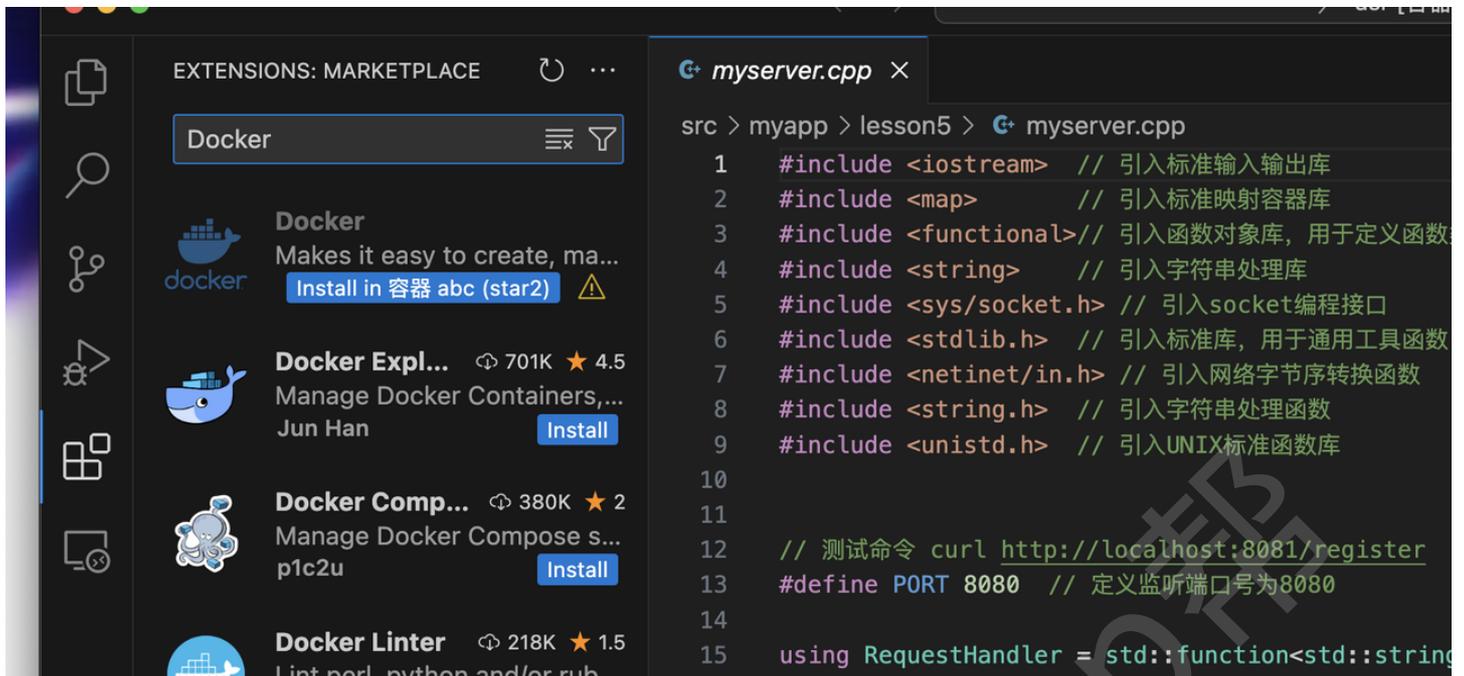


到这里说明配置成功，你的第一个本地服务器已经成功运行起来了，他可以响应请求并返回Hello, World

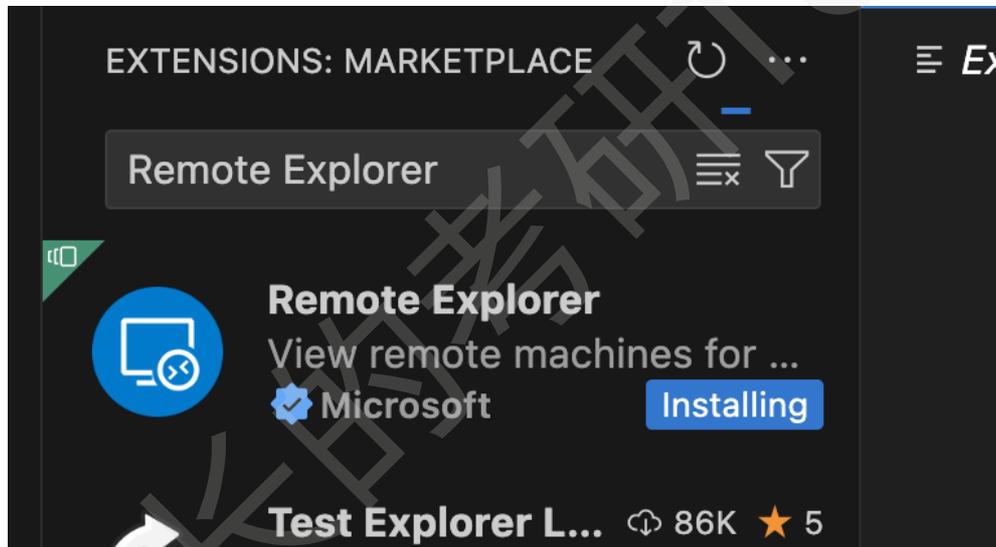
但这样并不能方便的看到代码

## 安装插件（不用每次都重新生成新的容器）

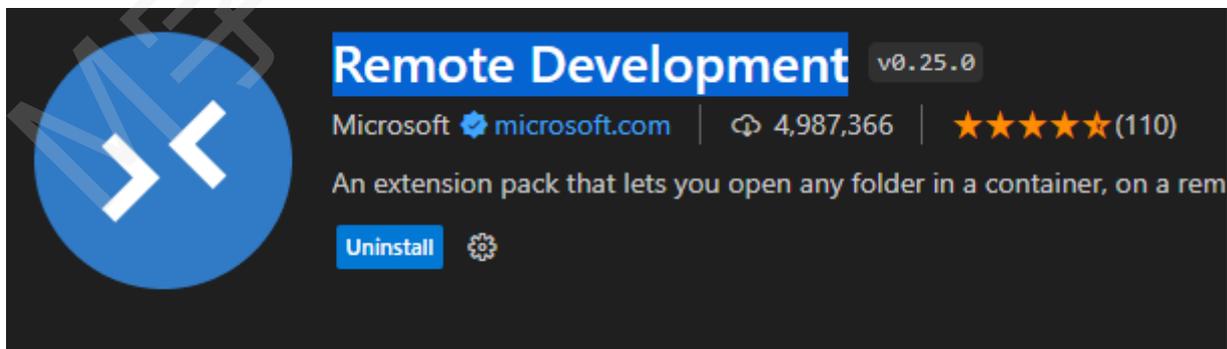
如果你之前成功安装了VScode可以安装Docker插件（左下角点击四个方块进入商城）



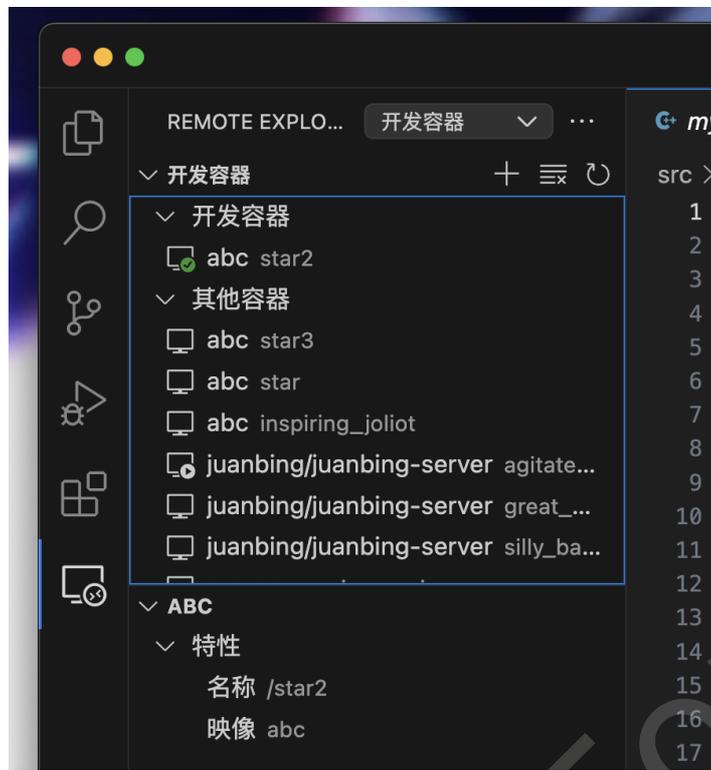
以及Remote Explorer



以及Remote Development

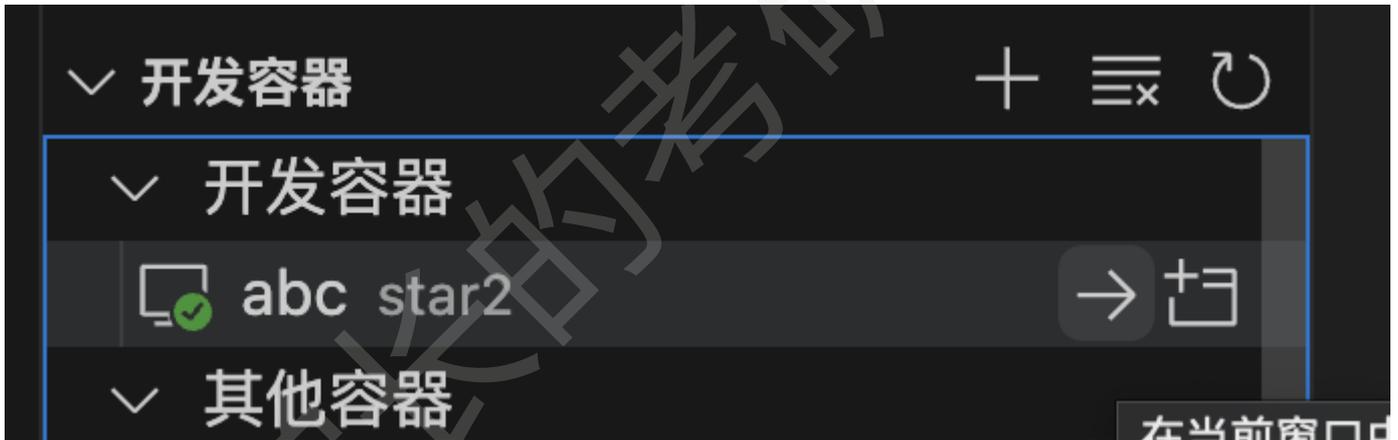


然后点击下面的Remote Explorer

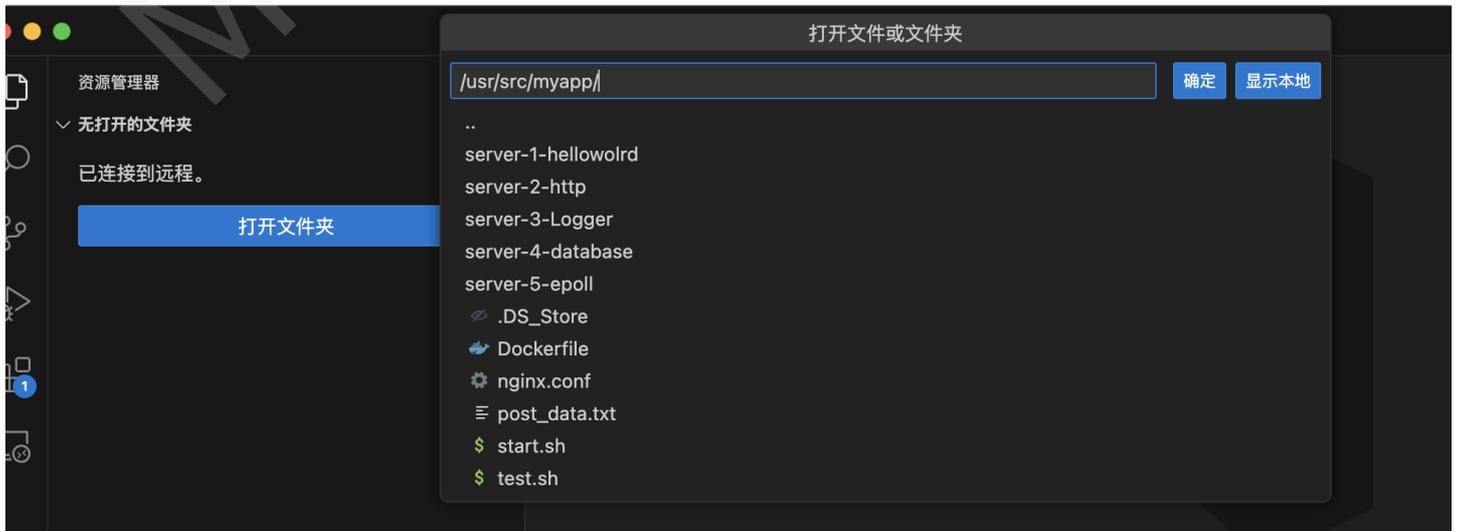


可以看到自己的容器（们）

左边是镜像名右边是容器名字点击箭头进入容器



这样就可以在容器里编辑代码测试服务器，不用每次修改完代码都重新生成镜像了，有时候会失败，退出再进去一次就好，如果需要目录的话我们的工作目录是`/usr/src/app`，点击确定即可进入代码目录



# Q/A

## 1. 打开VScode代码报错

```
Cpp > C++project > lesson5 > myserver.cpp > ...
1  #include <iostream> // 引入标准输入输出库
2  #include <map> // 引入标准映射容器库
3  #include <functional> // 引入函数对象库，用于定义函数类型
4  #include <string> // 引入字符串处理库
5  #include <sys/socket.h> // 引入socket编程接口
6  #include <stdlib.h> // 引入标准库，用于通用工具函数
7  #include <netinet/in.h> // 引入网络字节序转换函数
8  #include <string.h> // 引入字符串处理函数
9  #include <unistd.h> // 引入UNIX标准函数库
10
```

正常情况，我们的代码是针对Linux环境编写，在Windows和MAC系统中是无法运行的~，要按照操作流程进入容器里编写和运行我们的服务器

## 2. 如何删除之前的Docker容器，避免端口一直占用

```
PS C:\Users\y-yang\Documents\server> docker ps -a
```

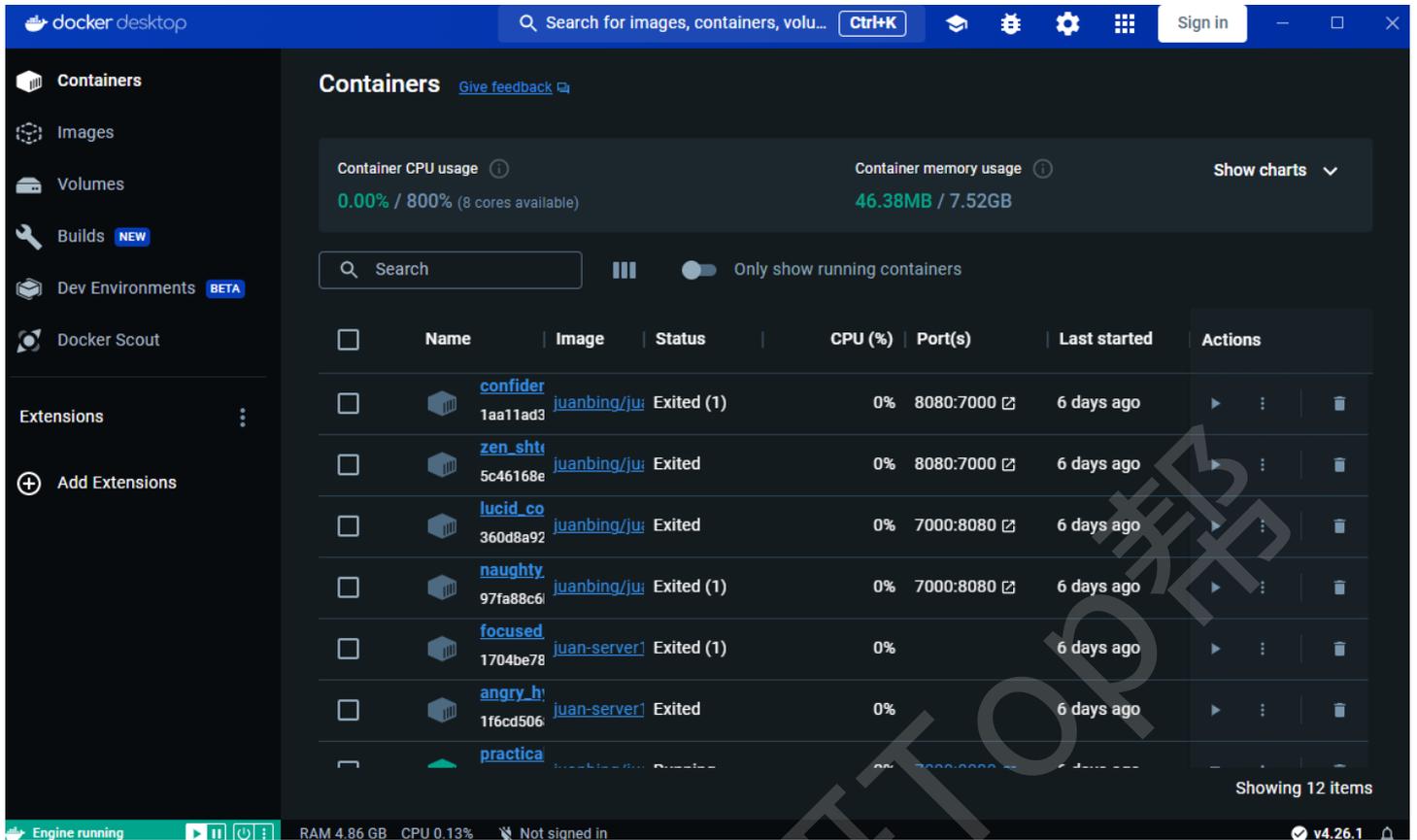
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9f3784b8e866	test_log	"bash"	8 hours ago	Up 8 hours	80/tcp, 8081/tcp, 0.0.0.0:7002->8080/tcp	serene_noether
33fd296f9810	juanbing/juanbing-server	"/bin/bash"	32 hours ago	Exited (130) 32 hours ago		interesting_proskuriakova
2befa05c7f5e	1a7db52a9428	"/bin/bash"	6 days ago	Exited (130) 32 hours ago		crazy_chaum
39534589bb38	1a7db52a9428	"/bin/bash"	6 days ago	Up 6 days	80/tcp, 8081/tcp, 0.0.0.0:7001->8080/tcp	sad_mendelev
1ab3a06e1325	1a7db52a9428	"/bin/bash"	6 days ago	Created		gifted_goldwasser
44802c00a52b	1a7db52a9428	"/bin/bash"	6 days ago	Up 6 days	80/tcp, 8081/tcp, 0.0.0.0:7000->8080/tcp	practical_hellman
1f6cd5068d69	1a7db52a9428	"/bin/bash"	6 days ago	Exited (0) 6 days ago		angry_hypatia
1704be784ecc	1a7db52a9428	"/start.sh"	6 days ago	Exited (1) 6 days ago		focused_dijkstra
97fa88c6b9b6	1a7db52a9428	"/start.sh"	6 days ago	Exited (1) 6 days ago		naughty_mcclintock
360d8a929364	1a7db52a9428	"bash"	6 days ago	Exited (0) 6 days ago		lucid_colden
5c46168e5589	1a7db52a9428	"bash"	6 days ago	Exited (0) 6 days ago		zen_shtern
1aa11ad3ad1e	1a7db52a9428	"/start.sh"	6 days ago	Exited (1) 6 days ago		confident_swanson

执行docker ps -a看到所有容器，最左一列就是容器的ID（PORTS那一栏可以看到端口映射）

执行docker stop 容器ID

停下容器后用docker rm 容器ID删除

也可以直接在Docker软件中删除



### 3. run之后报错显示端口被占用

```
ⓧ → C-Server-main docker run -it -p 7001:8080 juanbing/juanbing-webserver bash
docker: Error response from daemon: driver failed programming external connectivity on endpoint musing_norhcutt (0f6e6530e2a495dfa68b82b5da972b6737856e9759b1602c4da374ef9492de70): Bind for 0.0.0.0:7001 failed: port is already allocated.
```

按照上面的步骤停止并删掉正在占用端口的容器~或者自己换一个端口7009之类的再生成新的

### 4. 如何更新新的镜像

我上传新代码后，可以通过运行自动获取最新的镜像

```
docker pull juanbing/juanbing-webserver
```

### 5. Windows系统报错warning

```
E:\code\cpp>docker run -it -p 7001:8080 juanbing/juanbing-server bash
WARNING: The requested image's platform (linux/arm64/v8) does not match the detected host platform (linux/amd64/v3) and no specific platform was requested
root@71110c9b55f7:/usr/src/myapp#
```

warning是正常警告，因为我是在Mac生成的镜像，所以不是很兼容，但是可以运行的，底下已经变成root就说名已经进入了docker，接下来按照偷懒方案就行